

Sybase SQL Server Manager™ User's Guide

SQL Server Manager 11.0

Document ID: 32111-01-1100-02

Last Revised: October 30, 1995

Principal author: Systems Management Products Technical Publications

Contributing authors: Systems Management Products and Technical Support staff

Document ID: 32111-01-1100

This publication pertains to SQL Server Manager 11.0 of the Sybase database management software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Document Orders

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor.

Upgrades are provided only at regularly scheduled software release dates.

Copyright © 1989–1995 by Sybase, Inc. All rights reserved.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase Trademarks

APT-FORMS, Data Workbench, DBA Companion, Deft, GainExposure, Gain *Momentum*, Navigation Server, PowerBuilder, Powersoft, Replication Server, SA Companion, SQL Advantage, SQL Debug, SQL Monitor, SQL SMART, SQL Solutions, SQR, SYBASE, the Sybase logo, Transact-SQL, and VQL are registered trademarks of Sybase, Inc. Adaptable Windowing Environment, ADA Workbench, AnswerBase, Application Manager, APT-Build, APT-Edit, APT-Execute, APT-Library, APT-Translator, APT Workbench, Backup Server, Bit-Wise, Client-Library, Client/Server Architecture for the Online Enterprise, Client/Server for the Real World, Client Services, Configurator, Connection Manager, Database Analyzer, DBA Companion Application Manager, DBA Companion Resource Manager, DB-Library, Deft Analyst, Deft Designer, Deft Educational, Deft Professional, Deft Trial, Developers Workbench, DirectCONNECT, Easy SQR, Embedded SQL, EMS, Enterprise Builder, Enterprise Client/Server, Enterprise CONNECT, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, EWA, Gain Interplay, Gateway Manager, InfoMaker, Interactive Quality Accelerator, Intermedia Server, IQ Accelerator, Maintenance Express, MAP, MDI, MDI Access Server, MDI Database Gateway, MethodSet, Movedb, Navigation Server Manager, Net-Gateway, Net-Library, New Media Studio,

OmniCONNECT, OmniSQL Access Module, OmniSQL Gateway, OmniSQL Server, OmniSQL Toolkit, Open Client, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open Solutions, PC APT-Execute, PC DB-Net, PC Net Library, Powersoft Portfolio, Replication Agent, Replication Driver, Replication Server Manager, Report-Execute, Report Workbench, Resource Manager, RW-DisplayLib, RW-Library, SAFE, SDF, Secure SQL Server, Secure SQL Toolset, SKILS, SQL Anywhere, SQL Code Checker, SQL Edit, SQL Edit/TPU, SQL Server, SQL Server/CFT, SQL Server/DBM, SQL Server Manager, SQL Server Monitor, SQL Station, SQL Toolset, SQR Developers Kit, SQR Execute, SQR Toolkit, SQR Workbench, Sybase Client/Server Interfaces, Sybase Gateways, Sybase Intermedia, Sybase Interplay, Sybase IQ, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase Synergy Program, Sybase Virtual Server Architecture, Sybase User Workbench, SyBooks, System 10, System 11, the System XI logo, Tabular Data Stream, The Enterprise Client/Server Company, The Online Information Center, Warehouse WORKS, Watcom SQL, WebSights, WorkGroup SQL Server, XA-Library, and XA-Server are trademarks of Sybase, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Restricted Rights

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., 6475 Christie Avenue, Emeryville, CA 94608.

Table of Contents

Preface

Audience	xxiii
How to Use This Book	xxiii
Related Documents	xxiv
Other Sources of Information	xxv
Conventions	xxvi
Style Conventions	xxvi
Shortcuts	xxvi
Privileges	xxvii
Mouse Buttons	xxviii
Screen Images	xxviii
SQL Server 11.0 Features	xxviii
If You Need Help	xxix
Using Online Help	xxix
Contacting Technical Support	xxx

1. Introduction

About SQL Server Manager	1-1
Performing System and Database Administration Tasks	1-1
Making It Easy	1-2

2. Getting Started

What's in This Chapter	2-1
Starting SQL Server Manager	2-1
Starting with the SQL Server Manager Icon	2-1
Starting with the Run... Command	2-2
Command Line Options	2-2
Connecting to SQL Server	2-4
Diagnosing an Unsuccessful Connection	2-5
Changing an Expired Password	2-6
Exiting SQL Server Manager	2-7
Tutorial	2-7
Before You Begin	2-8
Lesson 1: General Activities	2-8
Part 1: Start SQL Server Manager and Connect to SQL Server	2-8
Part 2: Customize the SQL Server Manager Display	2-9

Part 3: Navigate Voyager	2-11
Part 4: Navigate Between Objects	2-14
Part 5: Open Another Voyager Window	2-18
Lesson 2: SQL Server Activities	2-19
Part 1: Start SQL Server Manager	2-19
Part 2: Create SQL Server Logins	2-20
Part 3: Create a Database	2-22
Part 4: Exit	2-24
Lesson 3: Database Activities	2-25
Part 1: Start SQL Server Manager	2-25
Part 2: Create Database Users	2-25
Part 3: Create a Database Table	2-28
Part 4: Assign Permission on the Table	2-31
Part 5: Generate Database DDL	2-35
Part 6: Clean Up and Exit	2-38

3. Introducing SQL Server Manager Windows and Dialog Boxes

What's in This Chapter	3-1
Using the Voyager Window	3-1
Features of Voyager	3-2
Moving Through Voyager	3-4
Opening Another Voyager Window	3-5
Customizing the Display	3-6
Selecting Display Formats	3-6
Filtering by Name	3-9
Using Drag-and-Drop Shortcuts	3-10
Menus and Toolbars	3-12
Standard Menus	3-12
Context-Sensitive Menus	3-13
Pull-Down Menus	3-14
Shortcut Menus	3-14
Toolbar and Status Bar	3-16
Standard Toolbar	3-16
Status Bar	3-18
Types of Dialog Boxes	3-19
Object Dialog Boxes	3-20
Using Dialog Box Tabs	3-20
Navigating Between Objects	3-21
Closing an Object Dialog Box	3-22
Command Dialog Boxes	3-22
Closing a Command Dialog Box	3-23

Confirmation Dialog Boxes	3-23
Closing a Confirmation Dialog Box	3-24
Message Dialog Boxes	3-24
Closing a Message Dialog Box	3-25
Refreshing Window and Dialog Box Displays	3-25
Moving Between Open Windows and Dialog Boxes	3-26

4. Managing the SQL Server Operating Environment

What's in This Chapter	4-1
Connecting to and Disconnecting from SQL Server	4-1
Connecting	4-2
Diagnosing an Unsuccessful Connection	4-4
Disconnecting from SQL Server	4-4
Displaying SQL Server Details	4-6
Examining SQL Server Properties	4-6
Examining SQL Server Objects	4-6
Shutting Down SQL Server	4-7
Configuring SQL Server	4-8
Who Can Set Configuration Parameters	4-9
Parameters Requiring Restart	4-9
Configuring a Release 10.x SQL Server	4-10
Configuring a Release 11.0 SQL Server	4-11
Setting Configuration Parameters	4-12
Displaying and Killing SQL Server Processes	4-13
Displaying Processes	4-13
Viewing Additional Process Details	4-14
Viewing SQL Server 11.0 Process Information	4-15
Killing a Process	4-15
Enabling Communication with Other Servers	4-15
Creating a Remote Server	4-16
Deleting a Remote Server	4-18
Managing Remote Server Logins	4-19
Displaying Remote Server Details	4-22
Generating and Executing SQL Server DDL Scripts	4-23
Generating SQL Server DDL	4-24
Executing SQL Server DDL	4-25
Examining and Editing DDL Script Files	4-26

Checking Consistency in SQL Server 11.0	4-29
---	------

5. Managing Cache

What's in This Chapter	5-1
Creating and Deleting a Named Data Cache	5-1
Creating a Cache	5-2
Deleting a Cache	5-3
Displaying and Modifying Cache Information	5-4
Displaying Cache Properties	5-5
Changing Cache Size or Type	5-6
Defining Buffer Pools	5-6
Creating a Buffer Pool	5-8
Deleting a Buffer Pool	5-9
Binding Objects to a Cache	5-9
Rules for Binding	5-9
Creating a Binding	5-10
Deleting a Binding	5-12

6. Managing Physical Resources

What's in This Chapter	6-1
Strategies for Recovery and Performance	6-1
Creating and Deleting Devices	6-2
Considerations for Creating a New Device	6-2
Creating a Database Device	6-2
Deleting a Database Device	6-5
Displaying and Modifying Device Information	6-6
Examining Database Device Properties	6-6
Examining Allocations on a Database Device	6-7
Navigating to a Database	6-8
Examining Segment Mapping on a Database Device	6-9
Navigating to a Segment	6-10
Using Disk Mirroring	6-10
Adding Disk Mirroring	6-11
Disabling and Re-Enabling Disk Mirroring	6-13
Storage Management Considerations and Tips	6-16
Managing for Recovery	6-16
Changing Default Device Designation	6-16
Using Disk Mirroring	6-17
Separating Data and Transaction Log	6-17
Managing for Performance	6-17

7. Managing Databases and Segments

What's in This Chapter	7-1
Creating and Deleting a User Database	7-1
Considerations for Creating a New Database	7-1
Estimating Database Size	7-1
Deciding Where to Allocate Storage	7-2
Creating a Database	7-3
Deleting a Database	7-6
Deleting a Database	7-6
Deleting a Damaged Database	7-7
Displaying and Modifying Database Information	7-7
Examining Database Details	7-8
Modifying Database Storage Allocations	7-9
Allocating Space on the Same Device	7-10
Allocating Space on a New Device	7-10
Examining Database Storage Allocations	7-11
Transferring Database Ownership	7-12
Changing Log Buffer Size	7-13
Generating and Executing Database DDL Scripts	7-13
Generating Database DDL	7-14
Executing Database DDL	7-15
Examining and Editing DDL Script Files	7-16
Changing Database Options	7-18
Instructions	7-19
List of Options	7-20
Issuing a Database Checkpoint	7-22
Using Database Segments	7-23
Creating and Deleting Segments	7-23
Creating a Segment	7-23
Deleting a Segment	7-25
Displaying and Modifying Segment Information	7-25
Examining Segment Details	7-26
Navigating to a Database Device	7-27
Extending a Segment	7-27
Examining Objects Stored on a Segment	7-28
Navigating to an Index or Table on a Segment	7-29
Using Segment Thresholds	7-29
Threshold Concepts: Last-Chance Threshold	7-29
Threshold Concepts: Hysteresis Value	7-30
Creating a Threshold	7-30
Deleting a Threshold	7-32

Modifying a Threshold	7-32
Preparing for Backup and Recovery	7-33
Developing a Backup and Recovery Plan	7-33
Identifying the Backup Server	7-34
Managing Dump Devices	7-34
Creating a Dump Device:	7-34
Deleting a Dump Device	7-36
Examining Dump Device Properties	7-37
Backing up a Database	7-38
Scheduling Database Backups	7-38
Guidelines for Backing up User Databases	7-38
Guidelines for Backing up the <i>master</i> Database	7-39
Using the Database Backup Dialog Box	7-39
Responding to Volume Change Prompts	7-42
To Proceed with the Backup	7-43
To Abort the Backup	7-44
To Get Information on Backup Server Messages	7-44
Restoring a Database	7-44
Deciding How to Restore	7-44
Recovery Considerations	7-45
Using the Database Restore Dialog Box:	7-45
Responding to Volume Change Prompts	7-48
To Abort the Restore	7-49
To Proceed with the Restore	7-49
To Get Information on Backup Server Messages	7-50
Checking Database Consistency	7-50
Instructions:	7-51
Consistency Check Options	7-52
Overall Database Option	7-52
Database Allocations Option	7-53
System Catalog Option	7-54
Other dbcc Command Option	7-55
Setting a Database Online	7-55

8. Controlling Access

What's in This Chapter	8-1
About Access Privileges and Permissions	8-1
Access Based on Administrative Role	8-2
Access Based on Special User Status	8-3
Access Based on Group Membership	8-4

Access Based on Command Permissions	8-4
Access Based on Object Permissions	8-5
Access Example.....	8-7
Creating and Deleting SQL Server Logins	8-7
Creating a SQL Server Login.....	8-8
Deleting a SQL Server Login.....	8-9
Displaying and Modifying Login Information	8-10
Modifying Login Information.....	8-11
Locking a SQL Server Login.....	8-12
Changing the Login Password.....	8-13
Displaying Databases Owned by a Login.....	8-14
Sharing Logins by Using Aliases	8-14
Adding an Alias to a Login.....	8-15
Deleting an Alias from a Login	8-16
Navigating to Users.....	8-16
Creating and Deleting Users	8-17
Creating a User	8-18
Deleting a User	8-19
Displaying and Modifying User Information.....	8-20
Displaying User Properties	8-21
Modifying Group Membership and User Aliases.....	8-21
Navigating to Logins (Aliases).....	8-22
Displaying Objects Owned by the User	8-23
Navigating to an Owned Object	8-23
Granting and Revoking Object Permissions for a User.....	8-24
Navigating to Object Permissions Information.....	8-27
Granting and Revoking Command Permissions for a User.....	8-27
Creating and Deleting Groups	8-29
Creating a Group	8-30
Deleting a Group	8-31
Removing Users from a Group	8-31
Deleting an Empty Group.....	8-32
Displaying and Modifying Group Information	8-33
Displaying Group Properties	8-34
Navigating to Users.....	8-35
Modifying Group Membership:.....	8-35
Granting and Revoking Object Permissions for a Group.....	8-36
Navigating to Object Permissions Information.....	8-38
Granting and Revoking Command Permissions for a Group	8-39

Summary of Access Privileges and Permissions	8-41
Summary of Transferable Privileges and Permissions	8-46

9. Managing Database Objects

What's in This Chapter	9-1
Operations Common to All Objects	9-1
Creating an Object	9-1
Closing an Object's Dialog Box	9-3
Displaying Details About an Object	9-3
Displaying Object Properties	9-4
Displaying SQL Server and Database Objects	9-5
Displaying an Object Relative to Other Objects	9-6
Deleting an Object	9-7
Managing Defaults	9-8
Creating a Default:	9-9
Example	9-10
Displaying Default Properties	9-12
Displaying Default Dependencies	9-13
Navigating to Objects with Dependencies on a Default	9-13
Binding a Default	9-14
Binding a Default to a Table Column	9-14
Binding a Default to a User Datatype	9-15
Deleting a Binding	9-16
Managing Indexes	9-17
Creating an Index	9-18
Displaying Index Properties	9-23
Placing an Index on a Segment	9-24
Navigating to a Segment	9-26
Checking Index Consistency	9-26
Instructions	9-26
Consistency Check Options	9-27
Updating Statistics	9-28
Managing Procedures	9-29
Creating a Procedure	9-29
Displaying Procedure Properties	9-32
Displaying Procedure Dependencies	9-34
Navigating to Objects with Dependencies on a Procedure	9-35
Updating User and Group Procedure Permissions	9-36
Navigating to Users and Groups	9-38

Managing Rules	9-39
Creating a Rule	9-40
Displaying Rule Properties	9-42
Displaying Rule Dependencies	9-43
Navigating to Objects with Dependencies on a Rule	9-43
Binding a Rule	9-44
Binding a Rule to a Table Column	9-45
Binding a Rule to a User Datatype	9-45
Deleting a Binding	9-46
Managing Triggers	9-47
Creating a Trigger	9-48
Example	9-50
Displaying Trigger Properties	9-52
Displaying Trigger Dependencies	9-53
Navigating to Objects with Dependencies on a Trigger	9-53
Managing User Datatypes	9-54
Creating a User Datatype	9-54
Displaying User Datatype Properties	9-57
Displaying User Datatype Dependencies	9-58
Navigating to Objects with Dependencies on a User Datatype ...	9-58
Managing Views	9-59
Creating a View	9-59
Example	9-61
Displaying View Properties	9-64
Displaying View Dependencies	9-65
Navigating to Objects with Dependencies on the View	9-66
Updating User and Group View Permissions	9-66
Displaying the Information	9-66
Setting the Permissions	9-68
Navigating to Users and Groups	9-69

10. Managing User Tables

What's in This Chapter	10-1
Creating a Table	10-1
Using Constraints	10-2
Implementing Defaults	10-2
Implementing Rules and Check Constraints	10-2
Implementing Referential Constraints	10-3
Constraints Summary	10-3
Instructions	10-4
Specifying Column Definitions	10-7

Specifying Column Constraints	10-8
Deleting a Table	10-11
Displaying Table Properties	10-11
Adding Columns to a Table	10-13
Specifying Column Definitions	10-13
Specifying Column Constraints	10-14
Check Constraint Format	10-16
Setting Maximum Rows Per Page	10-16
Displaying Table Dependencies	10-17
Navigating to Objects with Dependencies on a Table	10-18
Updating User and Group Table Permissions	10-18
Displaying the Information	10-18
Setting the Permissions	10-20
Navigating to Users and Groups	10-21
Placing a Table on a Segment	10-22
Navigating to Segments	10-24
Displaying Indexes and Triggers	10-24
Navigating to Indexes and Triggers	10-25
Creating a New Index or Trigger	10-25
Creating Check Constraints on a Table	10-25
Check Constraint Format	10-27
Partitioning and Unpartitioning a Table	10-27
Rules for Partitioning	10-28
Partitioning	10-28
Unpartitioning	10-29
Creating Referential Constraints on a Table	10-30
This Table References Another	10-32
This Table Referenced by Another	10-33
Recompiling Procedures and Triggers	10-34
Checking Table Consistency	10-35
Instructions	10-35
Consistency Check Options	10-36
Overall Table Consistency Option	10-36
Table Allocations Option	10-37
Reindex Option	10-38
Fix Text Option	10-38
Other dbcc Command Option	10-39

Updating Statistics	10-39
---------------------------	-------

11. Using System Management Tools

What's in This Chapter	11-1
Editing the sql.ini File with sqledit	11-1
Starting and Stopping sqledit	11-2
Creating a New sql.ini File	11-3
Creating a sql.ini File Entry	11-3
Updating an Existing sql.ini File	11-5
Modifying a sql.ini File Entry	11-5
Connection Format Examples	11-7
Editing Platform and Net-Library™ Driver Information	11-9
Editing the Platform List	11-10
Editing the Net-Library Driver List	11-11
Testing Whether a SQL Server is Running	11-11
Querying SQL Server with wisql	11-12
Starting and Stopping wisql	11-12
Setting wisql Run Options	11-13
Managing SQL Server Connections	11-14
Connecting	11-14
Switching Between Connections	11-15
Disconnecting	11-15
Creating and Modifying Queries	11-15
Creating a New Query	11-15
Modifying an Existing Query	11-16
Executing Queries	11-16
Saving Query Results	11-16

A. SQL Server Configuration Parameters

What's in This Appendix	A-1
SQL Server 10.x Parameter List	A-1
SQL Server 11.0 Parameter List	A-4
Parameters by Functional Group	A-10
Backup/Recovery Parameters	A-11
Cache Manager Parameters	A-11
Disk I/O Parameters	A-11
General Information Parameter	A-11
Languages Parameters	A-11
Lock Manager Parameters	A-12

Memory Use Parameter	A-12
Network Communications Parameters	A-12
Operating System Resources Parameters	A-12
Physical Memory Parameters	A-13
Processors Parameters	A-13
SQL Server Administration Parameters	A-13
User Environment Parameters	A-14

B. Guide to SQL Server Manager Icons

What's in This Appendix	B-1
List of Icons	B-1
Valid Drag-and-Drop Operations	B-3
SQL Server 11.0 Drag-and-Drop Operations	B-6

Glossary

Index

List of Figures

Figure 2-1:	SQL Server Manager icon.....	2-1
Figure 2-2:	SQL Server Manager Voyager window.....	2-2
Figure 2-3:	Connect dialog box.....	2-4
Figure 2-4:	Voyager showing connected SQL Server icon.....	2-5
Figure 2-5:	Password expired message	2-6
Figure 2-6:	Change Password dialog box.....	2-6
Figure 3-1:	SQL Server Manager Voyager window.....	3-2
Figure 3-2:	Multiple Voyager windows for multiple logins.....	3-4
Figure 3-3:	Object display formats	3-7
Figure 3-4:	Details view of databases, sorted by name and by size.....	3-8
Figure 3-5:	Filter by Name dialog box for filtering icons by name	3-9
Figure 3-6:	Drag-and-drop symbol	3-11
Figure 3-7:	Standard menu bar	3-12
Figure 3-8:	Pull-down menu for a table	3-14
Figure 3-9:	Shortcut menu for a table	3-15
Figure 3-10:	Toolbar and status bar.....	3-16
Figure 3-11:	SQL Server Manager toolbar.....	3-17
Figure 3-12:	ToolTip for the Refresh tool bar button	3-18
Figure 3-13:	Status bar.....	3-19
Figure 3-14:	Properties tab of the multiple-tab Table dialog box.....	3-20
Figure 3-15:	Dependencies tab of the multiple-tab Table dialog box.....	3-21
Figure 3-16:	Navigating to a related object.....	3-22
Figure 3-17:	Example of a command dialog box.....	3-23
Figure 3-18:	Example of a confirmation dialog box.....	3-23
Figure 3-19:	Example of a simple message dialog box.....	3-24
Figure 3-20:	Output dialog box.....	3-24
Figure 3-21:	Methods of refreshing a window or dialog box.....	3-26
Figure 3-22:	Methods of moving between open windows and dialog boxes	3-27
Figure 4-1:	SQL Server Manager window	4-2
Figure 4-2:	Connect dialog box.....	4-2
Figure 4-3:	Connected SQL Server icon in Voyager.....	4-3
Figure 4-4:	Disconnect dialog box.....	4-5
Figure 4-5:	Server Properties dialog box.....	4-6
Figure 4-6:	Configuration Parameters dialog box for 10.x SQL Server	4-10
Figure 4-7:	Configuration Parameters dialog box for 11.0 SQL Server.....	4-12
Figure 4-8:	Server Processes dialog box	4-14
Figure 4-9:	Create Remote Server dialog box	4-17
Figure 4-10:	Remote Server Login Mappings tab	4-20

Figure 4-11:	Remote Server Properties dialog box.....	4-22
Figure 4-12:	Server Generate DDL dialog box	4-24
Figure 4-13:	Server Run DDL dialog box	4-25
Figure 4-14:	Server Consistency Check dialog box.....	4-29
Figure 4-15:	dbcc engine output	4-30
Figure 5-1:	Create Cache dialog box	5-2
Figure 5-2:	Cache Properties tab.....	5-5
Figure 5-3:	Default cache and two named data caches	5-7
Figure 5-4:	Cache Bindings tab	5-11
Figure 6-1:	Logical database devices on a physical disk device	6-2
Figure 6-2:	Create Database Device dialog box.....	6-3
Figure 6-3:	Properties tab of Database Device dialog box	6-6
Figure 6-4:	Databases tab of Database Device dialog box	6-8
Figure 6-5:	Segments tab of Database Device dialog box	6-9
Figure 6-6:	Database device on disk1 mirrored on disk2.....	6-10
Figure 6-7:	Database Device dialog box with disk mirroring enabled	6-12
Figure 6-8:	Disk mirroring enabled.....	6-13
Figure 6-9:	Database device with mirroring disabled	6-14
Figure 7-1:	Create Database dialog box.....	7-3
Figure 7-2:	Database Properties dialog box	7-8
Figure 7-3:	Expanding a database	7-9
Figure 7-4:	Database Allocations dialog box	7-11
Figure 7-5:	Database Generate DDL dialog box.....	7-14
Figure 7-6:	Database Run DDL dialog box	7-15
Figure 7-7:	Database DDL directory and file structure	7-17
Figure 7-8:	Database Options dialog box	7-19
Figure 7-9:	Create Segment dialog box.....	7-24
Figure 7-10:	Properties tab of Segment dialog box	7-26
Figure 7-11:	Contains tab of the Segment dialog box.....	7-28
Figure 7-12:	Thresholds tab of Segment dialog	7-31
Figure 7-13:	Create Dump Device dialog box	7-35
Figure 7-14:	Properties tab of the Dump Device dialog box	7-37
Figure 7-15:	Database Backup dialog box	7-40
Figure 7-16:	Volume Change dialog box	7-43
Figure 7-17:	Database Restore dialog box	7-46
Figure 7-18:	Volume Change dialog box	7-48
Figure 7-19:	Database Consistency Check dialog box.....	7-51
Figure 7-20:	dbcc checkdb output	7-53
Figure 7-21:	dbcc checkalloc output.....	7-54
Figure 7-22:	dbcc checkcatalog output	7-54
Figure 7-23:	Confirmation message: setting a database online.....	7-55

Figure 8-1:	Obtaining access to a database object	8-7
Figure 8-2:	Create Login dialog box.....	8-8
Figure 8-3:	Login Properties tab	8-11
Figure 8-4:	Change Password dialog box.....	8-13
Figure 8-5:	Login Databases Owned tab	8-14
Figure 8-6:	Login Alias/Users tab.....	8-15
Figure 8-7:	Create User dialog box.....	8-18
Figure 8-8:	User Properties tab	8-21
Figure 8-9:	User Objects Owned tab	8-23
Figure 8-10:	User Object Permissions tab.....	8-24
Figure 8-11:	User Command Permissions tab	8-28
Figure 8-12:	Create Group dialog box	8-30
Figure 8-13:	Group Properties tab.....	8-32
Figure 8-14:	Group Properties tab.....	8-34
Figure 8-15:	Group Object Permissions tab	8-36
Figure 8-16:	Group Command Permissions tab.....	8-40
Figure 9-1:	Create dialog box for an index.....	9-2
Figure 9-2:	Confirmation dialog box.....	9-3
Figure 9-3:	Properties tab for an index	9-4
Figure 9-4:	Voyager window showing object hierarchy	9-5
Figure 9-5:	Dependencies tab for a view	9-6
Figure 9-6:	Delete confirmation dialog box	9-7
Figure 9-7:	Create Default dialog box.....	9-9
Figure 9-8:	Creating the statedflt default	9-10
Figure 9-9:	statedflt default as created by SQL Server Manager	9-11
Figure 9-10:	Default Properties tab	9-12
Figure 9-11:	Default Dependencies tab	9-13
Figure 9-12:	Default Bindings tab.....	9-15
Figure 9-13:	Create Index dialog box.....	9-19
Figure 9-14:	Index Properties tab	9-23
Figure 9-15:	Index Segments tab.....	9-25
Figure 9-16:	Index Consistency Check dialog box.....	9-26
Figure 9-17:	Output from Full option of index consistency check	9-28
Figure 9-18:	Create Procedure dialog box.....	9-30
Figure 9-19:	Procedure Properties tab	9-33
Figure 9-20:	Procedure Dependencies tab.....	9-35
Figure 9-21:	Procedure Permissions tab	9-37
Figure 9-22:	Create Rule dialog box.....	9-40
Figure 9-23:	Rule Properties tab	9-42
Figure 9-24:	Rule Dependencies tab.....	9-43
Figure 9-25:	Rule Bindings tab.....	9-45

Figure 9-26:	Create Trigger dialog box	9-49
Figure 9-27:	Creating the newhero trigger	9-50
Figure 9-28:	newhero trigger as created by SQL Server Manager	9-51
Figure 9-29:	Trigger Properties tab	9-52
Figure 9-30:	Trigger Dependencies tab	9-53
Figure 9-31:	Create User Datatype dialog box	9-55
Figure 9-32:	User Datatype Properties tab	9-57
Figure 9-33:	User Datatype Dependencies tab	9-58
Figure 9-34:	Create View dialog box	9-60
Figure 9-35:	Creating the titlepub view	9-62
Figure 9-36:	titlepub view as created by SQL Server Manager	9-63
Figure 9-37:	View Properties tab	9-64
Figure 9-38:	View Dependencies tab	9-65
Figure 9-39:	View Permissions tab	9-67
Figure 10-1:	Create Table dialog box	10-5
Figure 10-2:	Table Column Advanced dialog box	10-9
Figure 10-3:	Table Properties tab	10-12
Figure 10-4:	Table Column Advanced dialog box	10-15
Figure 10-5:	Table Dependencies tab	10-17
Figure 10-6:	Table Permissions tab	10-19
Figure 10-7:	Table Segments tab	10-23
Figure 10-8:	Table Indexes/Triggers tab	10-24
Figure 10-9:	Table Check Constraints tab	10-26
Figure 10-10:	Table Partition tab	10-28
Figure 10-11:	Table Partitions tab	10-29
Figure 10-12:	Table Referential Constraints tab	10-31
Figure 10-13:	Referential Constraints tab for referencing table	10-32
Figure 10-14:	Referential Constraints tab for referenced table	10-33
Figure 10-15:	Table Consistency dialog box	10-35
Figure 10-16:	Table consistency check output—overall consistency option	10-37
Figure 10-17:	Table consistency check output—allocations option	10-38
Figure 10-18:	Table consistency check output—reindex option	10-38
Figure 10-19:	Table consistency check output—fix text option	10-39
Figure 11-1:	SQLEDT window	11-2
Figure 11-2:	Edit Platform & Net-Library Driver dialog box	11-10
Figure 11-3:	Results of pinging the SOL_ENG11 SQL Server	11-11
Figure 11-4:	WISQL window	11-12
Figure 11-5:	WISQL Options dialog box	11-13
Figure 11-6:	Open Connection dialog box	11-14

List of Tables

Table 1:	Style conventions.....	xxvi
Table 2:	Shortcut symbols	xxvi
Table 3:	Displaying online help information.....	xxix
Table 2-1:	Command line options	2-3
Table 2-2:	Columns to add to heroes table	2-30
Table 3-1:	Activities available from standard menus	3-12
Table 3-2:	Shortcut menu commands	3-15
Table 4-1:	Files created during generation of server-level DDL	4-27
Table 4-2:	Files created during generation of database-level DDL.....	4-28
Table 6-1:	Effects of disabling mirroring	6-14
Table 7-1:	Files created during generation of database-level DDL.....	7-17
Table 8-1:	Object permission dialog boxes.....	8-6
Table 8-2:	Privilege and permission summary	8-41
Table 8-3:	Access granted through tabs and dialog boxes.....	8-46
Table 10-1:	Types of table constraints	10-4
Table 11-1:	Commands for online help on system management utilities	11-1
Table B-1:	SQL Server Manager icons	B-1
Table B-2:	Icon drag-and-drop operations.....	B-4
Table B-3:	SQL Server 11.0 icon-drag and-drop operations.....	B-6

Preface

The *Sybase SQL Server Manager User's Guide* describes how to use Sybase SQL Server Manager™. SQL Server Manager is a system administration tool that helps you manage Sybase SQL Server™ and its databases. This manual provides step-by-step instructions for performing each of the activities supported by SQL Server Manager.

Audience

This manual is for anyone who is authorized to perform administrative tasks on SQL Server, or who owns one or more databases. Generally, System Administrators, Database Administrators, and application designers perform the activities described in this manual.

Familiarity with SQL Server administration and some knowledge of the Sybase architecture are assumed.

How to Use This Book

To get started with SQL Server Manager, read:

- Chapter 1, “Introduction,” which gives an overview of SQL Server Manager.
- Chapter 2, “Getting Started,” which describes how to start and stop SQL Server Manager and contains a brief tutorial that walks you through several typical system administration activities.
- Chapter 3, “Introducing SQL Server Manager Windows and Dialog Boxes,” which introduces you to the desktop window and to the various types of dialog boxes used in SQL Server Manager.

You might find it helpful to have SQL Server Manager running while you read, so you can try out the features.

To learn more about how to use SQL Server Manager, read Chapters 4-10, in any order. Each chapter describes how SQL Server Manager helps you perform a set of SQL Server administration tasks:

- Chapter 4, “Managing the SQL Server Operating Environment,” describes how to connect to, configure, and control a SQL Server.
- Chapter 5, “Managing Cache,” describes how to create and bind objects to user-defined caches.

- Chapter 6, “Managing Physical Resources,” describes how to create and modify database and dump devices and how to set up device mirroring.
- Chapter 7, “Managing Databases and Segments,” describes how to create and modify databases and segments and how to set up and monitor the distribution of a database across one or more segments.
- Chapter 8, “Controlling Access,” describes the use of SQL Server logins, database users, database groups, and Sybase administration roles in controlling access to SQL Server and its databases and objects.
- Chapter 9, “Managing Database Objects,” describes how to create and administer the objects contained in a database.
- Chapter 10, “Managing User Tables,” describes how to manage database tables.
- Chapter 11, “Using System Management Tools,” describes how to use the utilities distributed with SQL Server Manager to edit *sql.ini* files and to create and send queries to SQL Server.

To look up reference information, see the appendixes and glossary:

- Appendix A, “SQL Server Configuration Parameters,” gives a brief definition of each SQL Server configuration parameter.
- Appendix B, “Guide to SQL Server Manager Icons,” is a pictorial guide to the icons used in SQL Server Manager to represent objects and utilities.
- The Glossary contains definitions of system administration and SQL Server Manager terms.

Related Documents

- *SQL Server Manager Installation Guide* describes how to install SQL Server Manager.
- *SQL Server Utility Programs* documents the Sybase utility programs, such as *isql*, that you can execute from the operating system command line.
- *Sybase Troubleshooting Guide* contains information that helps you prevent or quickly respond to trouble with SQL Server. It includes topics such as how to determine appropriate settings of configuration variables, how to manage transaction logs, how to

develop good recovery procedures, and how to use disk mirroring effectively.

- *System Administration Guide* describes how to administer and control SQL Server databases independent of system administration tools such as SQL Server Manager. It includes the Transact-SQL commands and system procedures used to perform the functions for which SQL Server Manager provides a graphical interface.
- *SQL Server Reference Manual* provides basic syntax and usage information for every command, function, system procedure, and catalog stored procedure.

Other Sources of Information

Sybase offers a system and database administration class called "SQL Server Administration." For details, contact:

Mail: Education Registrars

Sybase Professional Services
77 South Bedford Street
Burlington, MA 01803

Phone: (800) 8-SYBASE or (617) 564-6970

Fax: (800) 792-2733 or (617) 564-6960

E-mail: registrars@sybase.com

Conventions

The following sections describe the conventions used in this manual.

Style Conventions

In this manual, the following typefaces and fonts have special significance:

Table 1: Style conventions

Example	Used For
procedure	Command keywords, terms that are being defined, and emphasized text
<i>sql.ini</i>	Filenames, variable names, and book titles
ssm	Values that you type exactly as shown

Shortcuts

You can perform any SQL Server Manager activity by executing a drop-down menu command. For most activities, there are shortcuts that help you work more efficiently. For example, SQL Server Manager has toolbars that let you select a button as an alternative to choosing the corresponding command from a drop-down menu.

In the descriptions of how to perform SQL Server Manager operations, this manual first presents the procedure for executing the appropriate menu command. If an alternative method for exists for starting an operation, the manual describes it in a special section, following the basic procedure, titled “Shortcuts.”

Alternative methods of executing a drop-down menu command appear with the symbol shown in the following table:

Table 2: Shortcut symbols





Symbol	Action
	Double-click an icon

Table 2: Shortcut symbols (continued)

Symbol	Action
	Drag one of the icons shown and drop it on the other
	Select the toolbar button shown
	Execute a shortcut menu command

Privileges

Many activities in SQL Server Manager are restricted to users with specific Sybase roles or assignments, such as System Administrator, System Security Officer, System Operator, or database owner. When an activity has such restrictions, the manual indicates them at the beginning of the activity's description. The following example illustrates:

When you change ownership of a database, the new owner must have a SQL Server login but must not be a user of the database or have an alias in it.

Privileges	Only a System Administrator can transfer database ownership.
------------	--

► Note

Because a login with the System Administrator role can perform almost all SQL Server Manager activities, the "Privileges" notes include the System Administrator only when a System Administrator is the sole person who can perform the activity. The notes also indicate those activities that the System Administrator role **cannot** perform.

Mouse Buttons

To use SQL Server Manager, you need a pointing device such as a mouse.

This manual refers to mouse buttons as the left and right buttons, assuming the mouse has two buttons and is configured for right-hand use. On a mouse configured for left-hand use, the positions are reversed.

Screen Images


SQL Server Manager runs on Windows 3.x, Windows NT 3.51, and Windows 95. In each environment, windows and dialog boxes have a somewhat different look, although their contents are identical. For consistency, this manual displays screen images of SQL Server Manager running on a single platform, Windows NT 3.51.

SQL Server 11.0 Features

Some dialog boxes support features that are available only when you are connected to SQL Server at release 11.0:

- When you are connected to a release 10.x SQL Server:
 - Dialog boxes unique to SQL Server release 11.0 are inaccessible.
 - In dialog boxes with controls that support release 10.x and 11.0 features, those controls are invisible.
- When you are connected to a release 11.0 SQL Server, the full range of 10.x and 11.0 features is available.

In this manual, descriptions of release 11.0 features are identified with the symbol shown in the following example:

-  address lock spinlock ratio specifies the number of rows in the address locks hash table protected by one spinlock (rows per spinlock).


If You Need Help

Get help for using SQL Server Manager from the printed documentation, online help, and the Technical Support Center.

Using Online Help

Online help is available when you are using SQL Server Manager. The following table shows how to access the various forms of help.

Table 3: Displaying online help information

To Display This Type of Online Help	Do This
Information about the currently active dialog box.	Click the Help command button on the dialog box.
Details about the desktop window.	Click the Help button on the toolbar: 
Table of Contents for SQL Server Manager help. Gives access to all online help topics.	Choose Contents from the Help menu on the menu bar of the desktop window.
Help on <code>sqledit</code> or <code>wisql</code> utilities that come with SQL Server Manager.	Choose Interfaces... or ISQL... from the Help menu on the menu bar of the desktop window.
Information on how to use the Windows help facility.	Choose Using Help from the Help menu on the menu bar of the desktop window.
Online help search dialog box. Allows search for help topics based on topic string.	Choose Search from the Help menu on the menu bar of the desktop window.
About... dialog box for SQL Server Manager. Displays product version number.	Choose About SQL Server Manager... from the Help menu on the menu bar of the desktop window.
Brief description of a highlighted menu command or icon or of a pressed toolbar button.	See the status bar at bottom of the window.

Contacting Technical Support

Each Sybase installation site has one person (or more) designated to contact the Technical Support Center. If you cannot solve a problem using the manuals or online help, ask the designated person at your site to contact the Technical Support Center for help.

1

Introduction

About SQL Server Manager

SQL Server Manager is a graphical system administration and database administration tool for SQL Server. Its powerful collection of features and visual user interface make exacting administrative tasks easy to perform.

Performing System and Database Administration Tasks

SQL Server Manager helps with the following typical system or database administrator's tasks:

- Managing SQL Servers:
 - Connecting to, disconnecting from, configuring, and stopping servers
 - Displaying and terminating SQL Server processes
 - Setting up *sql.ini* file entries
 - Generating and running server Data Definition Language (DDL) scripts
 - Enabling and controlling access to or from remote servers
 - Troubleshooting SQL Server problems
- Managing data caches:
 - Creating user-defined data caches
 - Defining buffer pools
 - Binding databases and objects to a cache
- Managing SQL Server physical resources:
 - Creating and deleting database devices and dump devices
 - Creating mirror devices
- Managing databases:
 - Creating and deleting databases and database objects
 - Setting database options
 - Managing database storage allocation across devices and segments

- Backing up and restoring databases
- Issuing database checkpoints
- Generating and executing database DDL scripts
- Running the dbcc database checking and repair utility
- Managing access:
 - Creating and deleting SQL Server logins
 - Creating and deleting database users and user groups
 - Administering Sybase roles
 - Managing object and command permissions

Making It Easy

The aim of SQL Server Manager is to provide the power to easily perform complex administration tasks. Some special features that support this aim are:

- **Visual representation of objects.** With SQL Server Manager, you can easily see the databases and logins in each SQL Server and the objects in each database. The display in the Voyager window, representing SQL Servers and their objects, expands and contracts to display information at the level of detail you need.
- **Navigation between related objects.** To get more information about a database object related to the one whose dialog box you are displaying, you do not need to go back to the Voyager window to find and open the icon for the object. You can navigate directly through the displayed object's dialog box to the related object.

For example, to open the dialog box for a trigger defined on a table, just double-click the trigger name in the list where it appears in the table dialog box.

- **Drag and drop operations.** You can execute all SQL Server Manager operations with menu commands. In addition, for some frequently performed operations, you can drag one icon on top of another.

For example, to add a user to a group, drag the icon representing the user over the icon representing the group, and click the confirmation button on the dialog box that opens.

- **The ability to filter the display of objects.** SQL Server Manager allows you to limit the display to objects that match a specified string of characters.
- **A toolbar button for frequently performed tasks.** As an alternative to executing menu commands, click a button in the toolbar to perform such tasks as connecting to or disconnecting from a SQL Server, creating or deleting an object, or refreshing the current display of SQL Server information.

2

Getting Started

What's in This Chapter

This chapter describes what you need to know to use SQL Server Manager by providing the following:

- Instructions on starting and stopping the application
- A tutorial that walks you through some of the common activities you can use SQL Server Manager to perform

If you are new to SQL Server Manager, the tutorial in this chapter will be especially helpful. It is most effective if you have SQL Server Manager running as you proceed.

Starting SQL Server Manager

Start SQL Server Manager with the application icon or with the Run... command.

Starting with the SQL Server Manager Icon

To start, double-click the SQL Server Manager icon on the desktop



Figure 2-1: SQL Server Manager icon

The Voyager window of SQL Server Manager opens. The Voyager window is the primary window in SQL Server. It allows you to see and operate on the SQL Servers you administer and the objects in those SQL Servers. Voyager represents SQL Servers and their objects in a hierarchical format that you can expand or contract, as necessary.

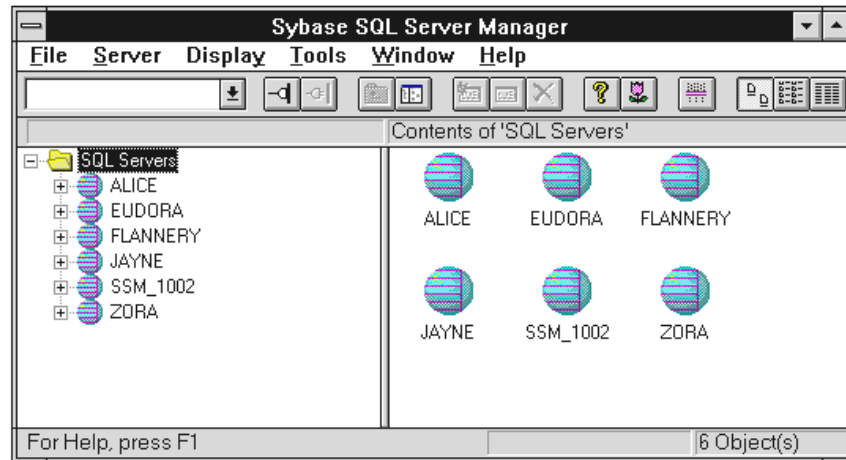


Figure 2-2: SQL Server Manager Voyager window

Starting with the Run... Command

In Windows 3.1x and Windows NT, you can start SQL Server Manager by using the Run... command of the File menu in Program Manager or File Manager. In Windows 95, you can use the Run... command of the Start menu.

To start SQL Server Manager with the Run... command, enter the `ssm` executable name along with its full path, followed by the standard options appropriate to your environment.

Command Line Options

The command line options are single letters and follow the standard for `isql` options. Precede each option letter with a hyphen (-). The option letters are not case-sensitive, and a space between the letter and the argument is allowed. For example, if the SQL Server Manager executable is located in the `c:\sql10_nt\bin` directory, you could enter:

```
c:\sql10_nt\bin ssm -S engineering -U sa
-P mypassword
```

The following table describes each option. The S (server), U (username), and P (password) options are required under most circumstances.

Table 2-1: Command line options

Option	Argument	Description
P	<i>password</i>	Password for connecting to SQL Server.
S	<i>server</i>	Name of SQL Server you want to connect to. This argument is required, as SQL Server Manager does not use the DSQUERY environment variable.
U	<i>username</i>	Login name SQL Server Manager uses for connecting to SQL Server.

The completeness of the information you enter on the command line determines how SQL Server Manager responds:

- If the information you enter is complete enough to allow SQL Server Manager to connect to the specified SQL Server, SQL Server Manager starts, the Voyager window opens, and the icon for the SQL Server changes to a connected SQL Server icon.
- If you specify a valid SQL Server with the S option, but SQL Server Manager needs additional information to complete the connection, SQL Server Manager starts and displays the Connect dialog box to collect the missing information. The dialog box already contains the information specified on the command line or the default from the environment. To complete the connection, you must supply the missing data.
- If you do not give a value for the S option, SQL Server Manager starts and opens the Voyager window with no SQL Servers connected. You must then connect to a SQL Server, as described in “Connecting to SQL Server.”

Connecting to SQL Server

Before performing system administration activities, you must connect to a SQL Server. To connect:

1. In the right pane of the Voyager window, select the icon for the SQL Server you want, and choose **Connect...** from the **Server** menu. The **Connect** dialog box opens.

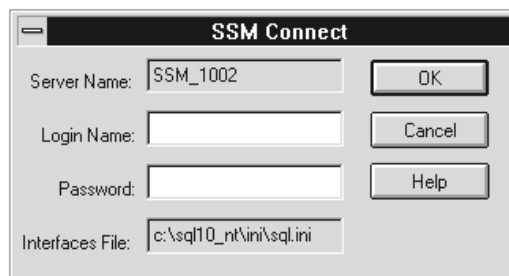


Figure 2-3: Connect dialog box

2. If the **Server Name** edit box is blank, or to change to a different SQL Server, select the SQL Server to connect to from the **Server Name** list. (The box is blank only if you chose the **Connect...** menu command or the **Connect** toolbar button without previously highlighting a SQL Server icon.)
3. Enter the login name with which to connect to the SQL Server.
4. Enter the password for the login.

► **Note**

If the login password has expired, SQL Server Manager displays the **Change Password** dialog box so you can enter a new password. For details, see “**Changing an Expired Password**” on page 2-6.

Click **OK**. If the connection is successful, SQL Server Manager changes the SQL Server icon to a connected SQL Server icon. For tips on diagnosing an unsuccessful connection, see “**Diagnosing an Unsuccessful Connection**” on page 2-5.

To exit without connecting to SQL Server, click **Cancel**.

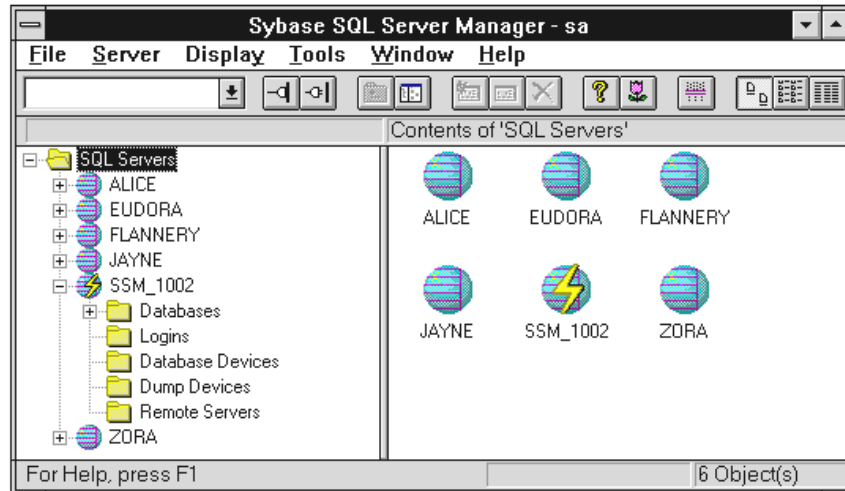
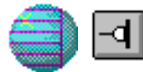


Figure 2-4: Voyager showing connected SQL Server icon

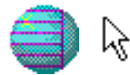
Shortcuts



Select the desired SQL Server icon in the right pane of Voyager, and click the Connect toolbar button.



Click the SQL Server icon or the plus button next to the SQL Server icon in the left pane of Voyager.



Double-click the desired SQL Server icon.

Diagnosing an Unsuccessful Connection

If the connection is not successful, try these techniques:

- Check your login name and password to be sure they are correct.
- Use the `sqledit` utility by double-clicking its icon on the desktop. From `sqledit`, select the Ping... button to invoke the ping utility. ping

tests whether a specified SQL Server can be reached in the current configuration.

For information on `sqledit` and on the format of `sql.ini` file entries, see Chapter 11, “Using System Management Tools.”

Changing an Expired Password

If password expiration is enabled for the SQL Server to which you are attempting to connect, you cannot connect to SQL Server with an expired password.

If SQL Server Manager detects an expired password on an attempt to connect, it displays a notification message.



Figure 2-5: Password expired message

If this message appears, click OK to display the Change Password dialog box, which allows you to enter a new password for the login.

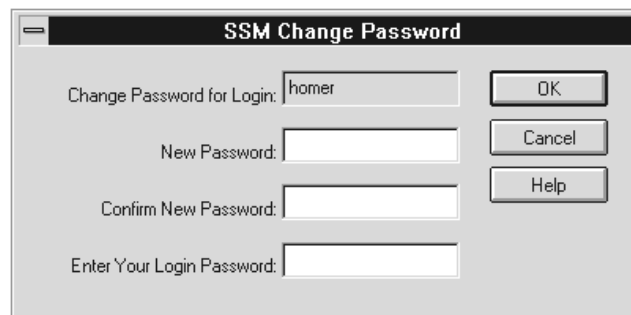


Figure 2-6: Change Password dialog box

To create a new password:

1. Enter a new password for the login, at least six characters long, in the New Password box.
2. Re-enter the new password in the Confirm New Password box.
3. Enter your expired login password in the Enter Your Current Password box.
4. Click OK.
5. Complete the required information in the Connect dialog box, using the new password for the login, and click OK.

Exiting SQL Server Manager

To exit SQL Server Manager, choose Exit from the File menu.

SQL Server Manager disconnects from all connected SQL Servers and closes all open application windows.

Tutorial

The following three-lesson tutorial walks you through typical system administration activities you can perform with SQL Server Manager:

- “Lesson 1: General Activities” on page 2-8 shows how to use some of the special features of the SQL Server Manager graphical user interface. In this lesson, you learn how to display and hide toolbars and icons, how to navigate Voyager, how to navigate between objects, and how to open another Voyager window.
- “Lesson 2: SQL Server Activities” on page 2-19 shows how to create a SQL Server login and database.
- “Lesson 3: Database Activities” on page 2-25 shows how to generate SQL Server DDL, how to create a database user and a database table, and how to assign permissions on the table. In this lesson, you also delete the objects you added in the course of the tutorial.

You can do Lesson 1 at any time; however, you must do Lesson 2 and Lesson 3 in order. To do Lesson 1, you must have the *pubs2* sample database loaded on the SQL Server to which you are connected.

Before You Begin

Before starting any lesson, start SQL Server Manager and connect to a SQL Server, as described in “Starting SQL Server Manager” on page 2-1.

► **Note**

Because you will be adding objects temporarily to the SQL Server to which you connect, use a nonproduction SQL Server for these lessons.

Lesson 1: General Activities

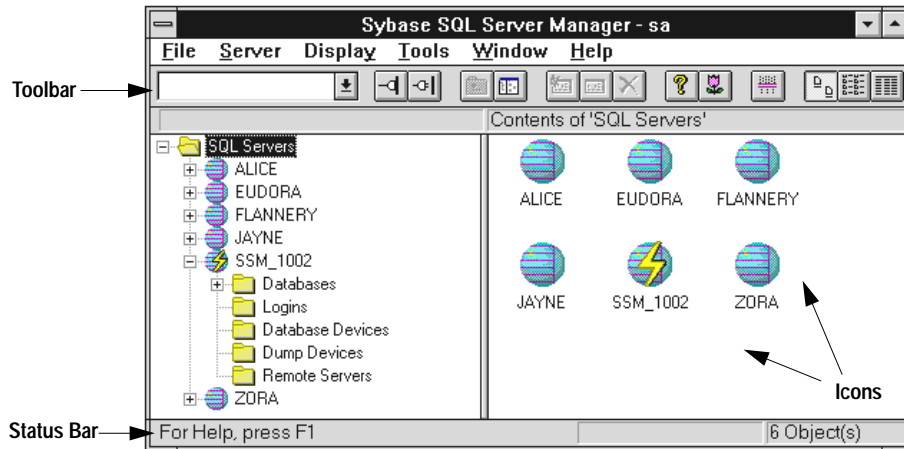
In this lesson, you experiment with several user interface features of SQL Server Manager. Before starting this lesson, make sure that the SQL Server to which you are connected has the *pubs2* sample database installed. *pubs2* is included with Sybase SQL Server.

Part 1: Start SQL Server Manager and Connect to SQL Server

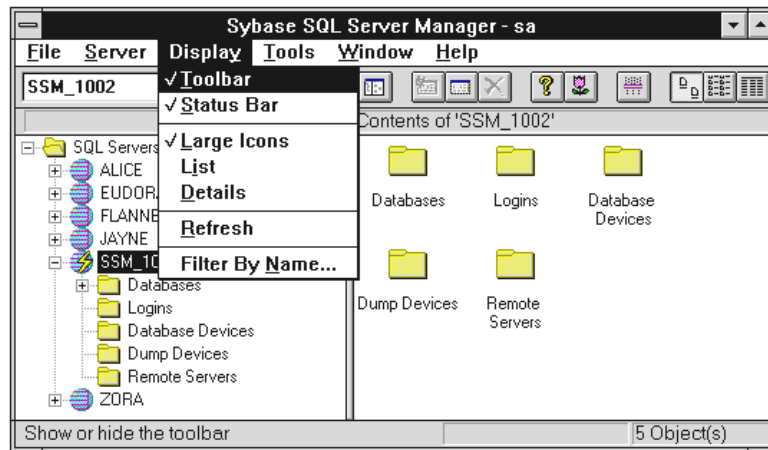
Start SQL Server Manager and connect to SQL Server as described in “Starting SQL Server Manager” on page 2-1. When connecting to SQL Server, use the “sa” login and its password on the SQL Server you use for the activities in the tutorial.

Part 2: Customize the SQL Server Manager Display

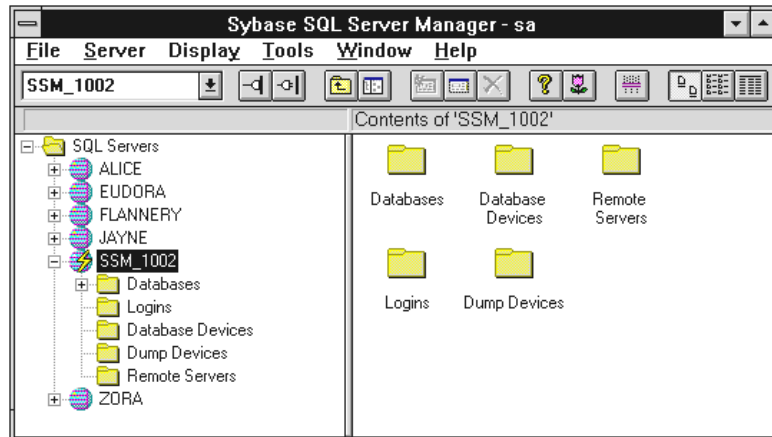
This part shows how to customize the appearance of your SQL Server Manager workspace by controlling the display of toolbars and icons. Note the default position of the toolbar, status bar, and icons.



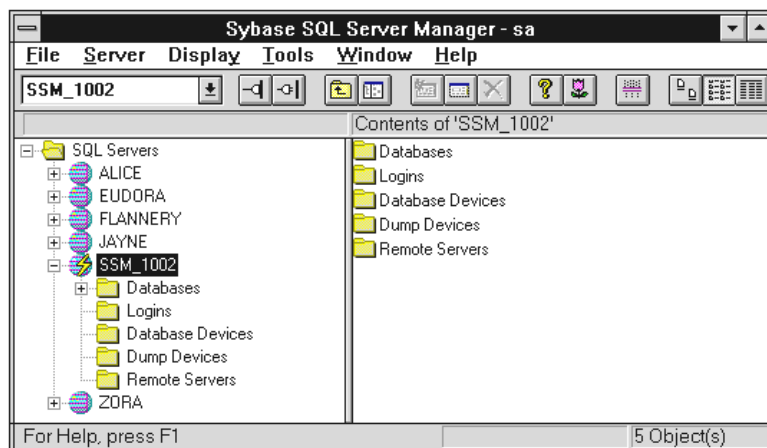
1. Select the Display menu and note that both the toolbar and status bar have a check mark (✓) next to them, indicating that they are visible:



- Choose the Status Bar command. The status bar disappears from the window.



- Select the Display menu again and note that the Status Bar command no longer has a check mark, indicating that the status bar is hidden. Choose the Status Bar command again and note the reappearance of the Status bar in the window. Try this with the toolbar. As you use SQL Server Manager, you can decide whether to display or hide the toolbar and status bar.
- In the default window display, the icons on the right pane of the window are in large icon format. To see an alternative icon view, choose List from the Display menu. Note the changed appearance of the icons in the right pane.



- For a quick way to change the icon display, click one of the three rightmost toolbar buttons.

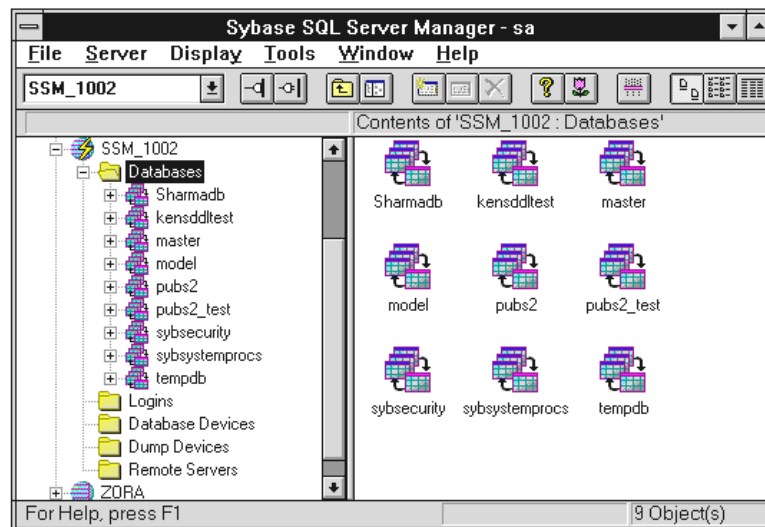


Note how the display changes with each selection. As you use SQL Server Manager, you can decide on the most convenient way to display the contents of the window's right pane.

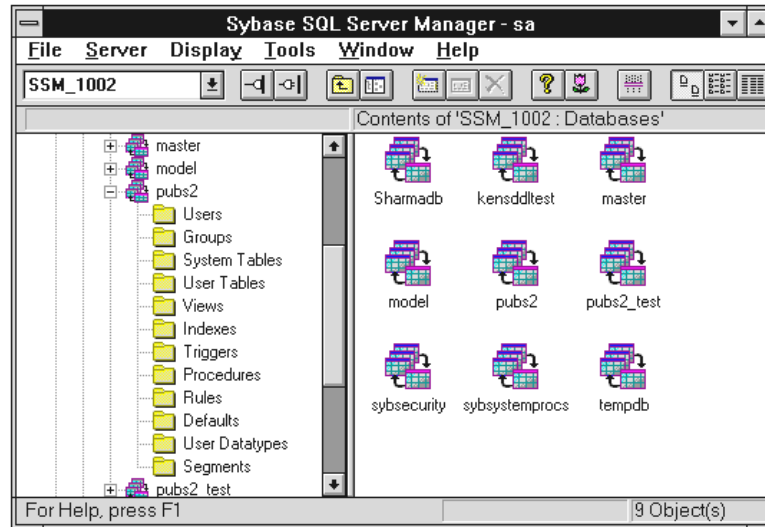
Part 3: Navigate Voyager

This part shows how to expand and collapse Voyager to display and operate on the part of the object hierarchy you need.

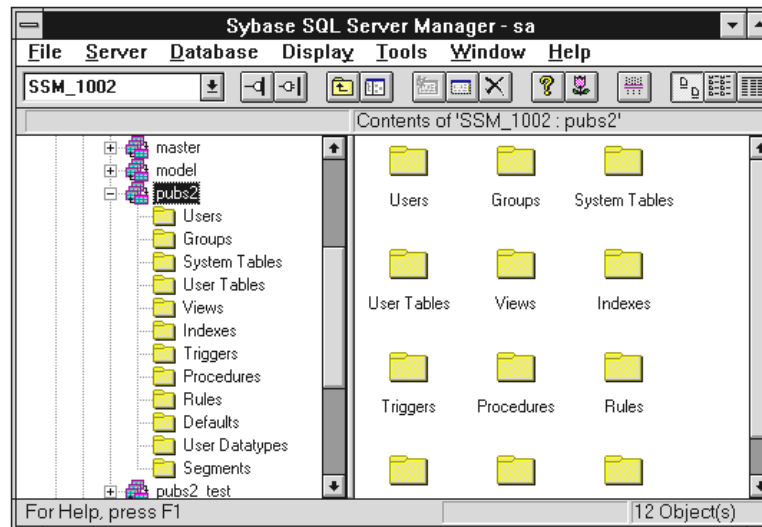
- In the right pane, double-click the Databases icon. The list in the left pane expands to show the databases defined for the SQL Server, and the icon list in the right pane displays an icon for each database. The Contents bar over the right pane changes to show the new context.



2. Now expand Voyager for the *pubs2* database by clicking the plus button next to the *pubs2* icon in the left pane.

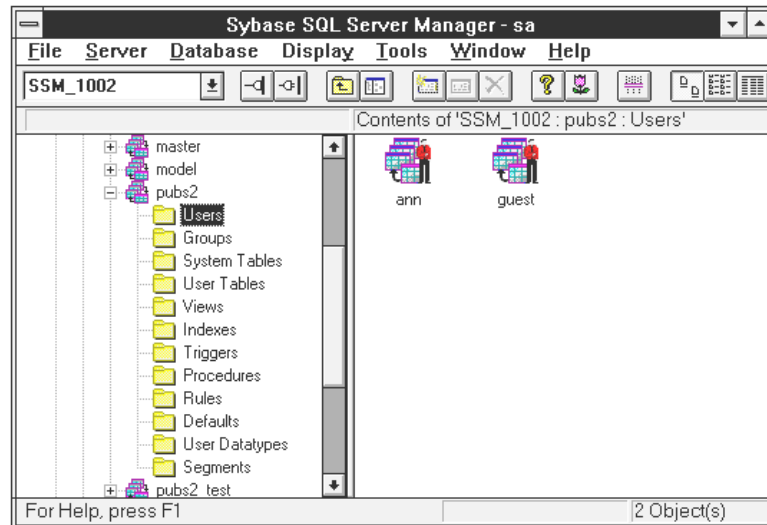


3. Click the *pubs2* icon in the left pane, or double-click the *pubs2* icon in the right pane. Note how the display changes to show the types of objects defined in *pubs2*.

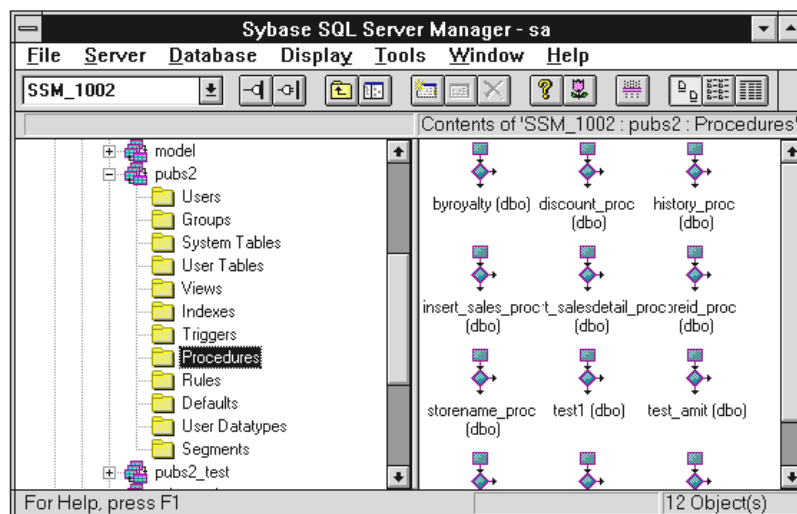


4. To display an example of Voyager expanded to its most detailed level, single-click the Users icon in the left pane, or double-click

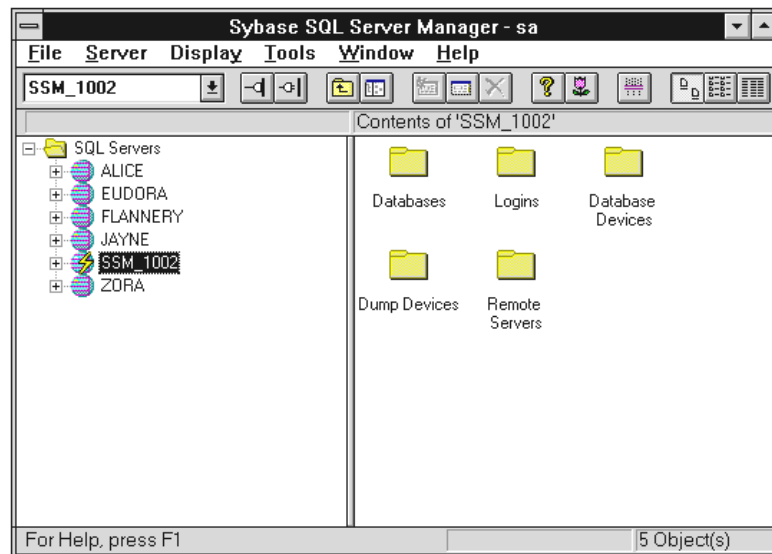
the Users icon in the right pane. The list of icons in the right pane shows an icon for each user defined in *pubs2*.



Now, display the same view for procedures by clicking the Procedures icon in the left pane, and note how the list of icons in the right pane changes to a list of the procedures in *pubs2*.



- To collapse Voyager so it displays a less detailed view, click the minus button next to the Databases icon in the left pane. Now, collapse the list to the SQL Server level by clicking the minus button next to the SQL Server to which you are connected.



Experiment with moving through the levels of Voyager by clicking the plus and minus buttons in the left pane and by double-clicking the icons in the left and right panes. Note how the appearance of the SQL Server Manager window changes in each case to reflect your current context.

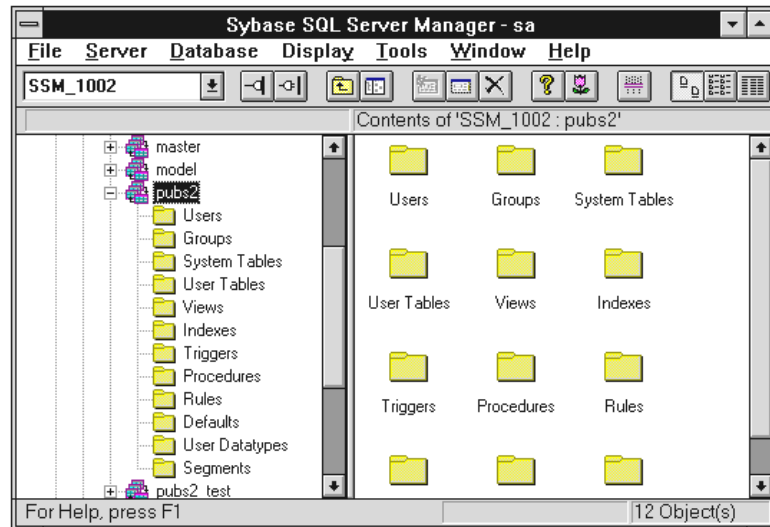
Part 4: Navigate Between Objects

Use this part to try out the SQL Server Manager object navigation feature. Start with the Voyager display collapsed to show only a list of SQL Servers. If the display shows more than SQL Server icons, click the Up button on the toolbar one or more times until the window is at the top level of the hierarchy:

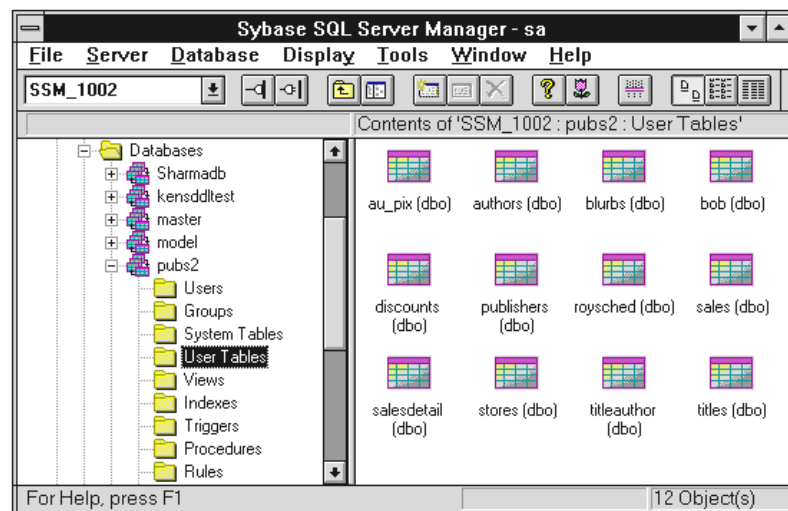


- In succession, click the plus buttons next to the icon for the SQL Server to which you are connected, the Databases icon, and the icon for the *pubs2* database. The hierarchy expands to show the

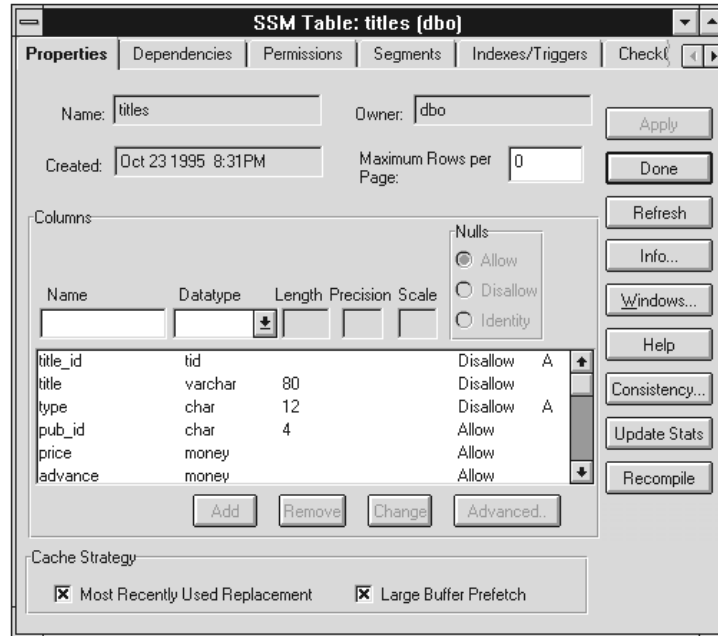
icons that represent *pubs2* tables, defaults, indexes, rules, triggers, users, groups, segments, procedures, user datatypes, and views.



2. In the left pane, single-click the User Tables icon, or in the right pane, double-click it. The list in the right pane changes to a list of *pubs2* tables.

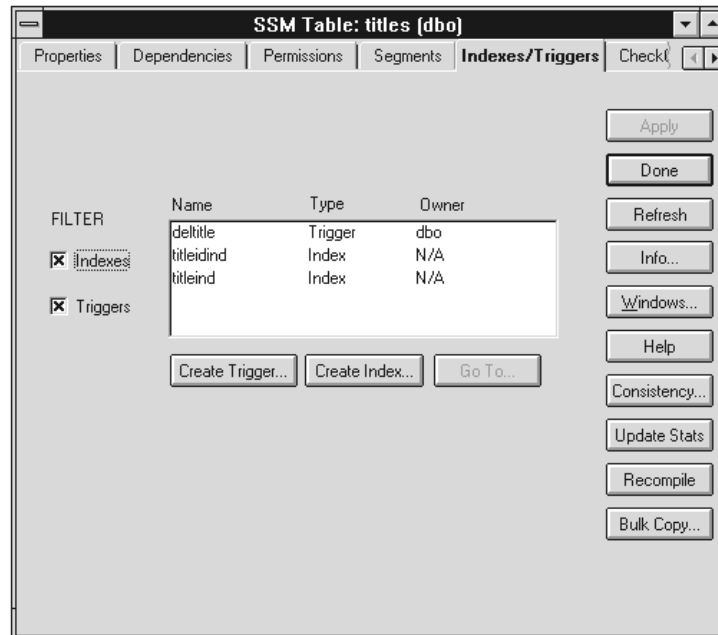


3. Double-click the icon for the *titles(dbo)* table. The Table dialog box opens, showing information about table properties on the Properties tab.

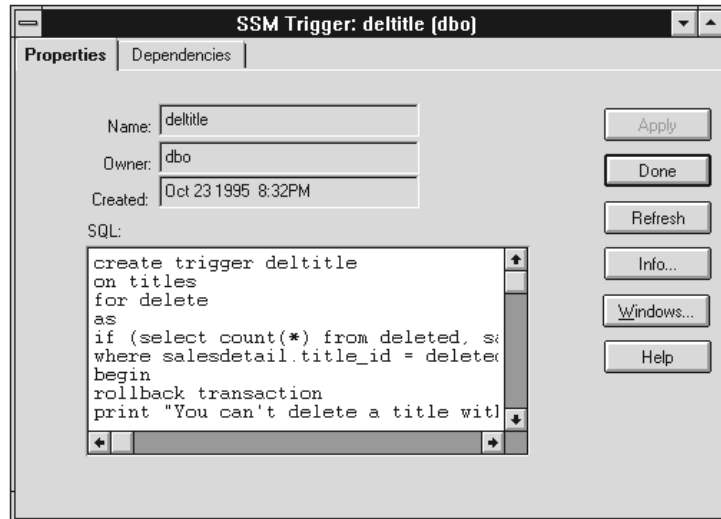


4. To see the various types of information on the *titles* table, select each tab button that appears on the top of the dialog box. The Dependencies, Permissions, Segments, Indexes/Triggers, Check Constraints, Referential Constraints, and Partitions (in SQL Server 11.0) tabs appear.

When you display the Indexes/Triggers tab, note the *deltitle* trigger and the two indexes defined for the *titles* table.



5. Select the *deltitle* row in the list by clicking it; then, click the Go To... button. The dialog box for the *deltitle* trigger opens, showing its Properties tab.

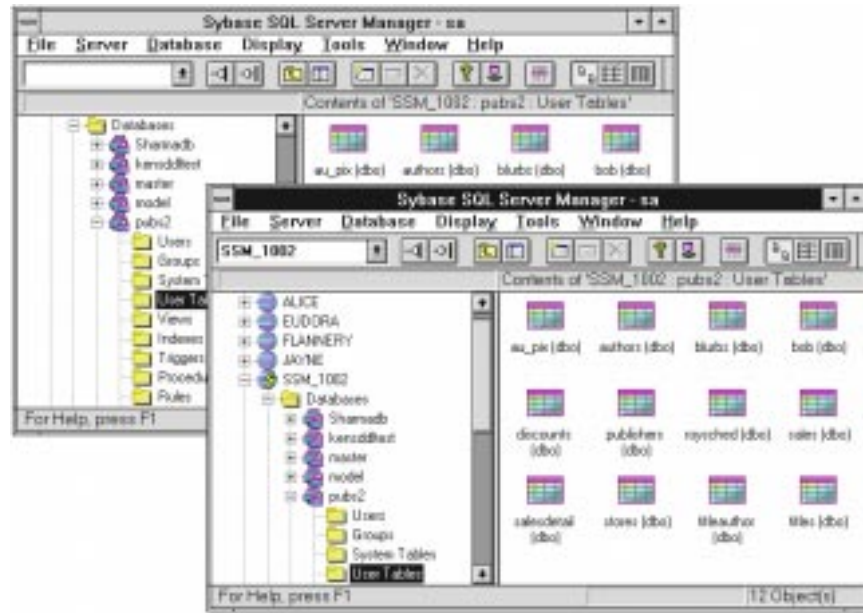


6. Select the Indexes/Triggers tab of the *titles* table dialog box to make it active again. Move back and forth between the open dialog boxes.
7. Close both dialog boxes by clicking the Done button on each.

Part 5: Open Another Voyager Window

You may sometimes want to display more than one copy of the Voyager window. For example, when comparing the contents of two databases on the same SQL Server, you may need to open two windows to ensure that the details you want to see are not scrolled out of range.

Choose New Window from the Window menu. A second copy of the Voyager window opens.



At this point, you can exit SQL Server Manager or go on to the next lesson. To exit SQL Server Manager, choose Exit from the File menu.

Lesson 2: SQL Server Activities

In this lesson, you create SQL Server logins and a user database, activities you may perform frequently as a SQL Server Administrator.

Part 1: Start SQL Server Manager

Start SQL Server Manager and connect to SQL Server as described in “Starting SQL Server Manager” on page 2-1. When connecting to SQL Server, use the “sa” login and its password on the SQL Server you use for the activities in the tutorial.

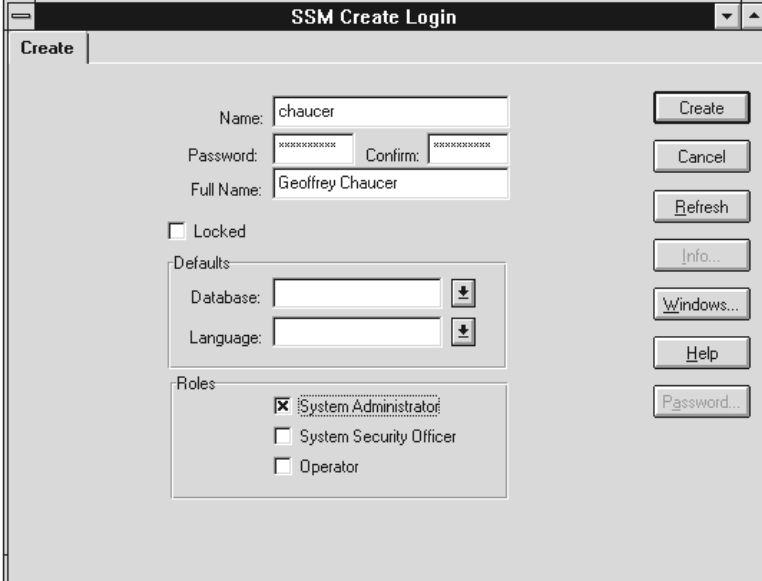
Part 2: Create SQL Server Logins

In this part, you create three SQL Server logins, “chaucer,” “homer,” and “virgil”:

1. From the Server menu, choose Create; then, choose Login from the cascading menu. The Create Login dialog box opens.

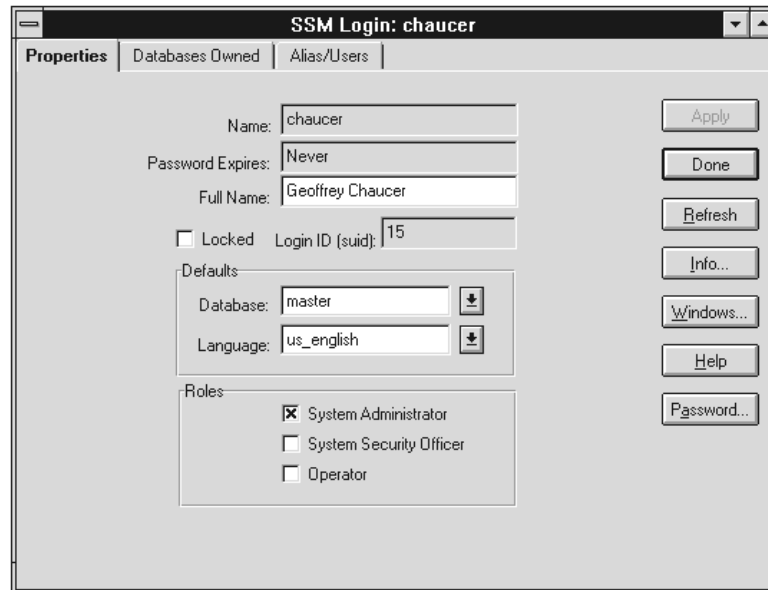


2. Enter the following information:
 - For Name, type
chaucer
 - For Password, type
canterbury
 - Retype the password in the Confirm box.
 - For Full Name, type
Geoffrey Chaucer
3. Ignore the Locked check box and the Defaults group box.
4. Assign the System Administrator role to the new login by selecting the System Administrator role in the Roles box. An “X” appears beside the role.



The screenshot shows the 'SSM Create Login' dialog box with the 'Create' tab active. The 'Name' field is filled with 'chaucer', the 'Password' and 'Confirm' fields are masked with asterisks, and the 'Full Name' field contains 'Geoffrey Chaucer'. The 'Locked' checkbox is unchecked. The 'Defaults' section includes 'Database' and 'Language' dropdown menus. The 'Roles' section shows 'System Administrator' selected with an 'X' in a box, while 'System Security Officer' and 'Operator' are not selected. On the right side, there are buttons for 'Create', 'Cancel', 'Refresh', 'Info...', 'Windows...', 'Help', and 'Password...'.

5. Click the Create button. The Create Login dialog box changes to the Login Properties tab to allow modification of the login. When you expand Voyager to display SQL Server objects, a new Login icon, named “chaucer,” is included in the list.



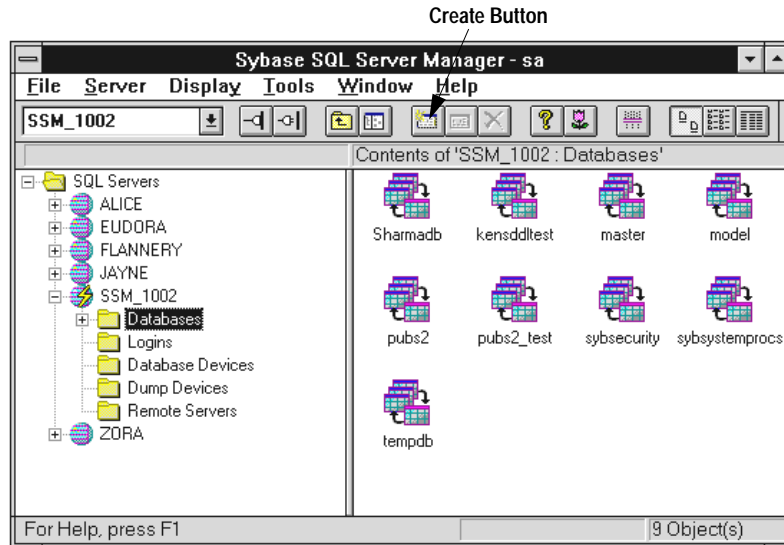
6. Dismiss the Login Properties tab for “chaucer” by clicking the Done button.
7. Now create two more logins by repeating Steps 1 through 5 of Part 2 and using the following data:

	Name	Password	Full Name
Second login	homer	achilles	Homer the Bard
Third login	virgil	aeneas	Virgil the Poet

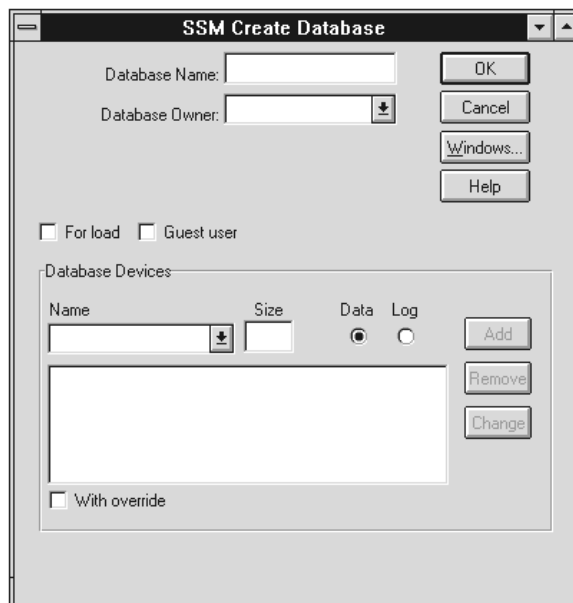
Part 3: Create a Database

In creating logins, you used the Create menu command. To create a database, use the Create toolbar button. Start by clicking the databases icon in the left pane of the Voyager window.

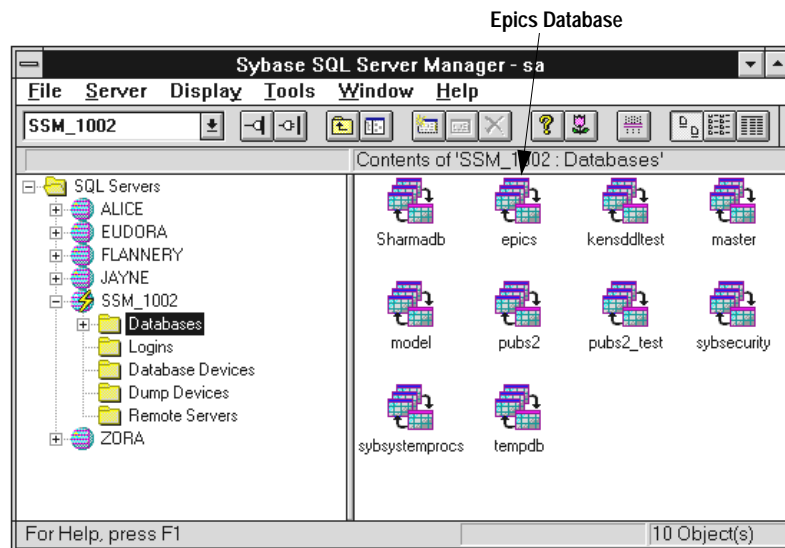
1. Click the Create button on the toolbar.



2. The Create Database dialog box opens:



3. Enter the following information:
 - For Database Name, type:
epics
 - For Database Owner, type the name of the first login you created:
chaucer
4. Leave the Database Devices fields blank, indicating that the database is to be created on a default device, with the default size, in your environment.
5. Click the OK button. Note that a new Database icon, named *epics*, is included in the list in the right pane.



Part 4: Exit

In this lesson, you created a SQL Server login and a database owned by that login. Go to Lesson 3 now or stop here, exit SQL Server Manager, and come back later.

To exit SQL Server Manager, choose Exit from the File menu.

Lesson 3: Database Activities

In this lesson, you can experiment with tasks you may perform frequently as a database administrator. You will:

- Create two database users and a database table in the *epics* database you created in Lesson 2.
- Assign permissions on the table.
- Generate the database definition language (DDL) to create the schema definition of the *epics* database.

Part 1: Start SQL Server Manager

If you left SQL Server Manager between Lesson 2 and Lesson 3, start SQL Server Manager and connect to SQL Server as described in “Starting SQL Server Manager” on page 2-1. When connecting to SQL Server, use the “sa” login and its password on the SQL Server you use for the activities in the tutorial.

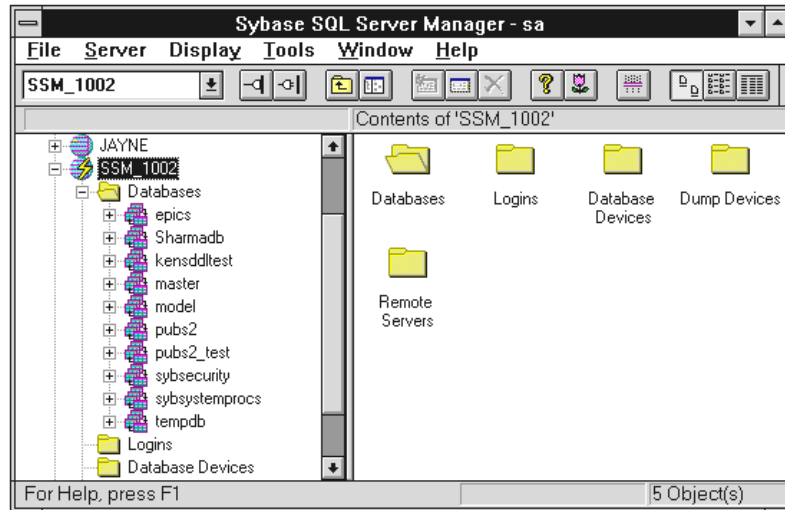
Part 2: Create Database Users

In this part, create database users mapped to the “homer” and “virgil” logins you created in Lesson 2. You can do this in several ways:

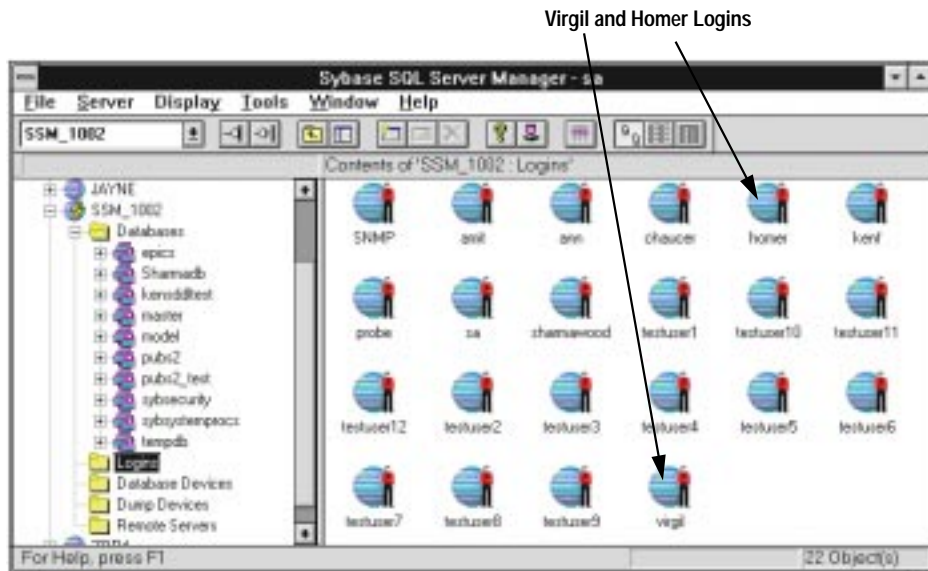
- By selecting the icon for the *epics* database, choosing Create from the Database menu, and choosing User from the Create menu.
- By selecting the icon for the *epics* database, displaying its shortcut menu, and choosing the Create and User commands.
- By expanding the icon for the *epics* database, clicking the Users container icon, and clicking the Create toolbar button.
- By expanding the icon for the *epics* database, displaying the shortcut menu for the Users container icon, and choosing the Create User command.
- By dragging a login icon onto the *epics* database icon.

In earlier parts of the tutorial, you used a Create menu command and the Create toolbar button. In this part, use the icon drag-and-drop method.

1. Expand the Voyager display so that the left pane lists each database on the SQL Server. To do this, click the plus buttons for the SQL Server and the Databases icon.



2. Single-click the Logins icon in the left pane, or double-click the Logins icon in the right pane. The right pane displays the currently defined logins, including "homer" and "virgil."

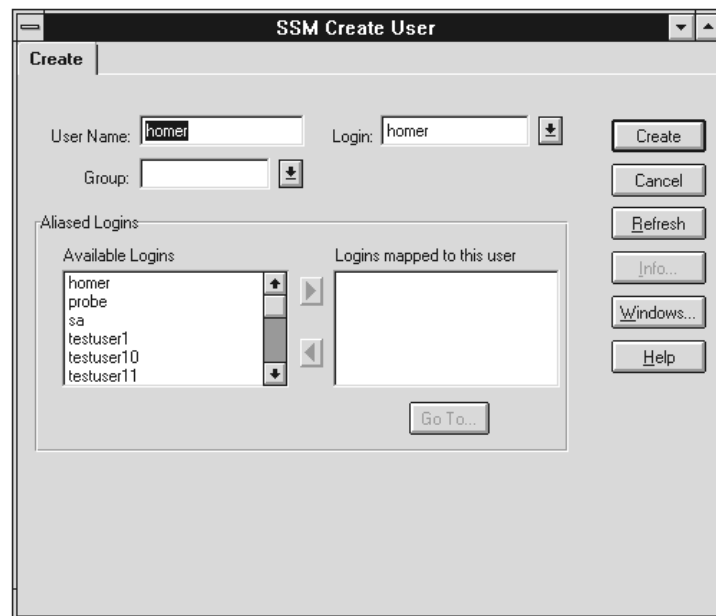


3. If the *epics* database icon is scrolled out of view when you select the “homer” or “virgil” icon, open a new window by clicking the New Window toolbar button; otherwise, go to Step 4.



4. Find the “homer” login icon, and drag and drop it onto the *epics* database icon in the left pane. If you opened a new Voyager window in Step 3, before dragging and dropping the login icon, scroll the new window display so the *epics* database icon is visible.

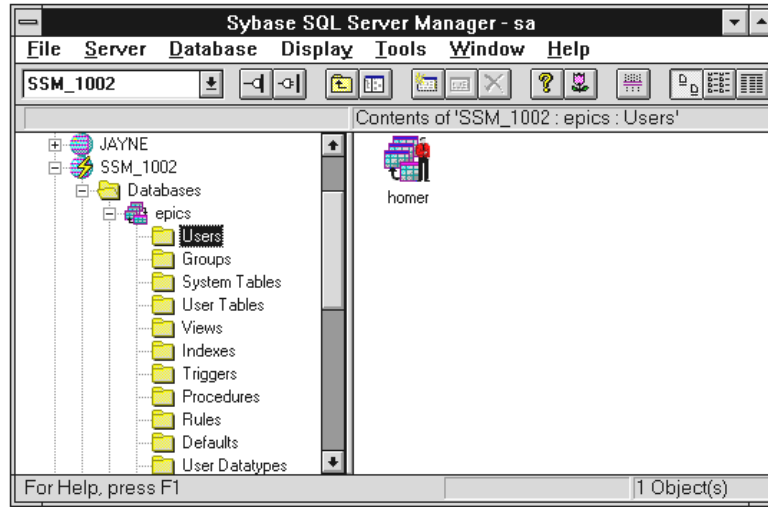
The Create User dialog box opens.



In an icon drag-and-drop operation, SQL Server Manager fills in required information based on the characteristics of the objects whose icons you choose. Therefore, you do not have to provide any further information.

5. You can change or supplement the data in the Create dialog box if you choose. However, for this tutorial, take the simple route and click the Create button.

6. Click the Done button to close the dialog box.
7. Expand Voyager for the *epics* database by double-clicking its icon. Double-click the Users icon. The list should now contain a new user icon with the name “homer”.

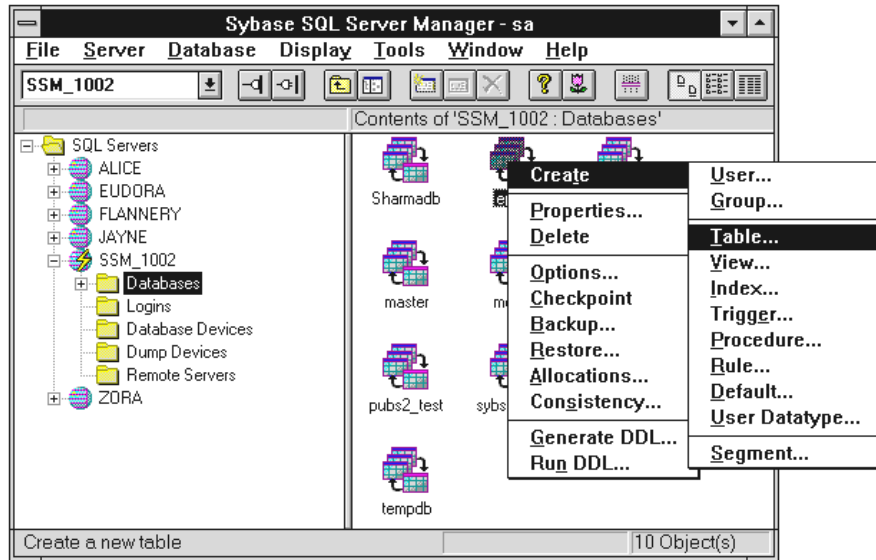


8. Create a second user called “virgil” by dragging and dropping the “virgil” login onto the *epics* icon. Repeat Steps 5 through 7. A second user icon with the name “virgil” appears. (In Step 5, you can accept the default values by pressing the Enter key instead of clicking the Create button.)

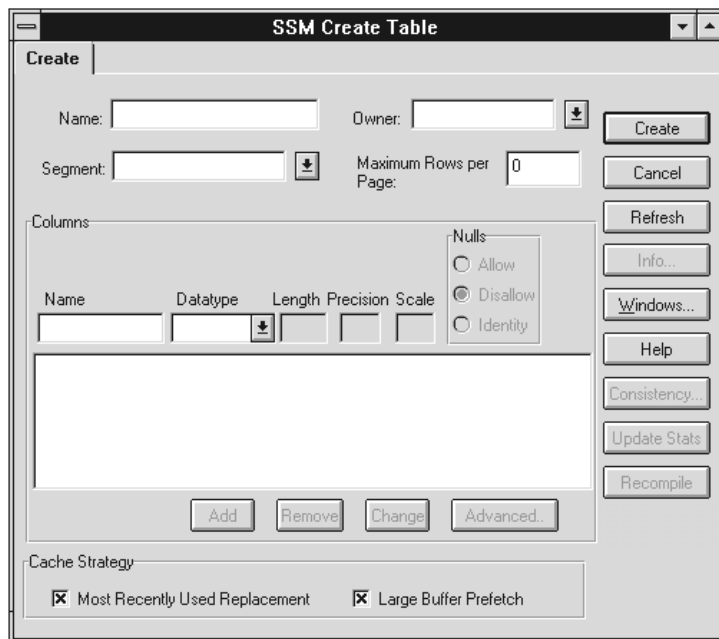
Part 3: Create a Database Table

Use this part to create a database table in the *epics* database. Start with the *epics* database icon visible.

1. Press and release the right mouse button over the *epics* database icon, and choose Create from the shortcut menu; then, choose Table from the cascading menu.



The Create Table dialog box opens.



2. Enter the following information:

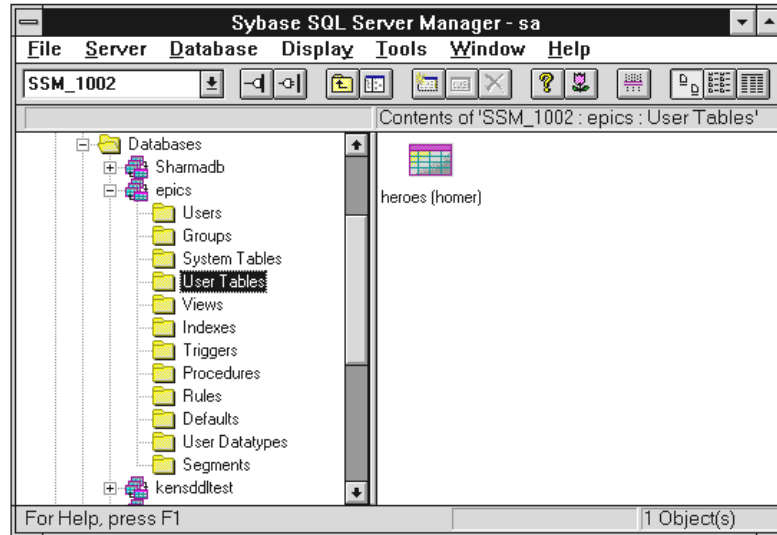
- For Name, type:
heroes
- For Owner, type or select the name of the database owner:
homer
- For Segment, type or select:
default
- In the Columns group box, enter the table's column definitions. For each column, type the data in the boxes directly over the multi-column list box, and then click the Add button. As you add each column definition, the information appears in the list box. If you make a mistake, highlight the row, change the information in the data entry boxes, and click the Change button.

The columns to add are as follows:

Table 2-2: Columns to add to heroes table

Name	Datatype	Length	Precision	Scale	Allow	Disallow	Identity
hero_id	numeric		10	0			X
hero_name	varchar	30				X	
hero_descr	varchar	40			X		
hero_deed	varchar	40			X		
hero_deedtype	char	6			X		

3. Click the Create button in the dialog box. Then, in Voyager, click the User Tables icon. The expanded *epics* database list contains a new user table icon, owned by user “homer,” with the name *heroes*.

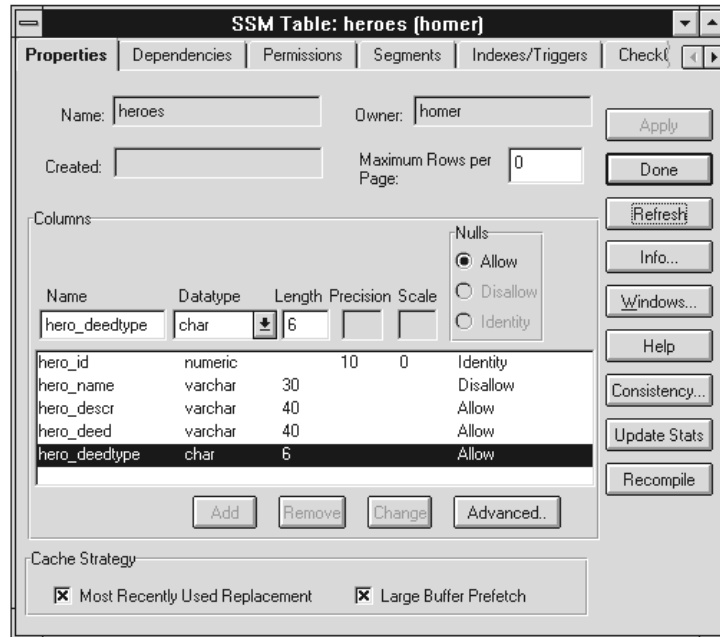


Part 4: Assign Permission on the Table

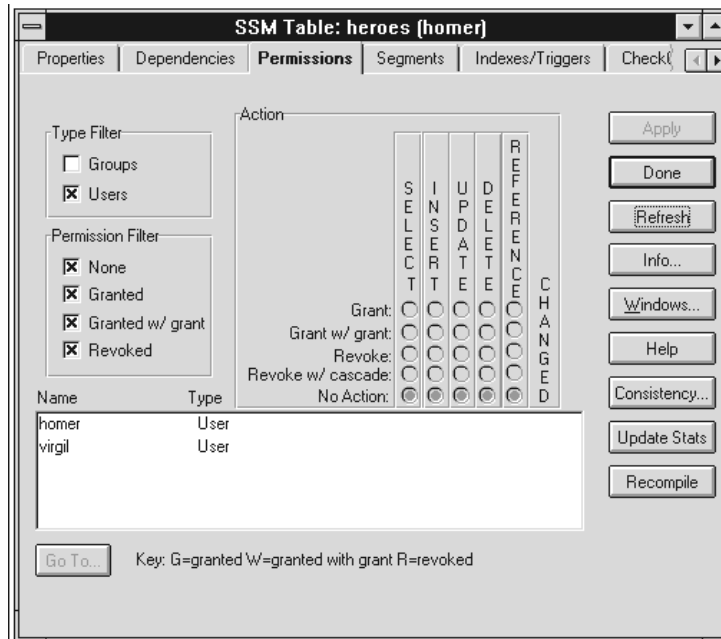
To assign permission on the *heroes* table:

1. The *heroes* table Properties dialog box should be open. If it is not, double-click the *heroes* table icon. The Table Properties dialog

box opens. The Columns group box lists the columns you added in Part 5.

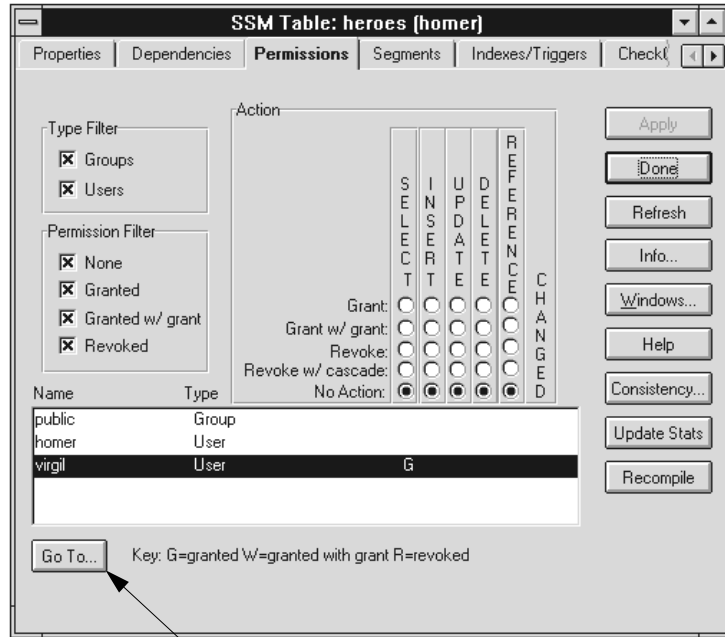


2. In the row of tabs across the top of the dialog box, select Permissions. The dialog box switches to the Permissions tab. The two users, "homer" and "virgil," appear in the scrolling list above the Go To... button.



3. The permission codes for both “homer” and “virgil” under the Select, Insert, Update, Delete, and References columns are blank. In the case of “homer”, although “homer” has all permissions implicitly by being the table owner, the permission codes are blank because SQL Server Manager reports only permissions explicitly granted, not permissions conveyed by administrative role, object ownership, or group membership. The permission codes for user “virgil” are blank because no permissions are granted to “virgil.”
4. To grant select permission on the table to user “virgil,” select “virgil,” select the Grant option button in the Select column of the Actions group box, and click Apply.

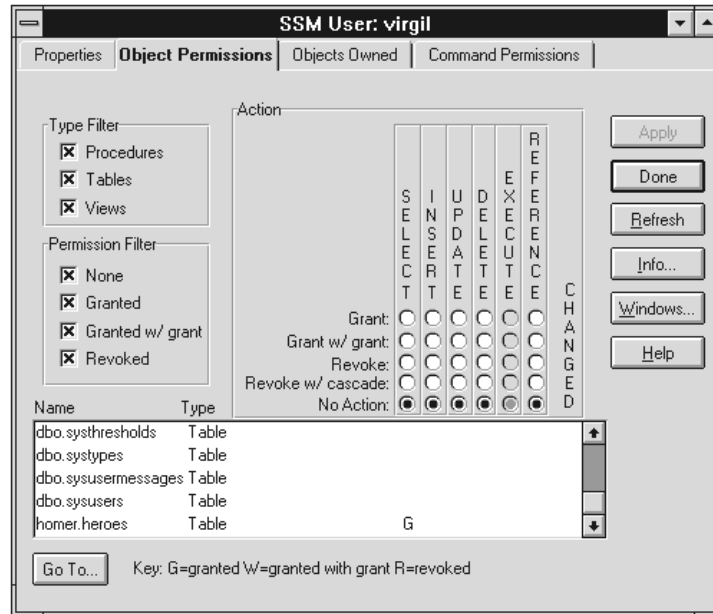
- To see the effect of granting permission to user “virgil,” open the User dialog box for “virgil” by clicking the Go To... button.



Go to Virgil

- Select the Object Permissions tab and note the record of permission granted on the *homer.heroes* table to “virgil” in the list in the Object Permissions group box. The letter “G” in the Select

column of the *homer.heroes* row indicates that “virgil” has been granted select permission on *homer.heroes*.

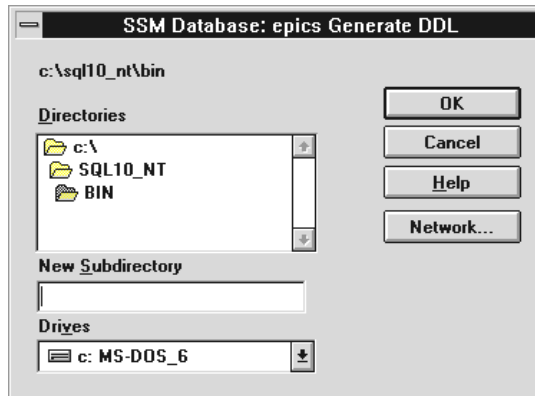


- Close the User and Table dialog boxes by clicking the Done button on each.

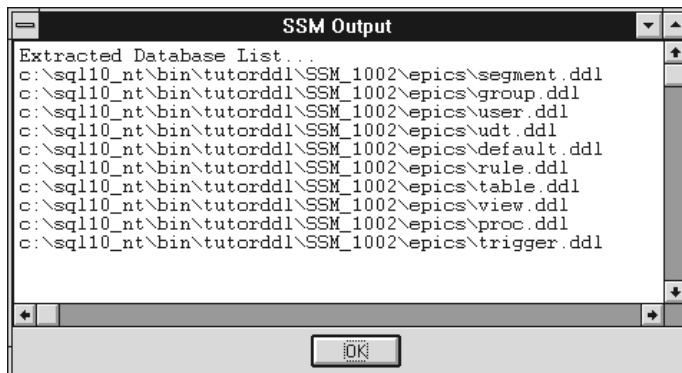
Part 5: Generate Database DDL

Use this part to generate the schema definition language that allows you to recreate the definition of the objects in the *epics* database if you need to recreate the database or move it to a different SQL Server. Start with the *epics* database icon selected.

1. From the Database menu, Choose Generate DDL... The Generate DDL dialog box opens.

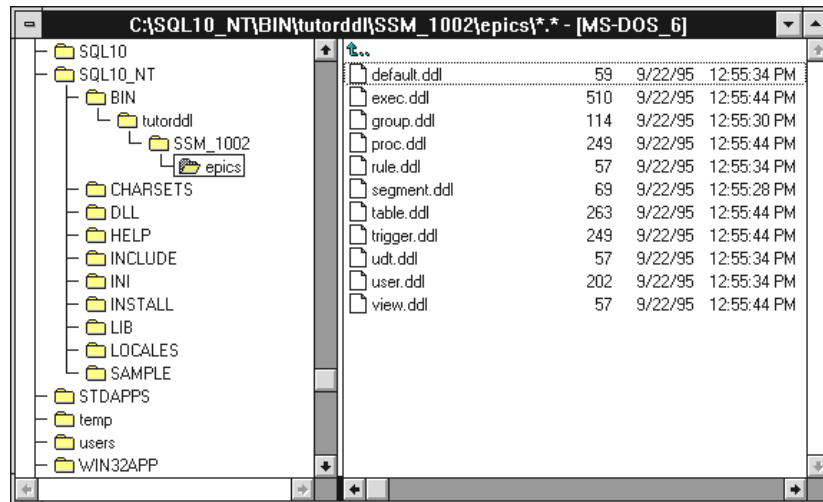


2. Enter the following information:
 - For New Subdirectory, type:
`tutorddl`
 - For Drives, use the default.
3. Click the OK button. SQL Server Manager creates the *tutorddl* directory and the batch files needed to create the definitions of each *epics* database object. A message window displays status messages as the generation progresses.



To dismiss this window, click OK.

4. After SQL Server Manager finishes generating DDL batch files, use the File Manager (Explorer in Windows 95) to see the directory and file structure it created.



5. Use the Notepad application to examine the text of the files. Here is a sample of the text of the *user.ddl* file, which contains the schema definition language needed to recreate users “homer” and “virgil.”

```

File Edit Search Help
use epics
go

print 'homer'
EXECUTE sp_adduser 'homer', 'homer', 'public'

go

print 'virgil'
EXECUTE sp_adduser 'virgil', 'virgil', 'public'

go
DUMP TRAN epics WITH TRUNCATE_ONLY
go

```

Part 6: Clean Up and Exit

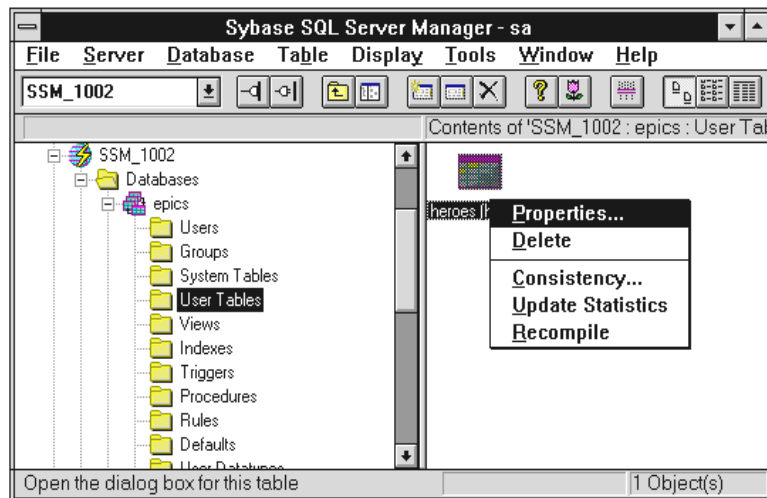
The final activities are to delete the objects and files you have created and exit SQL Server Manager:

1. Delete the *tutorddl* directory and its files by using the File Manager (Explorer in Windows 95).
2. Delete the objects in the *epics* database in this order:
 1. *heroes* table
 2. “homer” and “virgil” users

You can delete objects by choosing the Delete... command from the object's drop-down menu. However, in this part, use the shortcut menu for each object's icon.

For each object in turn:

1. Double-click the object type icon (for example, the Tables icon) to display a list of objects in the right pane.
2. Move the cursor over the object's icon.
3. Display the icon shortcut menu by pressing the right mouse button.
4. Choose the Delete... command.



5. When the delete confirmation dialog box appears, click Yes. The icon disappears from Voyager.

3. Delete the SQL Server objects you created in this order:

1. *epics* database
2. “chaucer,” “homer,” and “virgil” logins

For each object in turn:

1. Double-click the object type icon (for example, the Databases icon) to display a list of objects in the right pane.
 2. Move the cursor over the object’s icon.
 3. Display the icon shortcut menu by pressing the right mouse button.
 4. Choose the Delete... command.
 5. Click Yes in the confirmation dialog box. The icon disappears from Voyager.
4. Disconnect from SQL Server and exit SQL Server Manager by choosing Exit from the File menu.

Now that you have seen some SQL Server Manager display and navigation features and walked through some common system and database administration activities, read Chapter 3, “Introducing SQL Server Manager Windows and Dialog Boxes,” for more information on the graphical user interface.

3

Introducing SQL Server Manager Windows and Dialog Boxes

What's in This Chapter

This chapter introduces the windows and dialog boxes of SQL Server Manager and familiarizes you with the application's common features.

If you are new to SQL Server Manager, you may find it helpful to read straight through this chapter and have the product running as you proceed.

Using the Voyager Window

The Voyager window is the primary window in SQL Server Manager. It allows you to see and operate on the SQL Servers you administer and the objects in those SQL Servers.

This section describes the appearance of the Voyager window and tells how to:

- Move through the Voyager hierarchy
- Open additional Voyager windows
- Customize the display
- Use drag-and-drop shortcuts

Features of Voyager

The following figure shows a view of the SQL Server Manager window and indicates the main features of the Voyager display. A description of each feature follows.

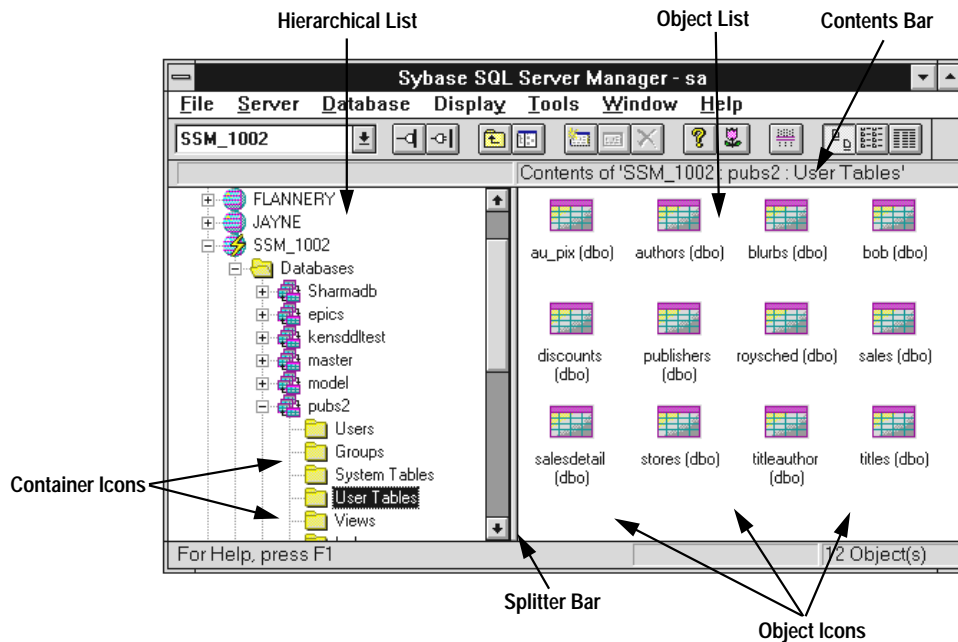


Figure 3-1: SQL Server Manager Voyager window

- The window is split into left and right panes:
 - The left pane contains the hierarchical list, which shows SQL Servers and their objects in a tree list display.
 - The right pane displays the contents of the object selected in the left pane. The Contents bar at the top of the pane describes the current contents of the pane.

To adjust the size of the left and right panes, drag the splitter bar to the left or right with the mouse pointer.

- The window includes container icons and object icons:
 - Each **container icon** represents all objects of its type within a SQL Server or database. For example, the Databases container icon in the SSM_1002 SQL Server represents all databases in

SSM_1002. Container icons can appear in either the left pane or the right pane of the window.

- Each **object icon** represents a single SQL Server or database object. For example, the “authors” user table icon represents the “authors” user table. Object icons, except databases, appear only in the right pane of the window. Database icons can appear in the hierarchical list in the left pane or in the object list in the right pane.

In most activities, before performing an operation on an object, you select the object by clicking its icon. The type of object selected determines the range of commands available to you.

- A Voyager window shows SQL Server connections for only one login at a time. When you connect to more than one SQL Server with the same login, Voyager displays the object hierarchies of all connected SQL Servers. When you connect to a SQL Server by using a different login than the original one, a second Voyager window opens for the new login.

For example, if you connect to the JAYNE and the SSM_1002 SQL Servers by using the “sa” login, the Voyager window shows the hierarchies of both SQL Servers. If you then connect to the SSM_1002 SQL Server by using the “ann” login, SQL Server Manager opens a new Voyager window showing the SSM_1002 SQL Server hierarchy.

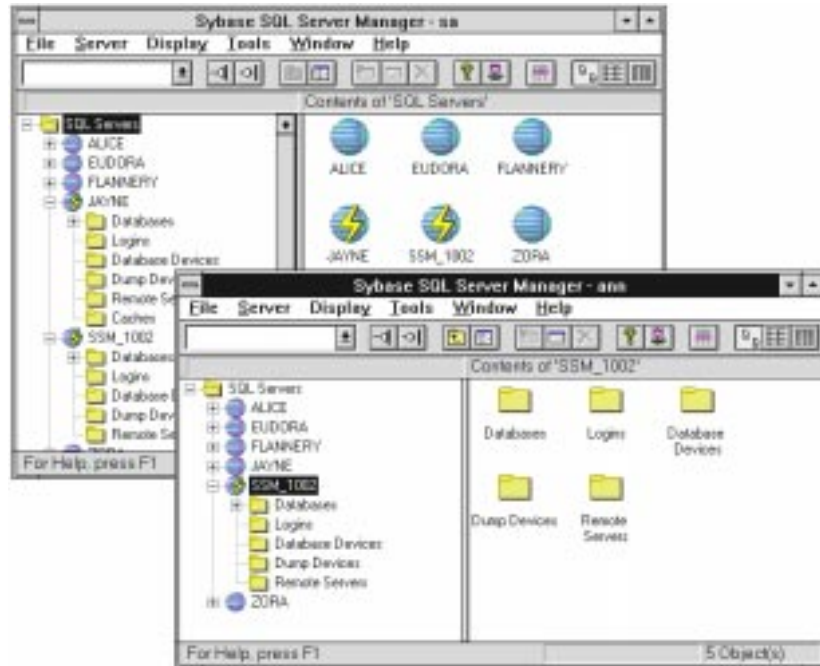


Figure 3-2: Multiple Voyager windows for multiple logins

Moving Through Voyager

To see different parts of the Voyager list, use the following techniques:

- To move vertically through the current display, use the scroll bar on the left or right pane.
- To expand or collapse Voyager to show different levels of detail, do one of the following:
 - Click plus or minus buttons. A plus button next to an icon indicates that the list of objects for that icon can be expanded. A minus button indicates that the list of objects for the icon is fully expanded.
 - Double-click a container icon or its label in Voyager.

► **Note**

Double-clicking a **container** icon in the right pane also expands the list and changes the view in the right pane to a list of objects in the container. Double-clicking an **object** icon in the right pane opens a dialog box that displays the object's properties.




Opening Another Voyager Window

At times, it is necessary or convenient to open one or more Voyager windows, in addition to the original window. SQL Server Manager automatically opens a second window when you connect to a SQL Server with a different login than you used in your first connection. Other reasons for opening additional Voyager windows are:

- To view different parts of a long Voyager hierarchy at the same time
- To perform a drag-and-drop operation involving two different types of objects in the same database or involving an object that is scrolled out of view
- To view objects in multiple containers simultaneously

To open a new window, choose **New Window...** from the **Display** menu. SQL Server Manager creates a copy of the entire Voyager. In the new window, the container icon selected in the original window is in view and is selected. Its objects are displayed in the right pane.

Shortcut

	Drag any container icon and drop it onto the desktop outside the Voyager window
	Select the New Window toolbar button.
	Select the New Window... command from the shortcut menu for any container icon in the left pane of the Voyager window

Customizing the Display

To customize the Voyager window, you can choose how information in the right pane is displayed. SQL Server Manager allows you to select from several display formats and to limit the display to objects that match a specified character string.

Selecting Display Formats

To select the display format for the right pane, choose one of the following commands from the Display menu:

- Large Icon—shows each object as a large icon with its label underneath.
- List—shows each object as a small icon next to its label.
- Details—shows a table of detailed information about each object. The details vary with the type of object.

The following figure shows an example of each format:

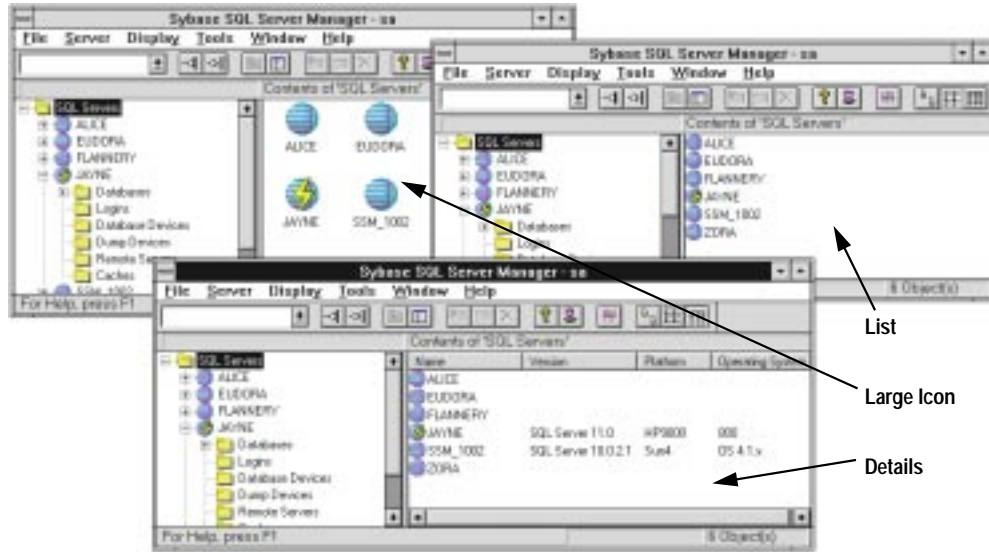


Figure 3-3: Object display formats

When you select the Details display format, by default, the data is sorted by object name. To sort by using the data in a different column than Name, click the column heading. For example, the following figure shows the Details view of the databases in the connected SQL Server sorted by database name and by size

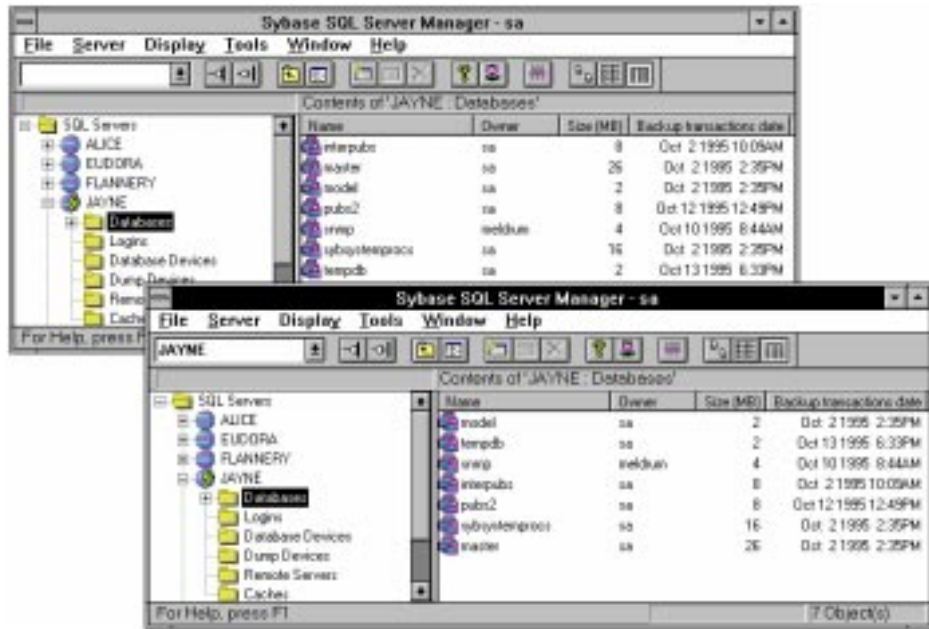


Figure 3-4: Details view of databases, sorted by name and by size

Shortcut



Choose the Large Icon, List, or Details button from the toolbar.



Click the right mouse button over an empty space in the right pane, and select Large Icon, List, or Details from the shortcut menu.

Filtering by Name

To select icons for display by searching for their names:

1. Choose the Filter By Name... command on the Display menu. The Filter by Name dialog box opens.

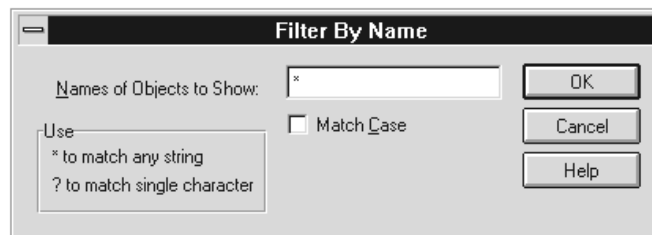


Figure 3-5: Filter by Name dialog box for filtering icons by name

2. Type a string in the Names of Objects to Show edit box. If you do not know the full name of the icon, you can use either or both of the following masking characters:

- An asterisk (*) can represent zero or more characters.
- A question mark (?) can represent any one character.

For example, the following string, entered when the right pane displays user tables, would return all user tables whose names start with "au":

`au*`

The following string:

`94??log`

might return icons named *9401log*, and *9402log*, but would not return an icon named *940115log*.

3. To match the case of the search string, check the Match Case box. For example, the string `gil*` with Ignore Case checked might return icons with names beginning with *Gil*, *GIL*, and *gil*.

Click OK. SQL Server Manager filters icons based on what you typed.

► **Note**

When filtering is in use, the status bar displays the current filter string.

Shortcut

Select the Filter by name toolbar button.



Using Drag-and-Drop Shortcuts

A quick way to perform operations involving the interaction of two SQL Server or database objects is to drag the icon for one object on top of an icon for another object. When you drag and drop icons for which SQL Server Manager understands the operations, the application invokes the appropriate dialog box with most or all of the required information filled in. Provide or modify information as necessary, and click OK.

SQL Server Manager supports **symmetrical drag-and-drop** actions; that is, it allows you to drag either of two icons on top of the other.

► **Note**

For a list of valid drag-and-drop actions, see Appendix B, “Guide to SQL Server Manager Icons.”

To drag and drop an icon:

1. Place the cursor on top of the icon, and press and hold the left mouse button.
2. Continuing to hold the button down, move the mouse to drag the icon on top of another icon. As you drag the icon, a faint image of the icon travels across the screen, along with a symbol indicating that you can not yet legally drop the icon. When the icon reaches its target location, the illegal drop symbol disappears, and you can release the mouse button.

In the following figure, the icon for the user named “virgil” is being dragged onto the “writers” group icon to add “virgil” to the

“writers” group. The ghost image of the “virgil” icon is almost close enough to drop onto the “writers” icon.

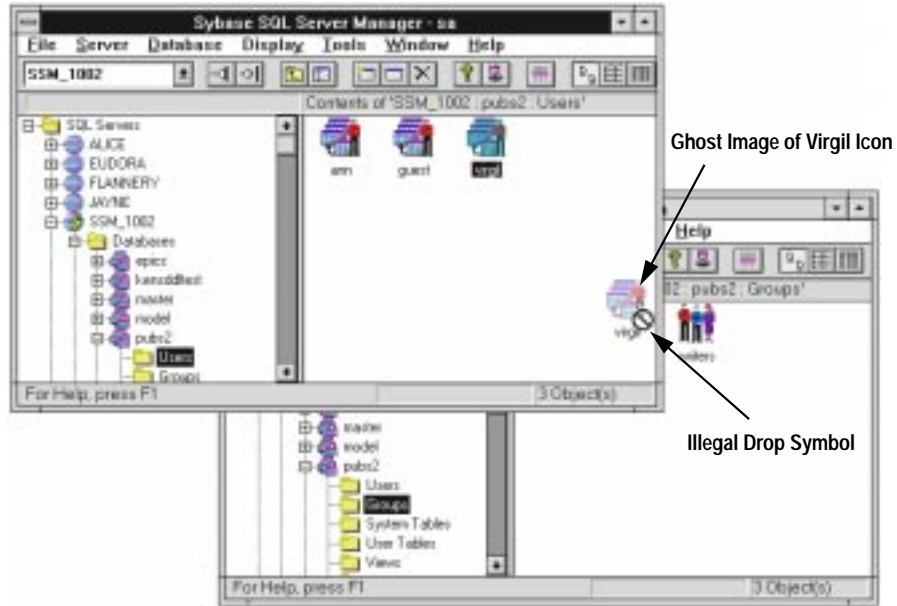


Figure 3-6: Drag-and-drop symbol

3. Release the mouse button. A dialog box opens for specification and confirmation of the operation.

Menus and Toolbars

This section describes the menus and toolbars you see in SQL Server Manager.

Standard Menus

The menu bar in the Voyager window contains the menus shown in the following figure.

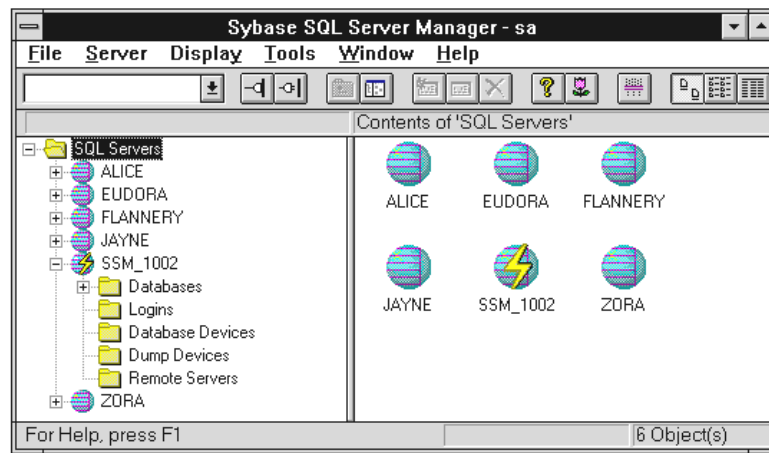


Figure 3-7: Standard menu bar

Unlike **context-sensitive menus**, which change in the menu bar according to the context of your current activity, standard menus remain constant as you use SQL Server Manager. The following table summarizes the activities to which they provide access.

Table 3-1: Activities available from standard menus

Menu	Activities
File	<ul style="list-style-type: none"> • Exit SQL Server Manager • Close the current Voyager window if more than one is open.

Table 3-1: Activities available from standard menus (continued)

Menu	Activities
Server	<ul style="list-style-type: none"> • Connect to or disconnect from SQL Server • Create SQL Server objects • Display SQL Server properties • Update configuration variables • Display and kill user processes • Check network I/O consistency • Generate and execute DDL • Shut down a SQL Server
Display	<ul style="list-style-type: none"> • Hide or display the standard toolbar and status bar • Select the format for displaying object icons in the right pane of the Voyager window • Update the display with fresh data from SQL Server
Tools	Execute the following utilities: <ul style="list-style-type: none"> • <code>sqledit</code> utility for editing <code>sql.ini</code> files • <code>wisql</code> utility for connecting to SQL Server through the <code>isql</code> interface
Window	<ul style="list-style-type: none"> • Open a new Voyager window • Activate the selected window or dialog box
Help	<ul style="list-style-type: none"> • Display SQL Server Manager online help contents • Search for an online help topic • Display information on how to use online help • Open online help for <code>sqledit</code> or <code>wisql</code> • Display the About dialog box for SQL Server Manager

Context-Sensitive Menus

Context-sensitive menus contain commands that operate on a specific object type or a specific object. These menus appear in the menu bar when you select a container icon or object icon in the right pane of Voyager.

Each context-sensitive menu has the name of the object it represents; for example, the Database menu contains commands for database creation, updating, and administration.

Context-sensitive menus appear in two forms: pull-down menus and shortcut menus. The pull-down and shortcut menus for a specific object icon have the same list of commands.

Pull-Down Menus

To activate a context-sensitive pull-down menu, click a container icon or object icon representing a SQL Server or database object. The menu is added to the menu bar. Then, select the menu by clicking it or pressing the Alt key and the mnemonic (underlined) character. The following figure shows the Table menu for the selected database table.

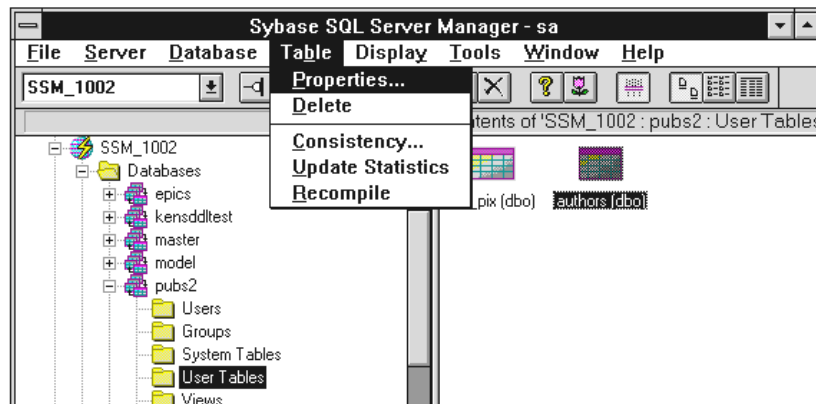


Figure 3-8: Pull-down menu for a table

Shortcut Menus

To activate a shortcut menu, click the right mouse button over a container icon or object icon representing a SQL Server or database object. From the shortcut menu, choose the appropriate menu command.

Shortcut menus have different menu commands according to their context:

Table 3-2: Shortcut menu commands

Context	Commands
Container icons in left pane	<ul style="list-style-type: none"> • Create <i>object_type</i>, where <i>object_type</i> is the object represented by the container—for example, Create Login • New Window
Container icons in right pane	Create
Database icons in the left pane and all object icons in the right pane	Same commands as the pull-down menu for the object
Right pane (when clicked in an area containing no icons)	<ul style="list-style-type: none"> • Large Icon • Small Icon • Details • Create <i>object_type</i>, where <i>object_type</i> is the object in the pane—for example, Create Login

The following figure shows the shortcut version of the Table menu for the *au_pix* table.

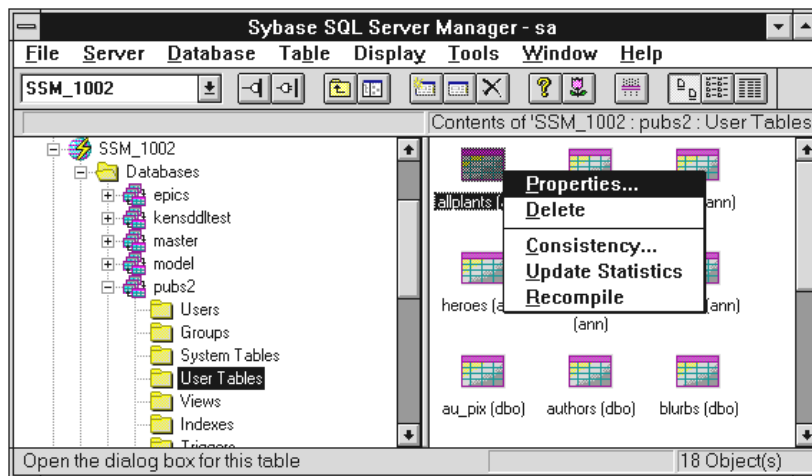


Figure 3-9: Shortcut menu for a table

Toolbar and Status Bar

The SQL Server Manager toolbar gives you a quick alternative to executing frequently-used menu commands. The status bar displays helpful information about highlighted window controls and objects. The following figure illustrates the toolbar and status bar.

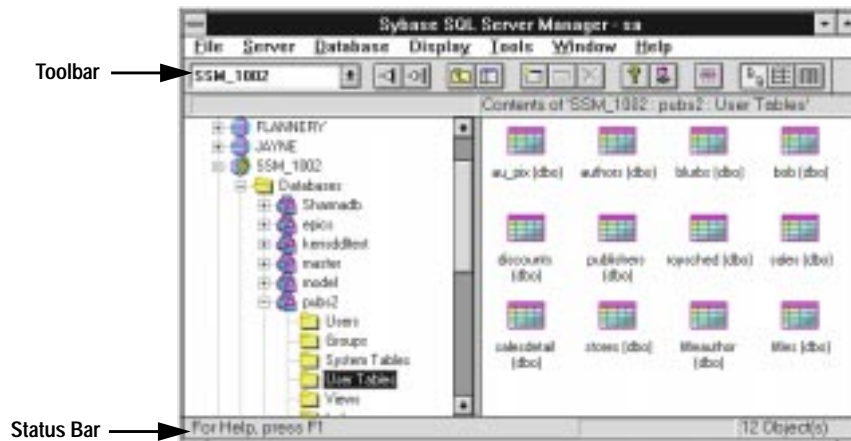


Figure 3-10: Toolbar and status bar

To display or hide the toolbar or status bar, toggle the **Toolbar** or **Status Bar** command on the **Display** menu. For example, to display or hide the standard toolbar, choose **Toolbar**. When the menu command has a check mark (✓) next to it, the toolbar is visible in the window.

Standard Toolbar

The standard toolbar consists of the following controls:

- A drop-down list box that allows you to focus the Voyager display on a selected SQL Server as an alternative to scrolling through a long list
- Buttons that provide a quick way to execute menu commands

The following figure illustrates the controls.

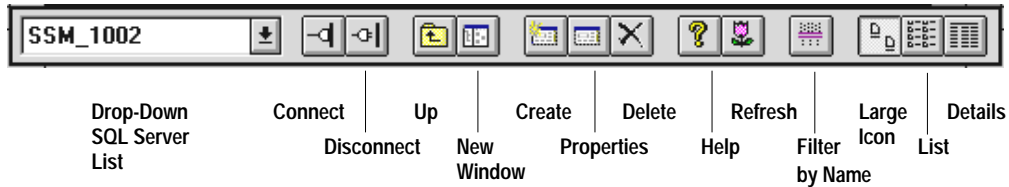


Figure 3-11: SQL Server Manager toolbar

- Drop-down list of SQL Servers—Allows you to select a SQL Server. The icon for the SQL Server you select becomes the top of the Voyager display.
- Connect—Opens the Connect dialog box so you can connect to the selected SQL Server.
- Disconnect—Disconnects from the current SQL Server.
- Up—Moves the Voyager display up one level in the object hierarchy. For example, if the current focus is on a database, clicking this button moves the focus to the SQL Server in which the database is located.
- New Window—Opens a new copy of the Voyager window.
- Create—Opens the Create dialog box associated with the selected container icon. For example, if the Logins container icon is selected, clicking this button opens the Create Login dialog box.
- Properties—Opens the Properties dialog box associated with the selected object icon.
- Delete—Deletes the selected object.
- Help—Opens the Help window and displays the Contents topic.
- Refresh—Updates all open windows with current information from SQL Server.
- Filter by Name—Allows selective display of icons based on a specified character string.
- Large Icon—Displays objects in the right pane of the Voyager window as a list of large-format icons.
- List—Displays objects in the right pane of the Voyager window as a list of object names, along with small-format icons.

- **Details**—Displays objects in the right pane of the Voyager window as a list of object names, along with details about each object. The information shown varies by object type.

To help identify the buttons in the toolbar, SQL Server Manager displays **ToolTips**—floating labels that appear when the cursor rests over a toolbar button. To display a ToolTip, place the cursor over a toolbar button for a second; it is not necessary to click the button.

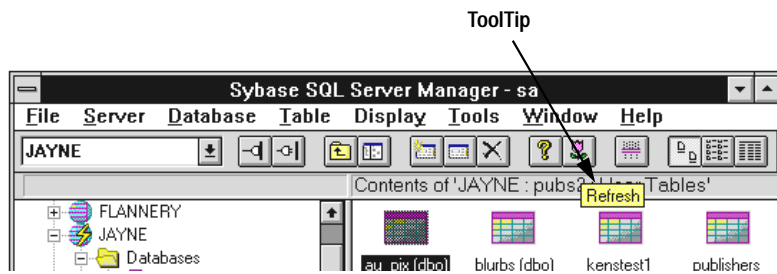


Figure 3-12: ToolTip for the Refresh tool bar button

Status Bar

The status bar is an information display bar located at the bottom of the application window. In SQL Server Manager, the status bar consists of the following elements:

- A help line containing a brief description of the activity to which the cursor is pointing. For example, if the cursor is pointing to a menu command, the help line shows a brief description of that command. The help line appears on the left side of the status bar.
- The filtering string currently in use, if applicable.
- The number of objects displayed in the right pane.

The following figure shows the status bar for a display in which the Properties... command is selected in the Table menu and filtering by name is activated.

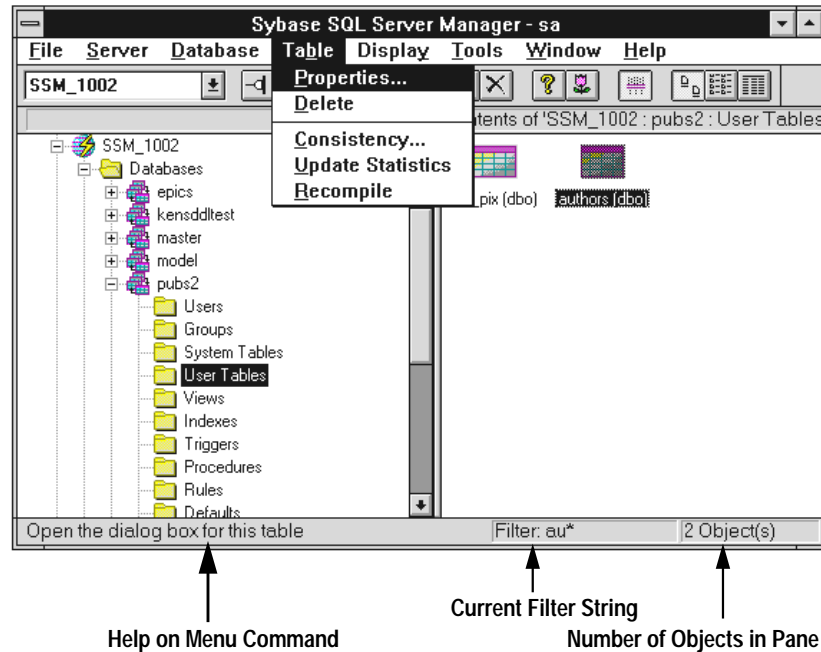


Figure 3-13: Status bar

Types of Dialog Boxes

SQL Server Manager has the following types of dialog boxes:

- Object dialog boxes that hold information about SQL Server and database objects
- Command dialog boxes for specifying how to execute menu commands
- Confirmation dialog boxes that ask you to confirm an action, such as deleting an object
- Message dialog boxes that communicate information from the application or SQL Server

Object Dialog Boxes

After you create a SQL Server or database object, you can display or update the object by opening its object dialog box. An object dialog box contains information about the object and about how it relates to other objects in the SQL Server or database. The dialog box also provides a direct navigation path to its related objects.

Using Dialog Box Tabs

Information about SQL Server and database objects appears on multiple screens within object dialog boxes. These dialog box screens are called **tabs**, referring to the scrolling row of tab-shaped controls across the top of the dialog box. To scroll the tab control row, use the scroll buttons at the right of the row.

When you open the dialog box for any object, you see the Properties tab, which provides a basic description of the object. The Table dialog box's Properties tab is an example.

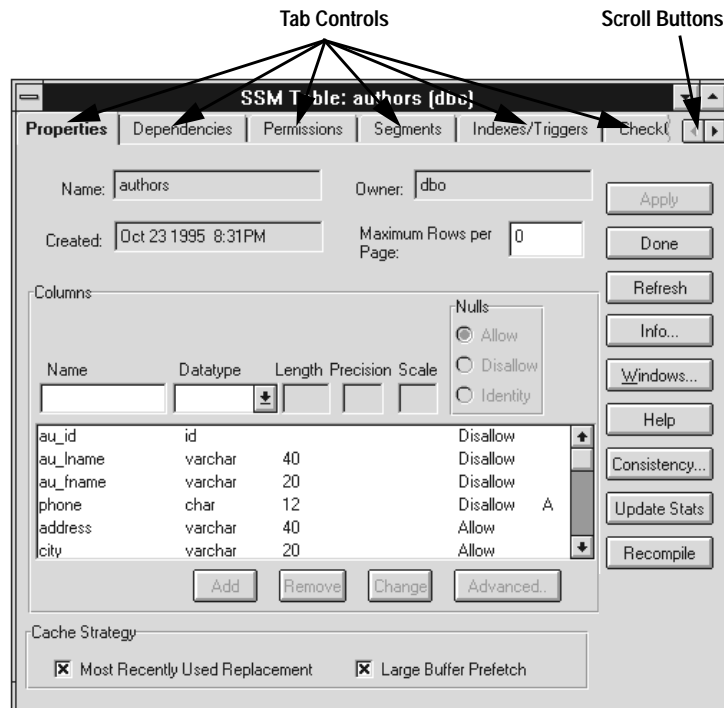


Figure 3-14: Properties tab of the multiple-tab Table dialog box

The other dialog box tabs vary according to the function of each object. To display a different tab, select that tab control. For example, to display the dependencies of this table, select the Dependencies tab control.

Note that the command buttons down the right side of the dialog box remain constant as you page between tabs:

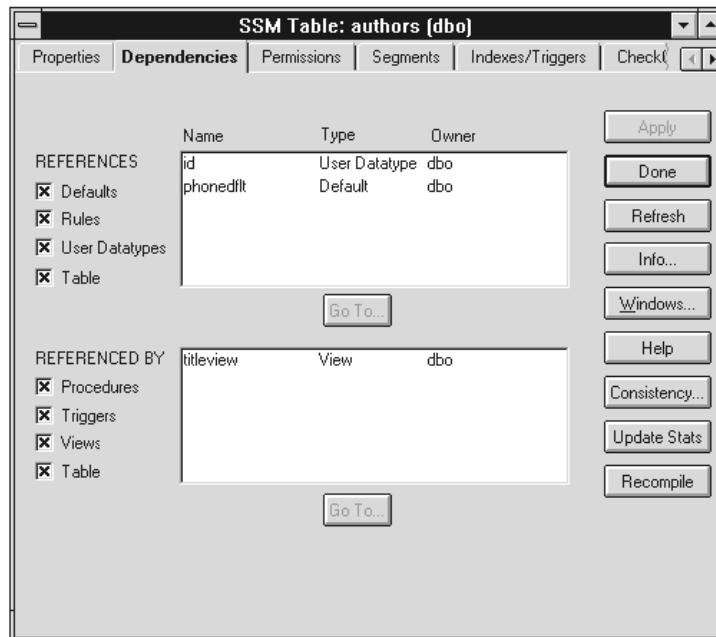


Figure 3-15: Dependencies tab of the multiple-tab Table dialog box

Navigating Between Objects

SQL Server Manager lets you move between related objects from within an object dialog box; you do not need to leave the dialog box and open the related object using menu commands. When a dialog box tab includes a list of objects, you can open any object's dialog box by highlighting its row in the list and clicking the Go To... button.

For example, if you highlight the “titleview” row in the Dependencies tab of the *authors* table and click Go To..., SQL Server Manager opens the Properties tab of the *titleview* view.

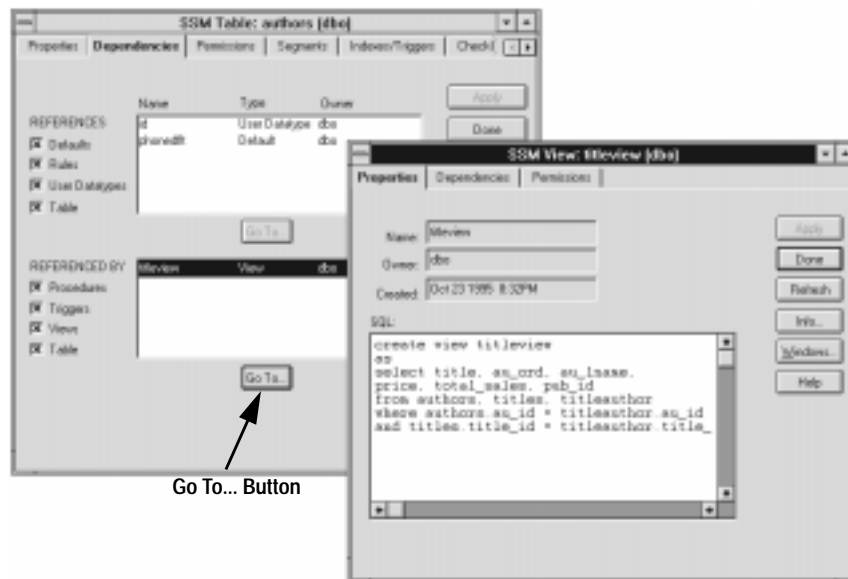


Figure 3-16: Navigating to a related object

Closing an Object Dialog Box

To close an object dialog box and save any changes you have made, click Create or Apply, depending on whether you are creating a new object or modifying an existing object. To close an object dialog box without saving changes, click Done. A confirmation dialog box opens asking if you want to close the dialog box without saving the changes. Click Yes or No.

Command Dialog Boxes

A command dialog box opens when you choose a command that requires user input. It contains fields that allow you to specify how the command should execute. For example, when you click the

Consistency... button on the Table dialog box, the Table Consistency Check command dialog box opens:

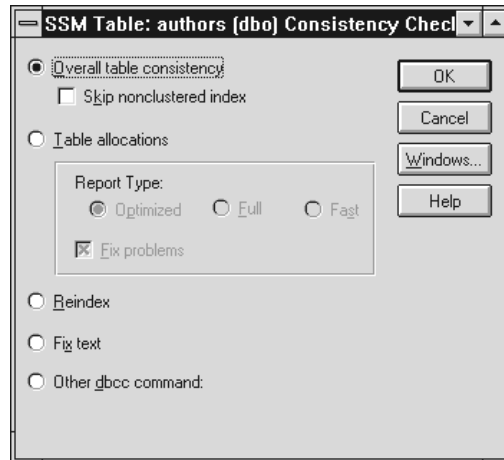


Figure 3-17: Example of a command dialog box

Closing a Command Dialog Box

To close a command dialog box and execute the command, click OK. To close the dialog box and cancel the command, click Cancel.

Confirmation Dialog Boxes

A confirmation dialog box opens when you execute a command that has permanent and possibly negative results, such as deleting an object or closing an object dialog box without saving the changes you made. The confirmation dialog box asks a question to confirm the action you want to take.



Figure 3-18: Example of a confirmation dialog box

Closing a Confirmation Dialog Box

Respond to the question by clicking the appropriate button. The specified action takes place and the dialog box closes.

Message Dialog Boxes

A message dialog box communicates information from SQL Server Manager or SQL Server. SQL Server Manager has two types of message dialog boxes:

- A simple message dialog box presents a single message. The following dialog box is an example:



Figure 3-19: Example of a simple message dialog box

- A scrolling output dialog box presents messages from SQL Server, showing multiple messages simultaneously. To see all messages, use the scroll bar.

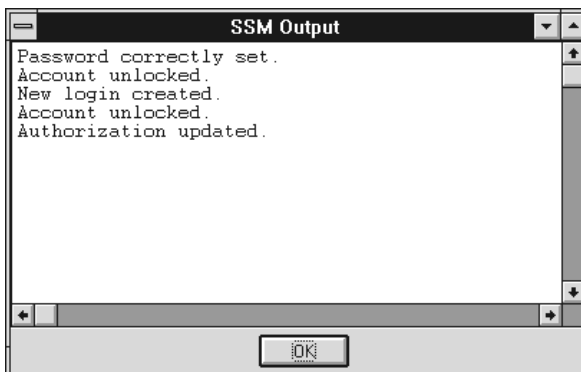


Figure 3-20: Output dialog box

► Note

To record the text of error or informational messages for future reference, copy and paste messages from the Output dialog box to a word processor or text editor such as Notepad.

Closing a Message Dialog Box

To dismiss either type of message dialog box, click OK.

Refreshing Window and Dialog Box Displays

As you work, you can update the contents of the active window or the open dialog box with new information from SQL Server. Use any of the following methods:

- To refresh the active window, choose Refresh from the Display menu.
- To refresh the active window, click the Refresh toolbar button on the active window.
- To refresh the active dialog box, click the Refresh command button.

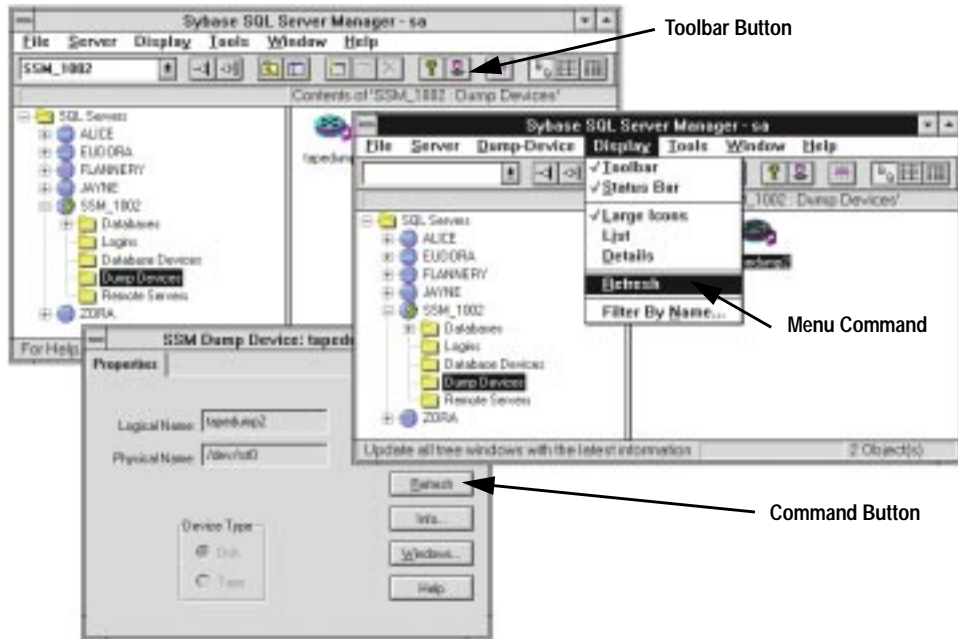


Figure 3-21: Methods of refreshing a window or dialog box

Moving Between Open Windows and Dialog Boxes

SQL Server Manager allows you to leave windows and dialog boxes open as you work so you can move between related activities without having to perform extra steps to re-open objects. To move between open windows or dialog boxes, use either of the following methods:

- Choose the window or dialog box name from the Window menu.
- Click the Windows... button on any dialog box and choose the window or dialog box name from the list that appears.

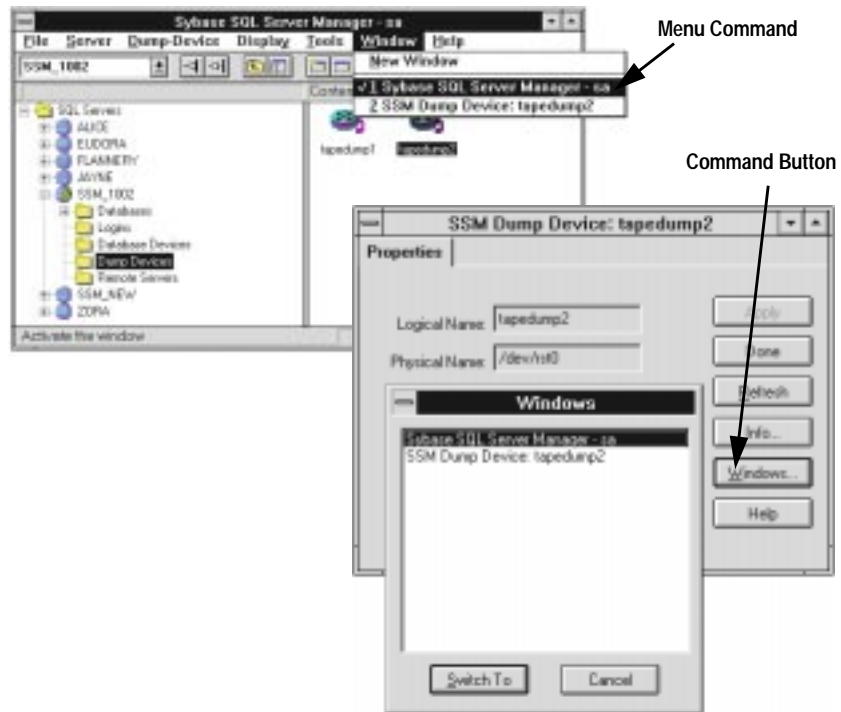


Figure 3-22: Methods of moving between open windows and dialog boxes

4

Managing the SQL Server Operating Environment

What's in This Chapter

This chapter describes the SQL Server administration tasks you can perform with SQL Server Manager. Use the instructions in this chapter to learn how to:

- Make and discontinue connections to SQL Server
- Display information about a particular SQL Server and the objects it controls
- Set SQL Server configuration parameters
- Display user processes and kill runaway processes
- Enable communication with remote servers
- Generate and execute DDL scripts
- Check engine network I/O for a release 11.0 SQL Server

Connecting to and Disconnecting from SQL Server

Before you can perform any SQL Server administration activities, you must be connected to the SQL Server you want to administer. You can use SQL Server Manager to connect to any SQL Server identified in your installation's *sql.ini* file, and you can maintain multiple SQL Server connections simultaneously.

Connecting

When you start SQL Server Manager by double-clicking its start-up icon, the SQL Server Manager (Voyager) window opens, displaying a list of icons representing SQL Servers in the *sql.ini* file.

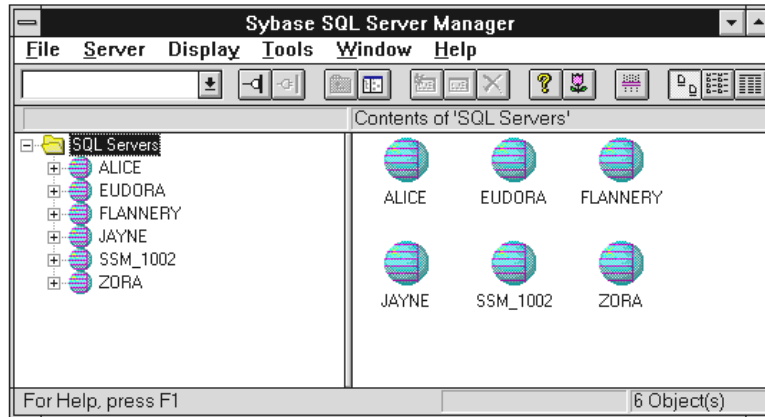


Figure 4-1: SQL Server Manager window

To connect to a SQL Server:

1. In the right pane, select the icon for the SQL Server you want and choose **Connect...** from the **Server** menu. The **Connect** dialog box opens.

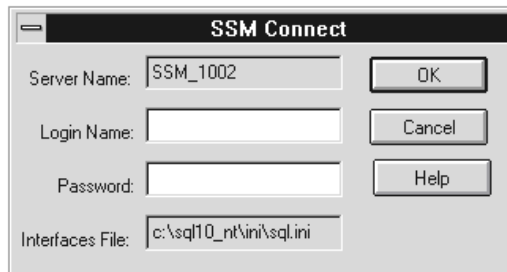


Figure 4-2: Connect dialog box

2. If the **Server Name** edit box is blank, or to change to a different SQL Server, select the SQL Server to connect to from the **Server Name** list. (The box is blank only if you chose the **Connect...**

- menu command without previously highlighting a SQL Server icon.)
3. Enter the login name with which to connect to the SQL Server.
 4. Enter the password for the login.
 5. Click OK. If the connection is successful, SQL Server Manager changes the SQL Server icon in Voyager to a connected SQL Server icon and allows you to expand the list.

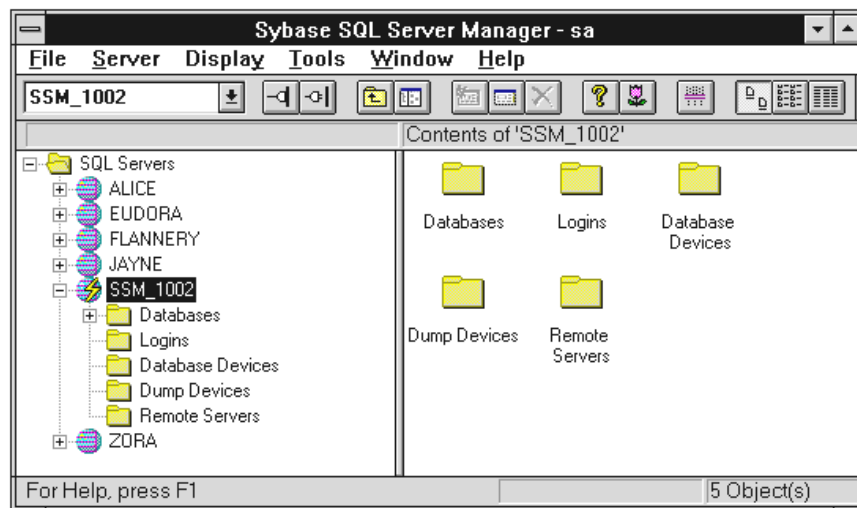
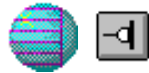


Figure 4-3: Connected SQL Server icon in Voyager

To exit without connecting to SQL Server, click Cancel.

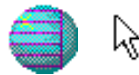
Shortcuts



Select the icon for the SQL Server you want, and select the Connect toolbar button.



Click the plus button or the SQL Server icon in the left pane.



Double-click the icon for the SQL Server you want.

► **Note**

If you start SQL Server Manager from the Run... command line in Windows or Windows NT or from the Start... command line in Windows 95, you can bypass the Connect dialog box by providing enough information for SQL Server Manager to make the connection. For details, see “Starting SQL Server Manager” on page 2-1.

Diagnosing an Unsuccessful Connection

If the connection is not successful, try these techniques:

- Check your login name and password to be sure they are correct.
- Use the `sqledit` utility by double-clicking its icon on the desktop. From `sqledit`, select the Ping... button to invoke the `ping` utility. `ping` tests whether a specified SQL Server is alive.

For information on `sqledit` and on the format of `sql.ini` file entries, see Chapter 11, “Using System Management Tools.”

Disconnecting from SQL Server

When you disconnect from SQL Server, the SQL Server icon in Voyager changes to a disconnected icon and you no longer have access to details about the objects in the SQL Server.

1. Select the icon of the SQL Server to disconnect.
2. From the Server menu, choose Disconnect. The Disconnect dialog box opens.

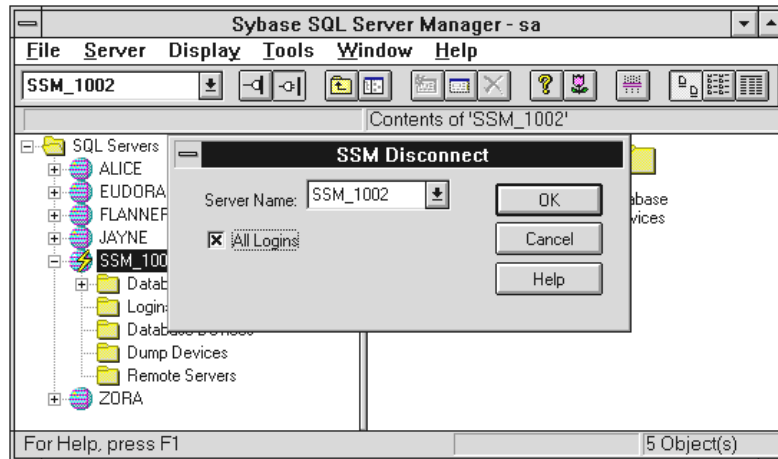
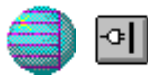


Figure 4-4: Disconnect dialog box

3. To disconnect from a SQL Server other than the one highlighted in the Voyager window, select the SQL Server name in the Server Name list.
4. To disconnect a specific login, clear the All Logins box. The Login Name list becomes visible. Select the login to disconnect and click OK. To disconnect all logins, leave the All Logins box selected and click OK.

Shortcut



Select the icon of the SQL Server you want, and select the Disconnect toolbar button.

Displaying SQL Server Details

When you are connected to a SQL Server, you can easily display details about the SQL Server or any of its objects.

Examining SQL Server Properties

To display a dialog box that shows the properties of a SQL Server to which you are connected, select the SQL Server icon and choose Properties... from the Server menu.

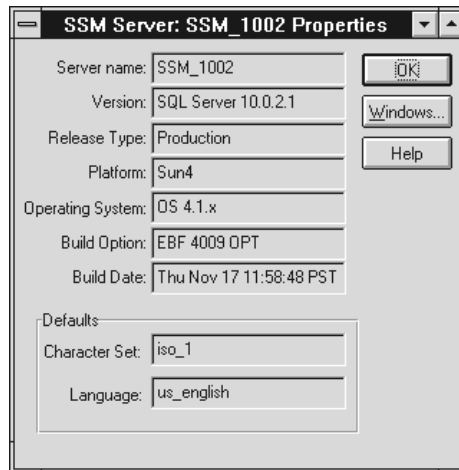


Figure 4-5: Server Properties dialog box

This dialog box shows the SQL Server name, version number, release type, platform, operating system, build option, and build date. The dialog box also shows the server default language and character set.

Examining SQL Server Objects



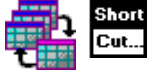
The Voyager window contains icons representing the objects that this SQL Server manages. For a picture of what each type of icon looks like, see Appendix B, “Guide to SQL Server Manager Icons.” To display the properties of any SQL Server object:

1. Select the object’s icon in the right pane of the Voyager window.
2. From the object’s menu in the menu bar, choose Properties...

For example, to display the properties of a database device managed by the current SQL Server, select the database device icon and choose Properties... from the Database Device menu.

The dialog box for the selected object opens with the Properties tab visible.

Shortcuts

	Double-click the object icon.
	Select the object icon and select the Properties toolbar button.
	Click the right mouse button over the object icon and choose Properties... from the shortcut menu.

Shutting Down SQL Server

You can shut down a SQL Server in either of the following modes:

- **Immediately**—shuts down SQL Server immediately, aborting user processes. If you use this mode, you can help minimize recovery time by issuing a checkpoint in each database before shutting down SQL Server. For information on issuing a checkpoint, see “Issuing a Database Checkpoint” on page 7-22.
- **When Ready**—shuts down SQL Server and performs the following activities to facilitate a smooth automatic recovery when you restart:
 - Disables logins, except for logins with the System Administrator role
 - Performs a checkpoint in each database, flushing from memory to disk any pages that have changed
 - Waits for currently executing SQL statements or procedures to finish

To shut down SQL Server:

1. Select the icon of the SQL Server to shut down.
2. From the Server menu, choose Stop. Then, choose one of the following commands from the cascading menu:
 - Immediately—shuts down immediately
 - When Ready—shuts down when ready

When you shut down a SQL Server, its icon changes from a connected server to an unconnected server.

► **Note**

SQL Server Manager cannot start a SQL Server. To restart on Unix, you must run the `startserver` utility on the host machine. For information on `startserver`, see the *SQL Server Utility Programs for Unix* manual. To restart on a PC platform, refer to the server documentation for the platform.

Configuring SQL Server

SQL Server configuration parameters allow you to control aspects of memory allocation and performance. When you install SQL Server, default configuration values take effect. To fine-tune memory allocation and performance, users with the System Administrator role can reset most configuration parameters. Users with the System Security Officer role can also reconfigure some of the parameters.

This section describes how to set configuration parameters. Parameter setting for release 10.x and release 11.0 SQL Servers is described in separate subsections.

Appendix A, “SQL Server Configuration Parameters” gives a short description of each configuration parameter. For a more in-depth explanation, see the *System Administration Guide* for the appropriate SQL Server release. For a discussion of configuration issues to consider when determining optimal settings, see the *Sybase Troubleshooting Guide*.

Who Can Set Configuration Parameters

The following rules govern who can set configuration parameters

- Users with logins assigned the System Security Officer role can reset the following configuration parameters:
 - allow updates
 - audit queue size
 - password expiration interval
 - remote access
- The default character set id parameter is set during SQL Server installation and cannot be reset from within SQL Server Manager.
- Users with logins assigned the System Administration role can reset all other parameters.

Parameters Requiring Restart

Some parameter values take effect as soon as you reset the value. Others do not change until you reset the value and then reboot SQL Server. In the Server Configuration Parameters dialog box, parameters requiring a SQL Server reboot have a check mark in the Requires Restart column.

Configuring a Release 10.x SQL Server

To reset configuration parameters for a release 10.x SQL Server:

1. Select the SQL Server to configure.
2. From the Server menu, choose Configuration... The Configuration Parameters dialog box opens.

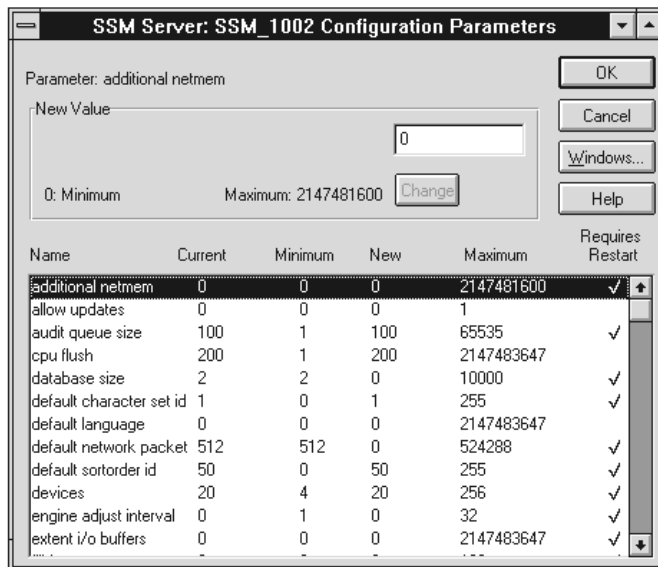


Figure 4-6: Configuration Parameters dialog box for 10.x SQL Server

3. Scroll through the list of configuration parameters to locate one that you want to reset. Then, select it by clicking anywhere in its row. The name of the parameter you select appears above the New Value group box. The Minimum and Maximum labels show the minimum and maximum allowed for a value.
4. In the edit box, enter the new value of the configuration parameter.
5. Click the Change button. SQL Server Manager updates the values of the New column to the values you entered. The New column shows the value most recently set for each configuration parameter.
6. Repeat Steps 3 through 5 for each parameter you want to update.

7. Click OK. SQL Server Manager updates the configuration values in the dialog box as follows and automatically issues a **reconfigure** command to SQL Server:
 - If the parameter you reset takes effect immediately, SQL Server Manager copies the value in the **New** column to the **Current** column.
 - If the parameter you reset requires a SQL Server reboot, SQL Server Manager does not update the **Current** column value until you reboot SQL Server. The **Requires Restart** column for such parameters contains a check mark (✓).



Configuring a Release 11.0 SQL Server

In SQL Server 11.0, configuration parameters are organized by functional group, according to the areas of SQL Server behavior that they affect. When displaying parameters, you can select the functional group to display.

Within functional groups, you can restrict the number of parameters displayed by setting the display level to one of the following:

- **Standard**—displays basic parameters for general SQL Server tuning
- **Advanced**—displays all Standard level parameters and additional parameters that allow for more comprehensive SQL Server tuning
- **Complete**—displays all parameters

Setting Configuration Parameters

To set configuration parameters for a release 11.0 SQL Server:

1. Select the SQL Server to configure.
2. From the Server menu, choose Configuration... The Configuration Parameters dialog box opens.

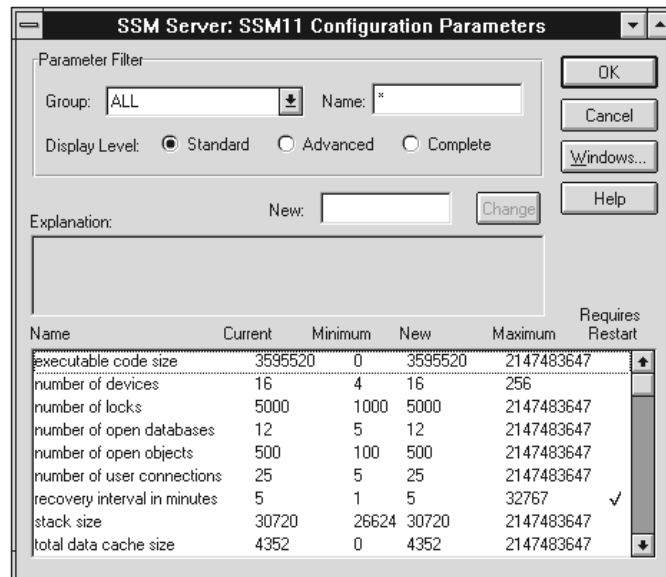


Figure 4-7: Configuration Parameters dialog box for 11.0 SQL Server

3. Select the parameters to display:
 - From the Group list, select the functional group to display, or select "ALL."
 - Select the appropriate Display Level option button.
 - Alternatively, if you know which parameter you want to update, enter its name in the Name box. The masking characters "*" and "?" are valid. For example, to display parameters within the Group list and Display Level whose name begins with "p", enter:

p*

4. In the parameter list at the bottom of the dialog box, select the parameter to update. For a brief description of the selected parameter, see the Explanation box.

5. In the New box, enter the new value of the parameter, and click Change.
6. Repeat Steps 3 through 5 for each parameter you want to update.
7. Click OK. SQL Server Manager updates the configuration values in the dialog box as follows:
 - If the parameter you reset is takes affect immediately, SQL Server Manager copies the value in the New column to the Current column.
 - If the parameter you reset is requires a SQL Server reboot, SQL Server Manager does not update the Current column value until you reboot SQL Server. The Requires Restart column for such parameters contains a check mark (✓).

Displaying and Killing SQL Server Processes

You can use SQL Server Manager to view the status of user processes on SQL Server and to kill processes that are interfering with other users, blocking access to database objects, or consuming excessive system resources.

Displaying Processes

To display user processes on a SQL Server to which you are connected, select the SQL Server icon.

Choose Processes... from the Server menu. The Server Processes dialog box opens.

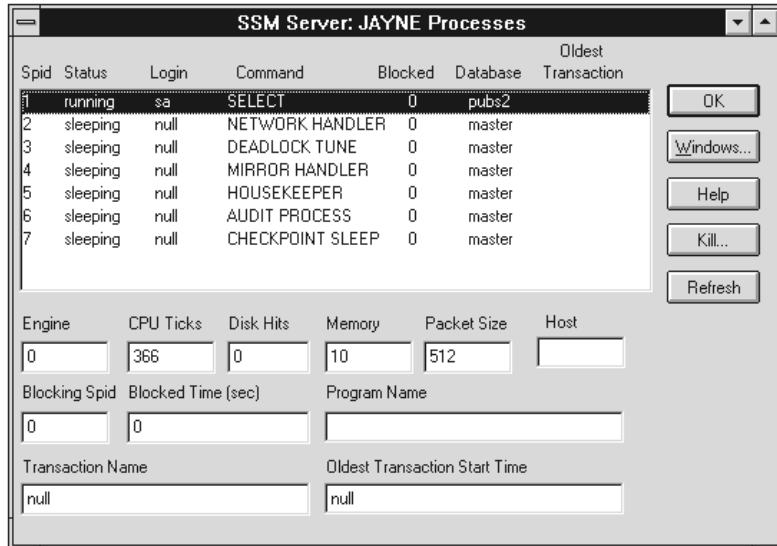


Figure 4-8: Server Processes dialog box

This dialog box displays information about the status and source of each user process, the command it is currently executing, and the database it is using.

Viewing Additional Process Details

In addition to the information shown for all transactions in the transaction list, the following details are displayed in boxes below the list for any transaction that you select:

- Number of the engine on which the process is being executed
- Cumulative CPU time for the process, in ticks
- Number of disk reads and writes for the current command
- Amount of memory allocated to the process
- Network packet size of the current connection
- Name of the host where the process is executing
- Process ID of the blocking process, if any
- Time blocked, in seconds

- Name of the program requesting the transaction
- Name of the active transaction



Viewing SQL Server 11.0 Process Information

When you are connected to a release 11.0 SQL Server, the following additional information is available:

- Oldest transaction
- Transaction start time

The oldest transaction, marked with a check in the Oldest Transaction column, appears at the top of the transaction list. When you select the oldest transaction in the list, its start time is included in the Transaction Start Time box below the list.

Killing a Process

To kill a process:

1. Select the process.
2. Click the Kill button. SQL Server Manager kills the process, removes its listing from the dialog box, and refreshes the dialog box.
3. Click the OK button to dismiss the dialog box.

Enabling Communication with Other Servers

SQL Server Manager allows you to configure the SQL Servers you administer so that a user connected to SQL Server can request execution of a stored procedure on another SQL Server. The results of these requests, called **remote procedure calls (RPCs)**, are returned to the calling process running on the SQL Server to which the user is connected.

To enable execution of RPCs, the following setup is necessary:

- The **interfaces file** of the local SQL Server must have an entry for the remote SQL Server. The interfaces file for a SQL Server, set up when the server is installed, lists the name and address of all servers the SQL Server can access.

Interfaces file editing is done outside SQL Server Manager; for details, see the *System Administration Guide Supplement* for the operating system on which the local SQL Server runs.

- The SQL Server where the remote stored procedure resides must be defined as a remote server to the local SQL Server.
- A user with a login to the local SQL Server must be allowed access to the remote SQL Server.
- The remote login name must be a user of the database where the remote stored procedure is located and must have permission to execute the procedure.

To make the remote login name a user in the remote database, connect to the remote SQL Server, create the user in the appropriate database, and give the user execute permission on the stored procedure. For details, see Chapter 8, “Controlling Access.”

This section describes how to create and display information about a remote server and how to give local SQL Server logins access to the remote server. It also describes how to set options that govern remote server connections.

Creating a Remote Server

To have access to a remote server when connected to a local SQL Server, you must create a definition of the remote server on the local server.

Privileges	Only a System Security Officer can create a remote server.
------------	--

To create a remote server:

1. Select the icon of the SQL Server on which to create the remote server definition. In this discussion, the SQL Server you select is the local server.
2. From the Server menu, choose Create; then, choose Remote Server. The Create Remote Server dialog box opens.

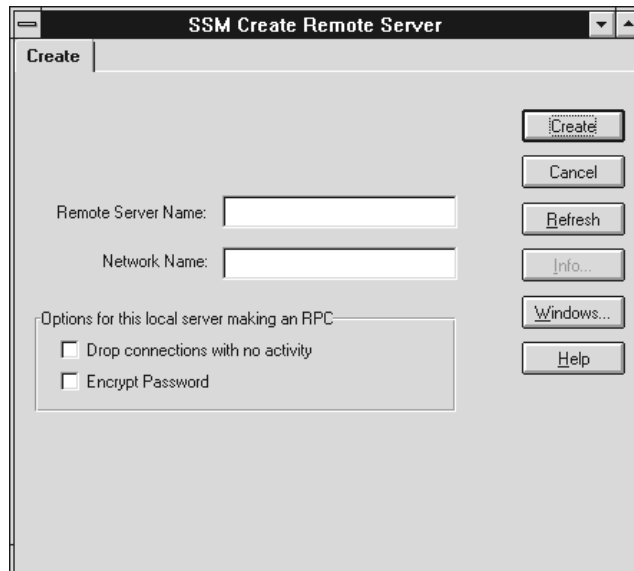


Figure 4-9: Create Remote Server dialog box

3. In the Remote Server Name box, enter the name of the remote server. This can be the name of the remote server in the interfaces file, or it can be an alias that you assign for local reference.
4. In the Network Name box, enter the network name of the remote server. This is the remote server name as it appears in the interfaces file. If you entered an alias in the Remote Server Name box, Network Name is required; otherwise, you can leave it blank.
5. Optionally, adjust the remote server options specified in the box labeled Options for This Local Server Making an RPC. Remote server options specify how the SQL Server to which you are connected accesses the remote server you are defining:
 - Drop Connections with No Activity—if selected, the site handler at the remote server drops the connection from the

local SQL Server when there has been no remote login activity for one minute.

Privileges	Only a System Administrator can set the Drop Connections with No Activity remote server option.
------------	---

- **Encrypt Password**—if selected, the local SQL Server encrypts its login passwords when connecting to the remote server, using an encryption key provided by the remote server.

Privileges	Only a System Security Officer can set the Encrypt Password remote server option.
------------	---

6. Click **Create**. SQL Server Manager creates the remote server and adds its icon to **Voyager** for the SQL Server to which you are connected.

Shortcuts



Select the remote server container icon for the appropriate SQL Server; then, select the **Create** toolbar button.



Click the right mouse button over the Remote Server container icon for the appropriate SQL Server. Choose **Create** from the shortcut menu.



Click the right mouse button on the SQL Server icon. Choose **Create** from the shortcut menu; then, choose **Remote Server**.

Deleting a Remote Server

To delete a remote server:

1. Select the icon of the server to delete.
2. From the **Remote Server** menu, choose **Delete**.
3. In the confirmation dialog box, click **Yes**. SQL Server Manager deletes the remote server and its login mappings.

Shortcuts



Select the remote server container icon for the appropriate SQL Server; then, select the Delete toolbar button.



Click the right mouse button over the Remote Server container icon for the appropriate SQL Server. Choose Delete from the shortcut menu.

Managing Remote Server Logins

When a login on a local SQL Server requests access to a stored procedure on a remote server, the login assumes the identity and access privileges of a login on the remote server. An administrator with the System Security Officer role controls how this mapping between logins occurs and whether the password of a login making an RPC from another server is checked.

Logins can be mapped from a calling server to a receiving server in several ways:

- Logins from the calling server keep their own names on the receiving server. For this scheme to work, you must set up a login on the receiving server for each login that will be issuing RPCs from the calling server.
- All logins from the calling server are mapped to a single login on the receiving server, taking on the identity and privileges of that single login.

► Note

Mapping more than one calling server login to a single receiving server login is not recommended, as it reduces individual accountability on the receiving server. Audited actions can be traced only to the receiving server login, not to the individual logins on the calling server.

- Each login from the calling server is mapped to a specific login on the receiving server. You can use this option in combination with either of the others; when you do, the explicit mappings set up for

logins under this option override the mappings established under the other methods.

Privileges	Only a System Security Officer can establish remote server login mappings.
------------	--

Login mapping is managed from the perspective of the remote server. When you set up login mappings, you specify how logins from the calling server (the SQL Server to which you are connected) map to logins on the server defined as remote. To set up remote server login mapping:

1. Select the icon of the remote server.
2. From the Remote Server menu, choose Properties...
3. Select Login Mappings. The display changes to the Login Mappings tab.

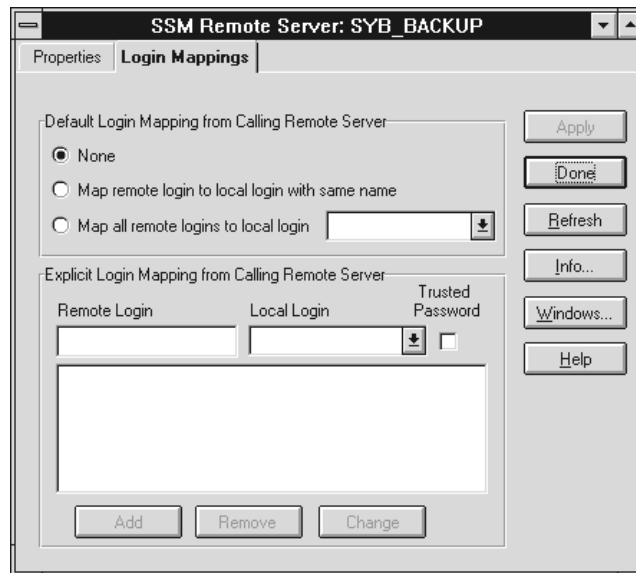


Figure 4-10: Remote Server Login Mappings tab




4. Select a login mapping scheme, or select None to indicate that no automatic mapping scheme is in effect:
 - To allow each calling server login to keep the same login name on the receiving server, select Map Remote Login to Local Login with Same Name.

- To map all calling server logins to a single receiving server login, select Map All Remote Logins to Local Login. From the drop-down list next to this option, select the receiving server login.
 - To specify that no automatic mapping is in place, select None. With this option, only logins explicitly mapped to logins on the receiving server are allowed access.
5. Optionally, set up explicit mappings between calling server logins and receiving server logins. When you set up explicit mappings in addition to an automatic mapping scheme, the explicit mappings override the automatic ones. When you specify None as the default login mapping, only logins explicitly mapped to a receiving server login have access to the receiving server.

To set up an explicit mapping for a calling server login:

1. Enter the name of the calling server login in the Remote Login box.
 2. Select a receiving server login to map to in the Local Login drop-down list.
 3. To specify that the calling server login should obtain access without password checking, select Trusted Password.
 4. To add the mapping to the list, click Add. Repeat this step as often as necessary to set up the required explicit login mappings. To remove a mapping from the list, select the mapping and click Remove. To change a mapping, select it. When the details appear in the boxes above the list, make adjustments and click Change.
6. When you are satisfied with all login mapping details, click Apply.

Shortcuts

	Double-click the remote server icon.
	Select the remote server icon; then, select the Properties toolbar button.
	Click the right mouse button over the remote server icon. Choose Properties from the shortcut menu.

Displaying Remote Server Details

To display information about an existing remote server:

1. Select the remote server icon.
2. From the Remote Server menu, choose Properties...

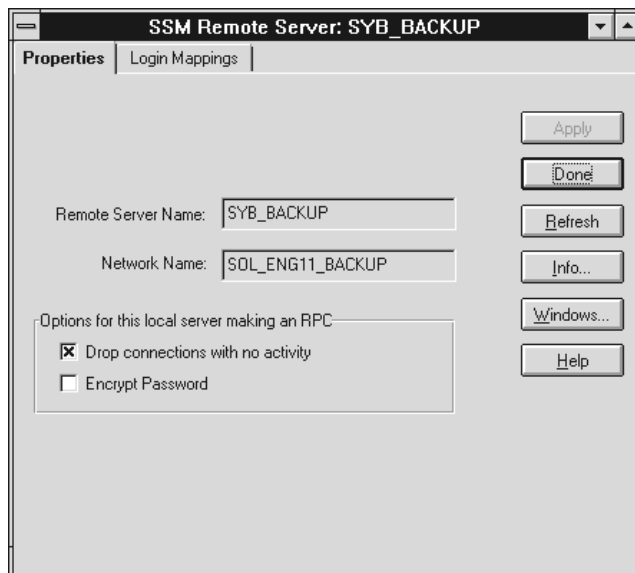





Figure 4-11: Remote Server Properties dialog box

From the Properties tab you can set remote server options or navigate to the Login Mapping tab to update login mappings.

Shortcuts

	Double-click the remote server icon.
	Select the remote server icon; then, select the Properties toolbar button.
	Click the right mouse button over the remote server icon. Choose Properties from the shortcut menu.

Generating and Executing SQL Server DDL Scripts

Data Definition Language (DDL) consists of Transact-SQL commands combined into scripts that capture the schema definition of a SQL Server or database and its objects. You can run these scripts to recreate a SQL Server configuration or database in its original location or on a different SQL Server.

SQL Server Manager allows you to generate DDL scripts automatically and then to execute them. You can also use any text editor to customize DDL scripts to reflect the environmental variations of a SQL Server different from the one on which you created them.

When you generate or execute DDL for a SQL Server, SQL Server Manager creates or executes the scripts for the following:

- The SQL Server configuration
- All logins, database devices, dump devices, remote servers, and, in SQL Server 11.0, named data caches defined for the SQL Server
- Optionally, all databases on the SQL Server, along with their objects

To create or execute DDL for a single database, use the commands described in this section, but execute them from the Database menu. For details, see “Generating and Executing Database DDL Scripts” on page 7-13.

Generating SQL Server DDL

To generate SQL Server DDL:

Privileges Only a System Administrator can generate SQL Server DDL.

1. In Voyager, select the icon of the SQL Server for which to generate DDL scripts.
2. Choose Generate DDL... from the Server menu. The Server Generate DDL dialog box opens.

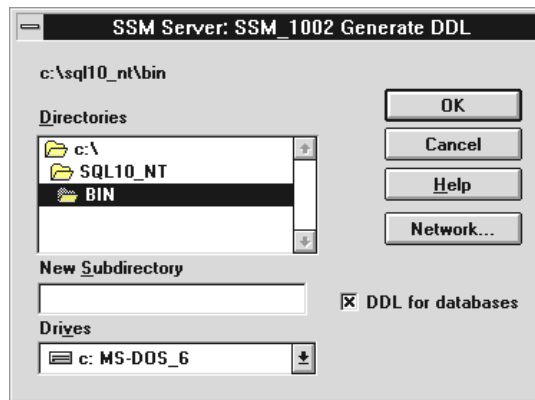


Figure 4-12: Server Generate DDL dialog box

3. Enter the following information:
 - For Directories, select the location in the directory hierarchy where you want SQL Server Manager to create the top-level DDL script directory.
 - For New Subdirectory, type the name of the top-level directory in which you want SQL Server Manager to create SQL Server DDL scripts. The default is the *ddl* subdirectory in the root directory of your Sybase installation.
 - For Drives, select the letter of the disk drive where you want to store DDL scripts. To access an unconnected drive, click the Network... button to display the Connect Network Drive dialog box.
4. To generate DDL for all the databases on the Server, select DDL for Databases.

- Click the OK button. SQL Server Manager creates the appropriate SQL Server DDL scripts in the directory location you specified. For a list of SQL Server DDL script files, see “Examining and Editing DDL Script Files” on page 4-26.

Executing SQL Server DDL

Before executing DDL scripts, make any required modifications to the *.ddl* files for the current SQL Server. For example, the *devices.ddl* file contains operating system-specific information that may change for the SQL Server where you execute the DDL Scripts.

To execute SQL Server DDL:

Privileges	Only a System Administrator can execute SQL Server DDL.
------------	---

- In Voyager, select the icon of the SQL Server for which to execute DDL scripts.
- Choose Run DDL... from the Server menu. The Server Run DDL dialog box opens.

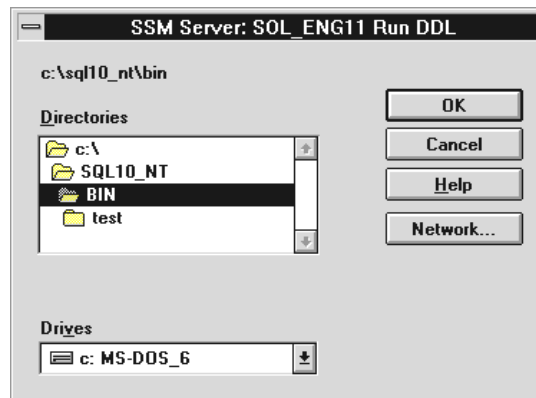


Figure 4-13: Server Run DDL dialog box

- From the Drives list, select the disk drive where the DDL scripts you want to use are stored. To access an unconnected drive, click the Network... button to display the Connect Network Drive dialog box.
- In the Directories list, select the directory named for the SQL Server where you generated the SQL Server DDL scripts. This

directory is one level **below** the directory you specified as New Directory in the Server Generate DDL dialog box when generating the scripts.

5. Click the OK button. SQL Server Manager executes the appropriate SQL Server DDL scripts. For a list of SQL Server DDL script files, see “Examining and Editing DDL Script Files” on page 4-26.

Examining and Editing DDL Script Files

When SQL Server Manager generates DDL scripts for a server, it places the scripts in the directory you specify in the Server Generate DDL dialog box. Within this directory, it places scripts in a subdirectory named *srv*, where *srv* is the name of the SQL Server for which the scripts were generated.

When SQL Server Manager generates database-level scripts, it creates a set of files for each database and places them in a subdirectory named for the database. The files in the *srv* subdirectory and any database subdirectories below it carry a *.ddl* extension.

The generated DDL scripts are plain text files. To view or edit them, use any text editor—for example, Windows or Windows NT Notepad.

Review the DDL files for applicability to the environment in which you plan to execute them. For example, if you plan to use SQL Server Manager to execute server-level DDL, you may need to edit the *devices.ddl* file, and if you use remote SQL Servers and logins, you will probably need to edit the *remote.ddl* file:

- The *devices.ddl* file contains statements that assign logical names to physical devices (tapes and disks) controlled by the host operating system. A physical device name on one machine may not be appropriate on another machine, or the device may not be available. Review the *devices.ddl* file, and replace machine-specific physical device names with device names applicable to the new machine.
- The *remote.ddl* file contains statements to configure remote servers and remote logins. What is remote on one SQL Server may be local on another. Update the server and login information in the *remote.ddl* file to reflect the remote servers and logins you want to access on the new machine.

► **Note**

DDL scripts created from a release 11.0 SQL Server do **not** execute successfully on a release 10.x SQL Server.

The following table lists the files in the *srv* subdirectory of the user-specified directory when the scripts were generated.

Table 4-1: Files created during generation of server-level DDL

File name	What the file contains
<i>stats.ddl</i>	Statistics about server configuration. SQL Server Manager uses this information when attempting to execute DDL scripts on another server.
<i>exec.ddl</i>	List of remaining server-level DDL files in correct execution order. At execution time, SQL Server Manager uses this file to determine which scripts to execute and in what order.
<i>config.ddl</i>	DDL to configure a server the way <i>srv</i> was configured when you generated DDL.
<i>devices.ddl</i>	DDL to create all logical devices known on <i>srv</i> at time you generated DDL. Logical device names are bound to same physical device names that existed on <i>srv</i> .
<i>database.ddl</i>	DDL to create all databases that existed on <i>srv</i> , with database options set as on <i>srv</i> .
<i>logins.ddl</i>	DDL to set up all user accounts (and roles) that existed on <i>srv</i> . To avoid placing passwords in an operating system file, SQL Server Manager creates user accounts with password set to "password" when you execute DDL.
<i>remote.ddl</i>	DDL to configure remote servers and remote logins.

The following table lists the files in each *db* subdirectory of the *srv* subdirectory, where *db* is the name of a database for which DDL scripts were generated.

Table 4-2: Files created during generation of database-level DDL

File name	What the file contains
<i>exec.ddl</i>	List of all other DDL files for <i>db</i> in the order they should be executed. SQL Server Manager uses this file at DDL execution time to determine which files to execute and in what order.
<i>segment.ddl</i>	DDL to create all segments and thresholds defined for <i>db</i> . Segments are created on same logical devices as on <i>srv</i> .
<i>udt.ddl</i>	DDL to create all user-defined datatypes in <i>db</i> and to bind defaults and rules to datatypes. Ownership of datatypes matches ownership in <i>db</i> on <i>srv</i> .
<i>user.ddl</i>	DDL to add users to <i>db</i> .
<i>groups.ddl</i>	DDL to create all groups in <i>db</i> .
<i>table.ddl</i>	DDL to create all indexes and tables in <i>db</i> , set permissions on tables, and bind defaults and rules to table columns. Tables are created on same segments on which they existed on <i>srv</i> . Table ownership matches ownership on <i>srv</i> .
<i>view.ddl</i>	DDL to create all views in <i>db</i> .
<i>default.ddl</i>	DDL to create all defaults defined in <i>db</i> on <i>srv</i> .
<i>rule.ddl</i>	DDL to create all rules defined in <i>db</i> on <i>srv</i> .
<i>proc.ddl</i>	DDL to create all user-defined stored procedures in <i>db</i> and to set permissions on them.
<i>trigger.ddl</i>	DDL to create all triggers defined on the tables in <i>db</i> .



Checking Consistency in SQL Server 11.0

When you are connected to a release 11.0 SQL Server, you can run the Database Consistency Checker (dbcc), a set of utility commands that check the logical and physical consistency of a database. When you run dbcc against a specified SQL Server, you can use it to check network I/O tasks or to execute any other valid dbcc command.

This section describes consistency checking for SQL Server. Additional dbcc options enable you to check the consistency of databases, tables, and indexes. For details, see Chapter 7, “Managing Databases and Segments” and Chapter 9, “Managing Database Objects.”

Privileges	A System Administrator can execute SQL Server dbcc commands.
------------	--

To run dbcc against a SQL Server:

1. Select the icon of the SQL Server.
2. From the Server menu, choose Consistency... The Server Consistency Check dialog box opens.

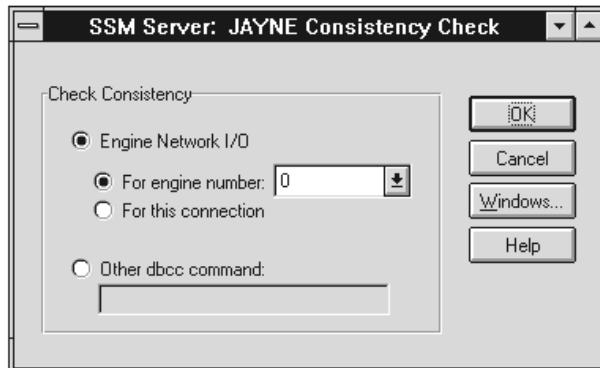


Figure 4-14: Server Consistency Check dialog box

3. To produce a listing of network I/O tasks, select Engine Network I/O. This selection executes the dbcc engine command option. Choose one of the following alternatives:
 - For Engine Number—performs a consistency check on the network I/O tasks currently assigned to a specified engine.

Select the engine number from the drop-down list that appears when you select the Engine Number option.

- For This Connection—performs a consistency check on the network I/O tasks for the engine servicing I/O for your user connection.
4. To execute a specified `dbcc` command, select Other `dbcc` Command. When you select this option, an edit box appears below the option button. Enter the text of the entire `dbcc` command option to execute, including the `dbcc` keyword.
 5. Click OK. SQL Server executes the `dbcc` command, and SQL Server Manager displays the results in the Output dialog box.

The following figure shows an example of a report produced with the Engine Network I/O selection.

```

SSM Output
DBCC execution completed. If DBCC printed error messages, contact a user with
System Administrator (SA) role.

Network affinity migration is enabled.

Tasks that are network affinitied to Engine 0:
  kpid   vsn   psn   status
  ----   --   ---   -
  3145776 1     11   sleeping(0xeal7ff70) setdataack
  3735609 2     22   sleeping(0xeal800c0) setdataack
  5308446 3     23   running(engine 0) setdataack

Total number of tasks that are currently network affinitied to engine 0 is 3

DBCC execution completed. If DBCC printed error messages, contact a user with
System Administrator (SA) role.
  
```

Figure 4-15: `dbcc` engine output

Shortcut



Click the right mouse button over the SQL Server icon. Choose Consistency... from the shortcut menu.

5

Managing Cache



What's in This Chapter

When you are connected to a release 11.0 SQL Server, you can use its **cache** management features. Cache is a portion of SQL Server memory allocated when a SQL Server is created. Its purpose is to temporarily hold information currently in use by SQL Server and to hold recently used data pages. Because it is faster to read from memory than from disk, holding information in cache improves performance.

SQL Server cache is divided between memory reserved for data and memory reserved for stored procedure execution plans. In SQL Server 11.0, you can partition the memory allocated for data cache (the default cache) into user-defined (named) data caches, and you can bind databases, indexes, or tables to a specific cache.

Within a named data cache, you can create buffer pools that are larger than the default size of 2K, enabling SQL Server to read larger amounts of data in a single I/O. Larger I/Os can enhance performance.

This chapter contains information on:

- Creating and deleting named data caches
- Displaying and modifying cache information
- Binding objects to a cache

Creating and Deleting a Named Data Cache

This section describes how to create and delete a named cache within the default data cache configured for a SQL Server.

Privileges	Only a System Administrator can create, modify, and delete a cache.
------------	---

Creating a Cache

To create a cache:

1. Select the icon of the SQL Server in which to create the cache.
2. From the Server menu, choose Create. Then, choose Cache from the cascading menu. The Create Cache dialog box opens.

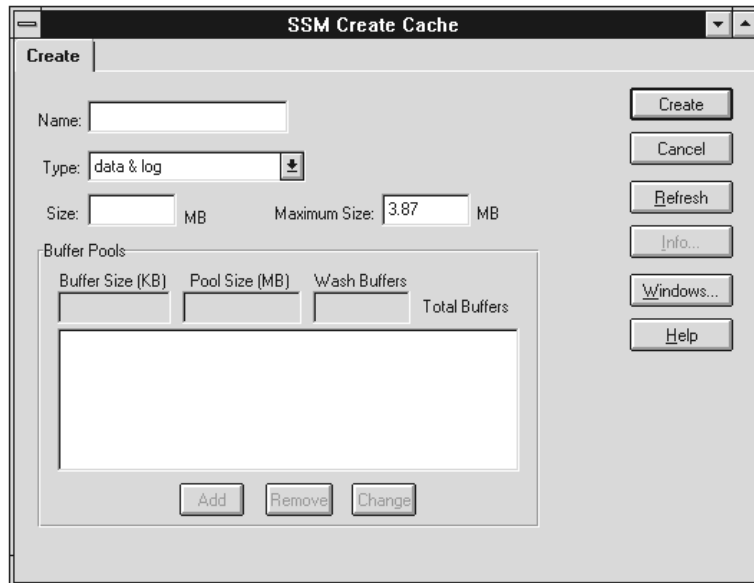


Figure 5-1: Create Cache dialog box

3. In the Name box, enter the name of the cache.
4. From the Type list, select one of the following:
 - To create a cache to hold both data and transaction log information, select “data & log.”
 - To create a cache to hold only transaction log information, select “log only.”
5. In the Size box, enter the size of the cache, in megabytes. The minimum size is .5MB. For reference, the Maximum Size box displays the amount of SQL Server memory available for named caches. This amount is based on the total amount of SQL Server memory configured for data cache and considers both the minimum size allowed for the default cache (.5MB) and the size of previously created active named caches.

► Note

Because creating a cache involves some overhead, the actual amount of available memory is slightly less than the amount in the Maximum Size box.

- Click **Create**. SQL Server creates the cache with a status of “Active (pending).” In this state, the cache is defined in SQL Server but cannot be used for holding data and cannot be bound to a database, index, or table until you reboot the server.

After you reboot SQL Server, the cache status becomes “Active,” and the cache is ready for use.

Shortcuts

Select the Cache container icon in the appropriate SQL Server, and select the **Create** toolbar button.



Click the right mouse button over the Cache container icon; then, choose **Create** from the shortcut menu.



Click the right mouse button over the SQL Server icon. Choose **Create** from the shortcut menu; then, choose **Cache**.

Deleting a Cache

To delete a cache:

- Select the icon of the cache to delete.
- From the **Cache** menu, choose **Delete**.
- In the confirmation dialog box, click **Yes**. SQL Server changes the cache status to “Delete (pending).” In this state the cache is marked for deletion but is not physically deleted until you stop and restart SQL Server.

Shortcuts



Select the icon of the cache to delete, and select the Delete toolbar button.



Click the right mouse button over the cache icon; then, choose Delete from the shortcut menu.

Displaying and Modifying Cache Information

The Cache Properties tab provides current information about a cache. In addition, it is the starting place for any modifications you make to a cache. These are the types of modifications you can make:

- Changing cache size or type
- Adding, deleting, and changing cache buffer pools
- Binding and unbinding databases or database objects to the cache

Displaying Cache Properties

To display cache properties:

1. Select the icon of the cache to display.
2. From the Cache menu, choose Properties...

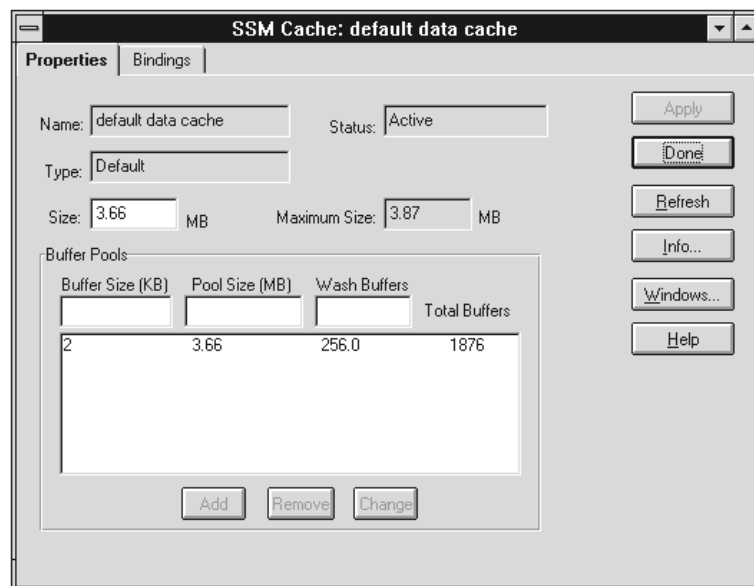
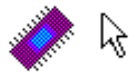


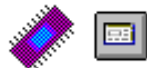
Figure 5-2: Cache Properties tab

Shortcuts

Double-click the cache icon.



Click the right mouse button over the cache icon; then, choose Properties... from the shortcut menu.



Changing Cache Size or Type

To change the size or type of a named cache:

1. Display the Cache Properties tab, as described in “Displaying Cache Properties” on page 5-5.
2. From the Type list, select one of the following:
 - To modify a cache to hold both data and transaction log information, select “data & log.”
 - To modify a cache to hold only transaction log information, select “log only.” This option is not available if the cache has any databases, indexes, or tables bound to it. To change a cache to a log-only cache, first unbind any nonlog bound objects from it. You cannot change the default cache to a log-only cache.
3. In the Size list, enter the new cache size. The new size is subject to the following constraints:
 - When you reduce the size, all space is removed from the default buffer pool. You cannot specify a size that reduces the default pool to less than the 512K minimum. For details, see “Defining Buffer Pools” on page 5-6.
 - When you increase the size, all new space is added in the default buffer pool. You cannot increase the size to an amount that exceeds the value in the Maximum Size box.

► **Note**

Because creating a cache involves some overhead, the actual amount of available memory is slightly less than the amount in the Maximum Size box.

4. Click Apply. For the new specifications to take effect, you must stop and restart SQL Server.

Defining Buffer Pools

When you first create a cache, all space is assigned to the default memory (buffer) pool consisting of buffers that are the default I/O size for the SQL Server (2K on most platforms). After creating a cache, you can create additional pools with buffers larger than the default. SQL Server I/Os for these pools are the size of their buffers—for example, 4K I/Os for a pool with 4K buffers. In a database with a high level of I/O activity, defining pools with buffers larger than the default can make I/Os more efficient.

► **Note**

Moving space between buffer pools within a cache takes effect immediately; a SQL Server restart is not required.

Each buffer pool you create is identified by the size of its buffers. The following sizes are allowed: 2K (the default on most SQL Servers), 4K, 8K, and 16K.

In any cache, you can have only one pool of each buffer size.

The following figure shows a data cache consisting of the default cache and two named caches. The default cache has only the original 2K pool. The *User_table* cache has a 2K pool and a 16K pool. The *Log* cache has a 2K pool, a 4K pool, and a 16K pool.

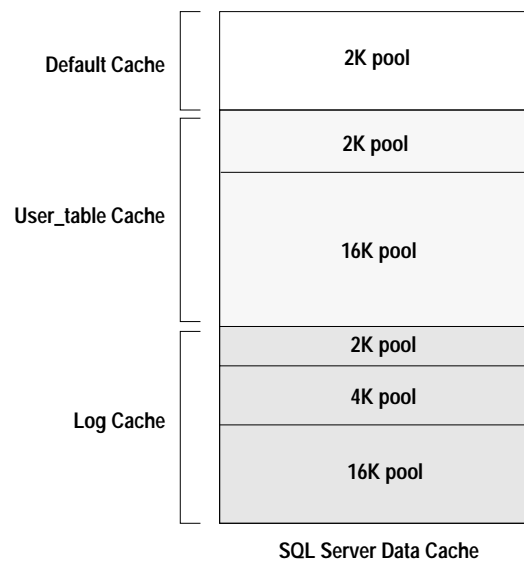


Figure 5-3: Default cache and two named data caches

When you create a buffer pool, you reassign space from the default pool. For example, if you create a data cache with 50MB of space, all of the space is initially assigned to the default pool. If you create a 30MB pool of 16K buffers, the default pool is reduced to 20MB. If you

remove a 4K, 8K, or 16K buffer pool, the space returns to the default pool.

Privileges	Only a System Administrator can create, modify, and delete buffer pools.
------------	--

Creating a Buffer Pool

You can define buffer pools on user-defined caches in Active status or on the default data cache.

To create a buffer pool:

1. Display the Properties tab as described in “Displaying Cache Properties” on page 5-5.
2. In the Buffer Size (KB) box, enter the size of the buffers in the pool. Allowable sizes are 4K, 8K, and 16K.
3. In the Pool Size (MB) box, enter the amount of space to reassign from the default buffer pool. The new pool must not reduce the 2K pool to less than 512K.
4. In the Wash Buffers box, enter the number of pool buffers to keep in the pool’s **wash area**. The default is 256 buffers. The wash area is a portion of each memory pool used to ensure that queries that need clean pages in a data cache can find them. When the number of dirty pages (pages that have been changed in cache) fills the rest of the buffers and begins to enter the wash area, SQL Server writes the data in the wash area to disk. When this write completes, the wash area is marked clean and is available for queries needing clean pages.

The minimum number of wash buffers you can specify is 10; the maximum number is 80% of the pool size.

5. To add the pool definition to the list of buffer pools, click Add. To remove a buffer pool, select the pool in the list and click Remove. To change a buffer pool, select the pool in the list, make changes in the boxes above the list, and click Change.
6. Click Apply.

Deleting a Buffer Pool

You can delete any buffer pool except the default pool. If you delete a 4K, 8K, or 16K buffer pool, the space returns to the default pool. To delete a buffer pool:

1. Display the Cache Properties tab, as described in “Displaying Cache Properties” on page 5-5.
2. In the Buffer Pools list, select the pool to delete, and click Remove.
3. Click Apply.

Binding Objects to a Cache

You can bind databases, indexes, or tables to a named data cache. Binding an object to a cache allows you to specify where recent reads from an object are held in memory and to take advantage of cache buffer pools that allow larger I/Os than the default. When you bind an object to a cache:

- SQL Server clears any pages for the object currently in memory.
- SQL Server reads pages for the object into the bound cache when subsequent queries need them.

Cache bindings take effect immediately; they do not require a SQL Server restart.

Privileges	Only a System Administrator can bind and unbind objects to a cache.
------------	---

Rules for Binding

The following rules govern cache bindings:

- You can bind objects only to an active cache.
- You can bind a database or database object to only one cache.
- You can bind a database to one cache and bind individual tables or indexes in the database to other caches. You can bind an index and the table it references to different caches.
- You cannot bind the *master* database, the system tables in *master*, or the indexes on the system tables in *master* to a named cache.

However, you can bind nonsystem tables in *master* and their indexes.

- If you bind a database to a cache and do not bind its objects, SQL Server treats the objects as if they were bound to the database cache.
- If you bind neither a database nor its objects to a cache, SQL Server uses the default cache for their reads.
- When you delete a database, table, or index, SQL Server deletes all of the associated cache bindings.

Creating a Binding

To bind a database or object to a cache:

1. Display the Properties tab as described in “Displaying Cache Properties” on page 5-5.
2. Click Bindings. The display changes to the Bindings tab. Objects available for display are in the Available Objects column. Objects previously bound to the cache are in the Bound Objects column. Only the bound objects that meet the filtering criteria specified under the Filter label appear in the Available Objects and Bound Objects columns. The list in the All Bound Objects group box shows bound objects of all types and in all databases.

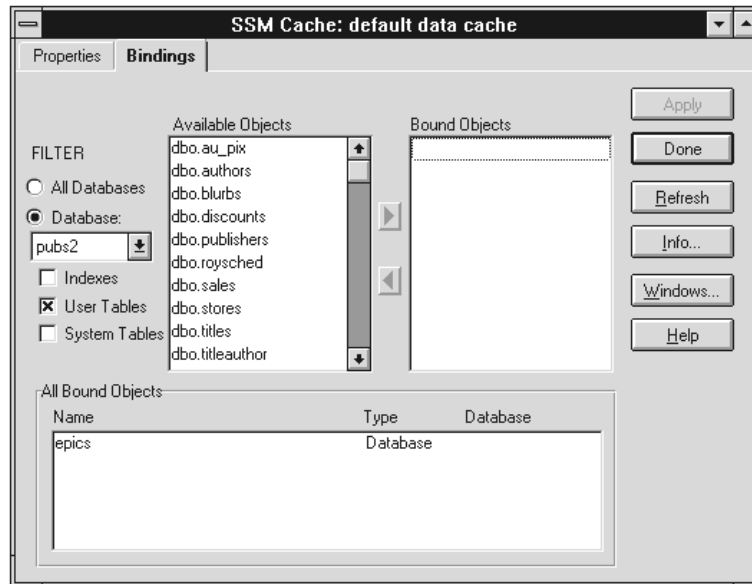
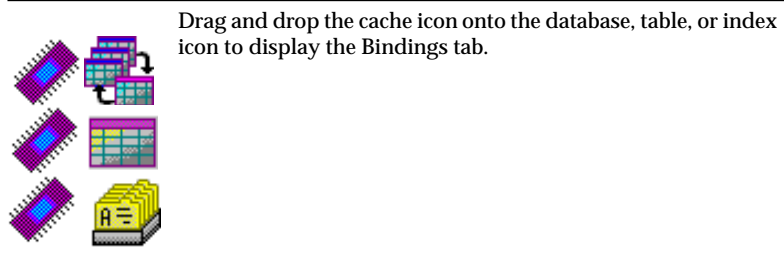


Figure 5-4: Cache Bindings tab

3. Filter the Available Objects list so it includes the objects you want to bind:
 - To display only database names, select the All Databases option button.
 - To display the objects in a single database, select the Database option button, and select the database name from the corresponding list. Within the selected database, clear the boxes for the objects (indexes, system tables, or user tables) you do not want to see.
4. To bind objects to the cache, move them from the Available Objects list to the Bound Objects list. To move an object, select the object in the list and click the right-pointing arrow.
5. Click Apply. Note that you must apply the binding of objects for one filter setting before changing the filtering to another setting. For example, if you select the All Databases option and bind a database to a cache, you must click Apply before setting the filtering to show the tables in a specific database in preparation for binding them.

Shortcut



Deleting a Binding

To unbind a database or object from a cache:

1. Display the Properties tab as described in “Displaying Cache Properties” on page 5-5.
2. Click Bindings. The display changes to the Bindings tab.
3. In the All Bound Objects list, find the object to unbind. Use the information in the Type and Database columns to set the filtering and display the object in the Bound Objects column:
 - If the object to unbind is a database, select the All Databases option button.
 - If the object to unbind is an index or table, select the Database option button, select the database from the drop-down list, and select the appropriate check box.
4. To unbind an object, move it from the Bound Objects list to the Available list. To move an object, select the object in the list and click the left-pointing arrow.
5. Click Apply.

6

Managing Physical Resources

What's in This Chapter

SQL Server Manager supports SQL Server defaults for many aspects of storage management—where to place databases, tables, and indexes and how much space to allocate for each of them. However, as a System Administrator running SQL Server Manager, you can also override SQL Server storage management defaults and customize the aspects of storage management that you require. This allows you to manage storage allocations for the level of performance and control you need in your environment.

Privileges	Only a System Administrator can perform the activities described in this chapter.
------------	---

This chapter describes how to use SQL Server Manager to:

- Create, modify, and delete database devices and to map them to physical resources in your environment
- Maintain a pool of default devices from which SQL Server allocates space when no specific device is requested
- Mirror database devices to ensure nonstop recovery in case of physical disk crashes

Strategies for Recovery and Performance

To make good storage allocation decisions as a System Administrator, you need to balance your requirements for recovery capability and performance. SQL Server Manager facilitates the following strategies:

- Disk mirroring for nonstop recovery in the event of a device crash.
- Maintaining logs on a separate physical device. This enables you to recover by loading a database backup and applying transaction log records.

This chapter discusses each of these strategies in “Using Disk Mirroring” on page 6-10 and “Storage Management Considerations and Tips” on page 6-16.

Creating and Deleting Devices

This section describes how to create and delete logical database and dump (backup) devices that map to your physical storage media.

Considerations for Creating a New Device

Before creating a new device, consider the following:

- When you create a database or dump device, SQL Server adds a row to the *sysdevices* table in the *master* database. Therefore, before creating a new device, you should back up the *master* database.
- Before creating a new database or dump device, make sure you have enough disk space on the physical device.
- Be sure that the “sybase” operating system account has write access on the physical device.
- After you create a new database or dump device, back up the *master* database again.

Creating a Database Device

A **database device** is a disk device used for storage of a database and its transaction log. As shown in the following figure, a database device can be all or a portion of a physical disk device, and multiple database devices (*data_dev1*, *log_dev1*) can exist on a single physical disk device (*disk_1*).

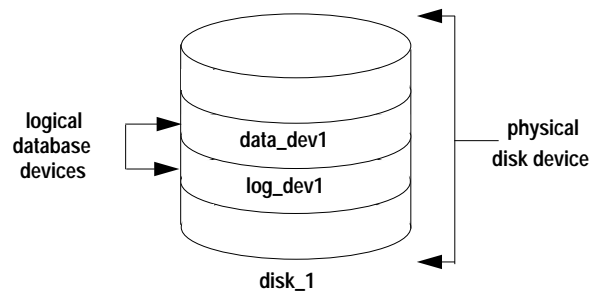


Figure 6-1: Logical database devices on a physical disk device

A SQL Server installation includes the creation of a master device for the *master*, *model* and *tempdb* databases and another device for the storage of the *sybssystemprocs* database. The master device, *d_master*, is the initial default storage device for all user-defined databases. As the master device stores the most critical SQL Server system tables, it is a good idea to:

- Designate other devices as default storage devices, and remove the default designation from the master device. “Changing Default Device Designation” on page 6-16 describes how.
- For ease of recovery, reserve the master device for the system databases, and create additional devices for your application databases.

To create a database device:

1. Select the icon for the SQL Server on which the device will reside.
2. From the Server menu, choose Create; then, choose Database Device from the cascading menu. The Create Database Device dialog box opens.

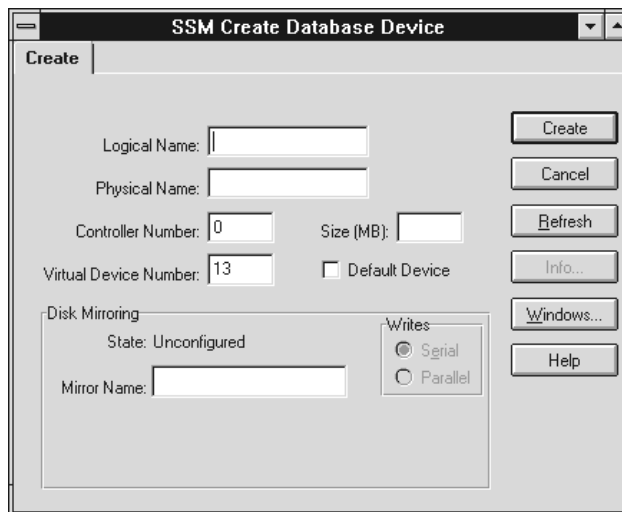
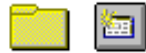


Figure 6-2: Create Database Device dialog box

3. Enter the following information:
 - For Logical Name, enter the name you want to assign to the device.

- For Physical Name, enter the full path name of the physical device in your environment to which to map the logical name. On a UNIX host, this can be the name of a raw partition or operating system disk file.
 - For Controller Number, enter the number of your disk controller, or accept the default.
 - For Size, enter the device size in megabytes if your physical device is a file. If the physical device is a raw partition, enter the size of the partition.
 - For Virtual Device Number, SQL Server Manager provides the next available sequential number. You may enter any unused device number from 1 through 1 less than the value of the devices SQL Server configuration variable.
4. If you want the device to be a default device, check the Default Device box. SQL Server uses a **default device** for storage of a database created with no specification of the device to store it on. For more information on default devices, see “Changing Default Device Designation” on page 6-16.
 5. If you want to mirror the device, enter the mirror specifications in the Disk Mirroring group box. When you **mirror** a device, SQL Server duplicates all writes to one device on a second device (the mirror device) that you specify. For more information, see “Using Disk Mirroring” on page 6-10. To mirror the device:
 - For Mirror Name, enter the physical name of the mirror device.
 - Specify whether you want writes to be serial or parallel by selecting the Serial or Parallel radio button.
 6. Click Create. SQL Server creates the database device, and a new database device icon appears in Voyager.

Shortcuts



Select the database device container icon for the appropriate SQL Server. Select the Create toolbar button.



Click the right mouse button over the database device container icon for the appropriate SQL Server. Choose Create from the shortcut menu.



Click the right mouse button over the SQL Server icon. Choose Create from the shortcut menu; then, choose Database Device.

Deleting a Database Device


To delete a database device:

1. Select the icon of the device to delete.
2. Choose Delete from the Database Device menu.
3. Confirm the deletion by clicking Yes in the confirmation dialog box.
4. If the device is mapped to an operating system disk file, delete the file to make the space available. This is not necessary if the device is mapped to a raw partition.


► **Note**

You cannot delete a device that has a database on it.

Shortcuts

-
- 

Select the icon of the database device to delete; then, select the Delete toolbar button.

 - 

Click the right mouse button over the database device icon to delete; then, choose Delete from the shortcut menu.
-

Displaying and Modifying Device Information

After creating a database device, you can examine or update its properties with the Database Device dialog box. You can also use the Database Device dialog box to get information on the databases allocated on the device.

Examining Database Device Properties

To display a database device dialog box:

1. Select the device's icon in the Voyager window.
2. From the Database Device menu, choose Properties.

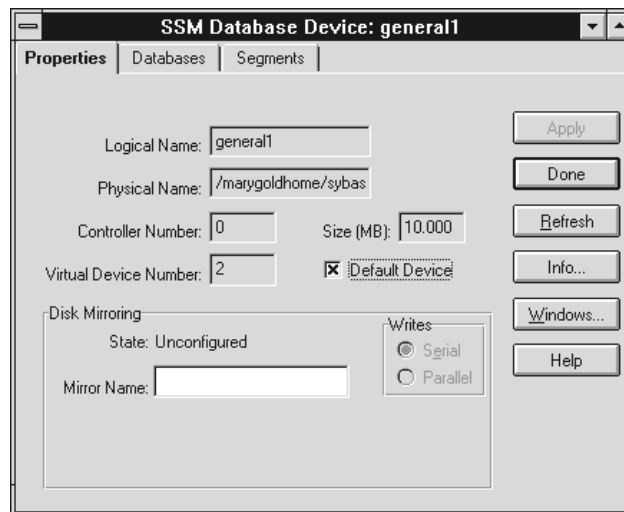
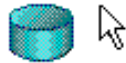


Figure 6-3: Properties tab of Database Device dialog box

Shortcuts

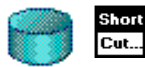
Double-click the database device icon.



Select the database device icon; then, select the Properties toolbar button.



Click the right mouse button over the database device icon; then, choose Properties... from the shortcut menu.



Examining Allocations on a Database Device

The Database Device dialog box shows which databases have allocated storage space on the device and allows you to navigate to those databases. This information makes up the Databases tab of the dialog box. To display the Databases tab:

1. Open the Database Device dialog box, as described in “Examining Database Device Properties” on page 6-6.
2. Click the Databases tab. The dialog box display changes to the Databases tab.

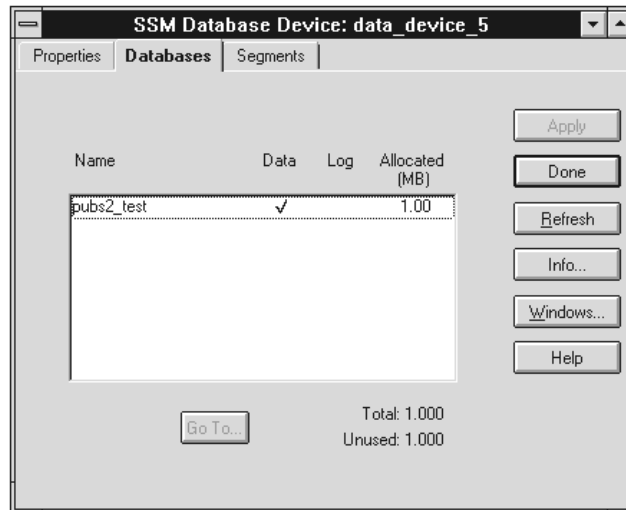


Figure 6-4: Databases tab of Database Device dialog box

The Databases group box shows:

- The name of each database allocated to the device
- The way the database is using the device: for its data, its transaction log, or both its data and its transaction log
- The amount of space allocated on the device for each database
- The total amount of allocated and unallocated space on the device

Navigating to a Database

To display the Database Dialog box for any database allocated on the database device:

1. Select the database by clicking its row in the list.
2. Click the Go To... button. The dialog box for the selected database opens.

Shortcut

Double-click the database name.

**Examining Segment Mapping on a Database Device**

For a more detailed look at the allocation of databases on a database device, display the Segments tab of the Database Device dialog box. The Segments tab shows the database segments mapped to the device. A **database segment** is a named portion of the database device storage allocated to a particular database. For information on database segments, see “Creating and Deleting Segments” on page 7-23. To display the Segments tab:

1. Open the Database Device dialog box, as described in “Examining Database Device Properties” on page 6-6.
2. Select the Segments tab. The dialog box display changes to the Segments tab.

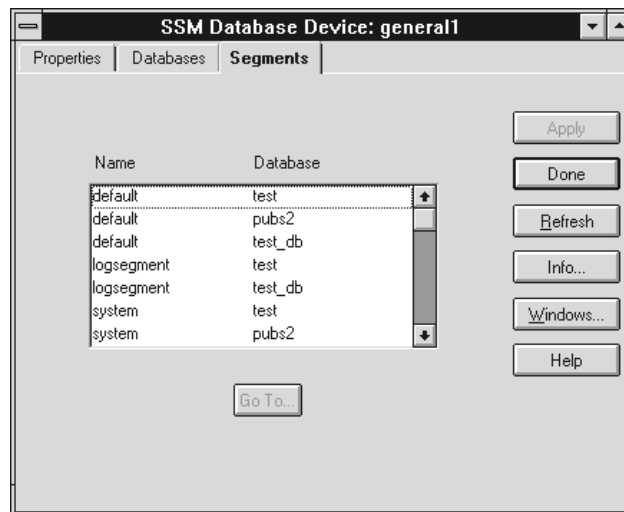


Figure 6-5: Segments tab of Database Device dialog box

Navigating to a Segment

To display the Segment Dialog box for any database segment mapped to the database device:

1. Select the segment by clicking its row in the list.
2. Click the Go To... button. The dialog box for the selected segment opens.

Shortcut

Double-click the segment row in the list.



Using Disk Mirroring

Mirroring is the duplication of the contents of an entire database device. When you set up a mirror device, all writes to the original device also go to the mirror device, and SQL Server reads from the original device or the mirror device based on efficiency. If one disk of a mirrored pair fails during a read or write, SQL Server automatically switches to the other disk and continues. The following figure illustrates disk mirroring.

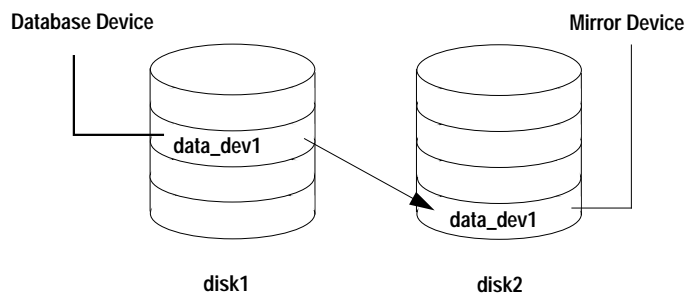


Figure 6-6: Database device on disk1 mirrored on disk2

Disk mirroring provides a large benefit in recovery capability:

- It prevents downtime due to disk failures.
- It ensures full, nonstop recovery.

- It can improve read performance because either device is available as a source.

Disk mirroring also carries a cost:

- It consumes additional disk storage resources.
- It slows writes because they are duplicated.

Sybase highly recommends using mirror devices if you have the additional resources available, in spite of the cost in resources and performance.

► **Note**

Mirroring takes place for a database device, not for a database. To mirror the activities of a database that is allocated across several devices, you must mirror all of those devices.

The following sections describe how to enable and disable disk mirroring.

Adding Disk Mirroring

“Creating a Database Device” on page 6-2 describes how to add disk mirroring when creating a database device. To add disk mirroring to an existing device:

1. Open the Database Device dialog box by selecting the device’s icon and choosing Properties... from the Database Device menu.
2. To mirror the device:
 - For Mirror Name, enter the physical name of the mirror device.
 - Specify whether you want writes to be serial or parallel by selecting the Serial or Parallel radio button.
3. Click Apply to modify the database device. The value of the State field changes from “Unconfigured” to “Enabled.”

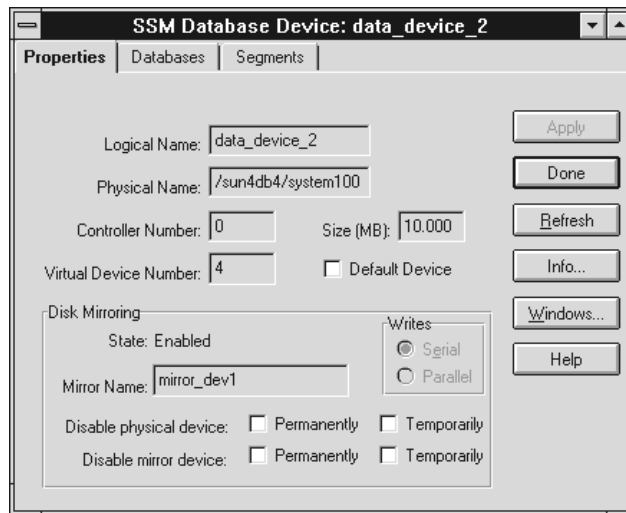








Figure 6-7: Database Device dialog box with disk mirroring enabled

Shortcuts

-
- 

Double-click the database device icon.

 - 

Select the database device icon; then, select the Properties toolbar button.

 - 

Click the right mouse button over the database device icon; then, choose Properties... from the shortcut menu.

Disabling and Re-Enabling Disk Mirroring

SQL Server Manager allows you to discontinue disk mirroring temporarily or permanently and to re-enable mirroring with a mirror device that you have temporarily disabled.

Disabling Disk Mirroring

1. Open the Database Device dialog box by selecting the device's icon and choosing Properties... from the Database Device menu.

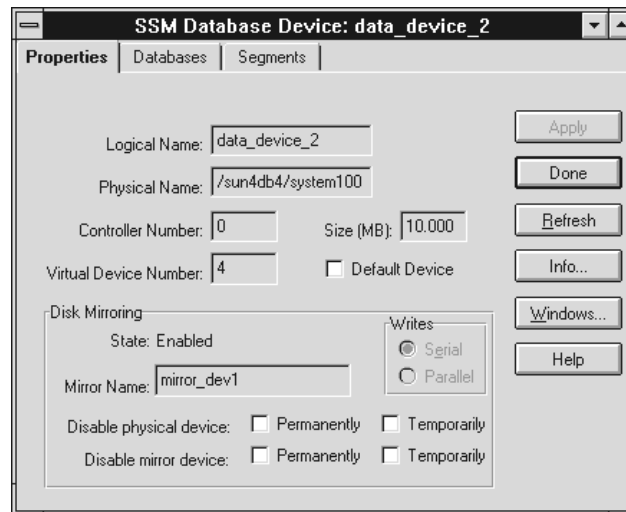


Figure 6-8: Disk mirroring enabled

2. Specify whether to disable disk mirroring temporarily or permanently by checking the Temporarily or Permanently box:
 - To disable writes to the **original database device**, select the Temporarily or Permanently box next to the Disable Physical Device label.
 - To disable writes to the **mirror device**, select the Temporarily or Permanently box next to the Disable Mirror Device label.

The following table summarizes the effects of temporarily or permanently disabling mirroring

Table 6-1: Effects of disabling mirroring

	Temporarily	Permanently
Database device	SQL Server suspends writes to the database device and directs I/Os only to the mirror device.	SQL Server changes the physical name of the database device to that of the mirror device and disables mirroring.
Mirror device	SQL Server suspends writes to the mirror device and directs I/Os only to the database device	SQL Server disables mirroring. Before you can re-establish mirroring, you must remove the operating system file that contains the mirror.
State	Disabled	Unconfigured

3. Click OK. SQL Server disables mirroring according to your specifications and updates the information in the Properties tab of the Database Device dialog box.

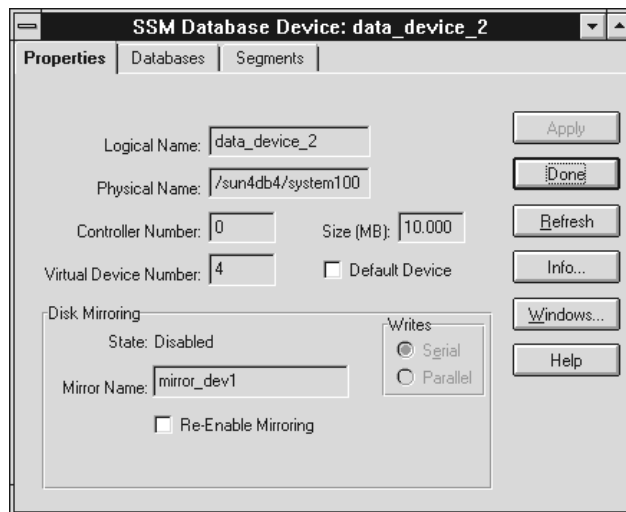
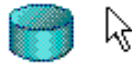


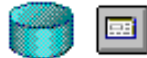
Figure 6-9: Database device with mirroring disabled

Shortcuts

Double-click the database device icon.



Select the database device icon; then, select the Properties toolbar button.



Click the right mouse button over the database device icon; then, choose Properties... from the shortcut menu.



Re-Enabling Disk Mirroring

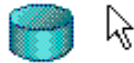
1. Open the Database Device dialog box by selecting the device's icon and choosing Properties... from the Database Device menu.
2. Check the Re-Enable Mirroring box.
3. Click Apply to modify the database device. SQL Server re-enables mirroring with the mirror device specified in the Disk Mirroring group box.

► **Note**

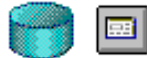
If you permanently disabled a mirror device for which the physical device is an operating system file, you cannot re-enable mirroring using the same physical file name unless you first manually delete the operating system file for the disabled mirror device.

Shortcuts

Double-click the database device icon.



Select the database device icon; then, select the Properties toolbar button.



Click the right mouse button over the database device icon; then, choose Properties... from the shortcut menu.



Storage Management Considerations and Tips

System Administrators must make many decisions regarding the physical allocation of space to SQL Server databases. Two major considerations that govern these decisions are recovery and performance. This section describes some strategies to consider.

Managing for Recovery

This section suggests strategies that can contribute to a clean recovery in the case of disk failure.

Changing Default Device Designation

When you create a database, you can specify the database device on which to create it. If you do not specify a device, SQL Server creates the database on a device that has been designated as a **default** device. If you have multiple default devices, SQL Server uses them in alphabetical order until each is full.

The master device created during a SQL Server installation holds the three system databases: *master*, *model*, and *tempdb*. This device is initially designated as a default device.

► Note

Sybase strongly recommends that you remove the default designation from the master device to prevent its getting cluttered and to make recovery easier in the case of a disk failure.

To change the default designation of a device:

1. Open its Database Device dialog box, as described in “Examining Database Device Properties” on page 6-6.
2. Select or clear the Default Device check box. Selecting the box makes the device a default device; clearing the box removes the default designation.

Using Disk Mirroring

If you mirror each database device containing a database allocation, and each mirror device is on a different physical device from the database device, you can ensure database recovery without downtime. Any other scenario requires some amount of recovery from backups.

Separating Data and Transaction Log

If you do not mirror your database devices, the next line of defense is to keep data and transaction logs separate. If a database device fails, you can recover by loading from the most recent database backup and applying the transaction log backups from the time of the database backup until the time of the failure.

To separate data and transaction logs, create multiple database devices before creating a database. When you create the database, you can specify that the space you are allocating on one or more devices is for log only. SQL Server automatically makes any other devices allocated for the database into data-only devices. For details, see “Creating a Database” on page 7-3.

Managing for Performance

Separation is the key to maximizing performance through storage management. The following separations can improve performance:

- Placing a table on one physical disk and nonclustered indexes on another ensures that physical reads and writes are faster, since the

work is split between two disk drives. For details, see Chapter 9, “Managing Database Objects.”

- Splitting large tables across two physical disks can improve performance, especially for multi-user applications.
- Placing a database or object on one physical disk and its mirror on another reduces read time, since both devices are active. On the other hand, mirroring a device increases the time needed to write, because SQL Server writes transactions to both the database device and its mirror.

If you use SQL Server Monitor™ at your installation, you may find it useful to run it with your SQL Server to examine how performance is affected by changes in physical storage allocations.

7

Managing Databases and Segments

What's in This Chapter

After creating database storage devices, you can create user databases and allocate them on database devices. You can also create database segments, which enable you to control the placement of objects on the devices. This chapter describes how to:

- Create and delete a user database and allocate storage for it
- Transfer database ownership to a different user
- Generate and execute Data Definition Language (DDL) scripts
- Change database options
- Issue a manual database checkpoint
- Create, extend, delete, and add thresholds to database segments
- Plan and perform database backup and recovery operations
- Execute Database Consistency Checker (dbcc) commands

Creating and Deleting a User Database

This section describes how to create and delete a user database.

Considerations for Creating a New Database

Before creating a database, decide:

- What size to make the database
- Where to place the database and whether there is sufficient space
- Whether you will store the transaction log on a device separate from the data

Estimating Database Size

When estimating database size, keep in mind that it is easy to increase the size of a database but difficult to decrease. Decreasing the size requires deleting the database and then recreating it.

To estimate the size of a database, consider:

- Anticipated size of its largest objects—tables and indexes

- Space for planned views, stored procedures, defaults, rules, and triggers
- Size of the transaction log
- Additional space for expansion based on anticipated activity

The SQL Server system procedure `sp_estspace` helps you estimate table and index space requirements based on the definition of a specific table. To use `sp_estspace` from SQL Server Manager:

1. On a nonproduction database, create the tables and indexes you anticipate the new database will contain. For information on creating tables and indexes, see Chapter 9, “Managing Database Objects.”
2. Open a `wisql` session with SQL Server so you can run the `sp_estspace` procedure. To run `wisql`, choose the `ISQL...` command from the Tools menu. For more information, see “Querying SQL Server with `wisql`” on page 11-12.
3. Run `sp_estspace` for each table, and add the results. For details on running `sp_estspace`, see the *SQL Server Reference Manual*.

The size of the transaction log depends on the type and quantity of transactions and the frequency of backups. As a starting point, allocate 10-25% of the overall size of the database for the transaction log.

Deciding Where to Allocate Storage

Consider the following storage options for a new database:

- You can store the database on a single database device or on multiple database devices.
- You can specify the amount of storage space to allocate on each device.
- You can store the transaction log on the same database device or on a different database device from the data. Storing the transaction log on a separate database device is highly recommended.

Creating a Database

Before creating a database, make sure enough space is available on the database devices you plan to use.

Privileges	Only a System Administrator or a user with explicit permission can create a database.
------------	---

To create a database:

1. Select the icon of the SQL Server in which to create the database.
2. From the Server menu, choose Create; then, choose Database from the cascading menu. The Create Database dialog box opens.

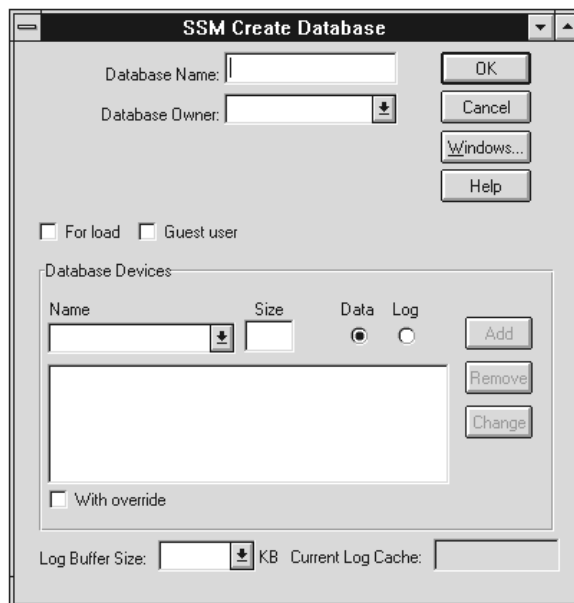


Figure 7-1: Create Database dialog box

3. Enter the following information:
 - In Database Name, enter the name of the database.
 - In Database Owner, enter the SQL Server login of the owner of the database. If you do not enter a login name, as creator of the database you become its owner.

4. If you are creating the database so you can restore it from a backup, check the For Load check box. This is the case only if you are recovering from media failure or if you are moving a database from one location to another.
5. To create a guest user in the database, check the Guest User check box.
6. In the Database Devices group box, enter specifications for how to allocate the database on one or more database devices. For each database device allocation, provide the following information:
 - Select the name of a database device from the Name drop-down list box.
 - Enter the size of the allocation on the device in the Size edit box. If you do not enter a size, SQL Server allocates either the value of the **database size** configuration variable or the size of the *model* database, whichever is larger.
 - If you have limited storage **and** must put the transaction log and the data on the same logical device, checking the With Override check box allows SQL Server to maintain the log on separate device fragments from the data.
7. If you store the transaction log on a separate device from the data, and this allocation is for the transaction log, select the Log option button.

Click the Add button to transfer the allocation information into the list of database devices allocated for this database. Repeat this step until you have specified all database device allocations for the database.

To change a database device allocation:


1. Select the device in the list of database devices.
2. Make the changes you want in the data fields above the list box.
3. Click the Change button to update the information in the list.

To remove a database device allocation from the list:

1. Select the device in the list of database devices.
2. Click the Remove button.

► Note







You cannot remove or change a database device after creating the database, unless you first delete the database.

8.  To specify a log buffer size, enter a value in the Log Buffer Size box. The value must be a power of 2 in the range of 2K to 16K. This option is available only if you are connected to a SQL Server at release 11.0 or later.

The log buffer size determines the size of the I/Os that SQL Server can perform to the memory buffer in the cache designated for transaction log I/Os. (The cache used for transaction log I/Os is shown in the Log Cache box.) The default is 2K, indicating that SQL Server performs transaction log I/Os one data page at a time.

9. Click Create. SQL Server Manager creates the database and adds a new database icon to the Voyager window.

Shortcuts

-
- | | |
|---|---|
|   | Select the database container icon in the appropriate SQL Server; then, select the Create toolbar button. |
|---|---|
-
- | | |
|---|--|
|   | Click the right mouse button over the database container icon in the appropriate SQL Server; then, choose Create from the shortcut menu. |
|---|--|
-
- | | |
|---|---|
|   | Click the right mouse button over the SQL Server icon. Choose Create from the shortcut menu; then, choose Database. |
|---|---|
-

Deleting a Database

This section describes how to delete a database, as well as how to delete a database if SQL Server detects that the database is damaged.

Deleting a Database

Deleting a database deletes the database and all its objects.

Privileges	Only the owner of a database can delete it.
------------	---

To delete a database:

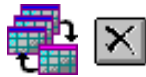
1. Select the icon of the database you want to delete.
2. Choose Delete... from the Database menu.
3. Confirm the deletion by clicking Yes in the confirmation dialog box.

► **Note**

It is a good practice to back up the *master* database after deleting a user database.

Shortcuts

Select the database icon; then, select the Delete toolbar button.



Click the right mouse button over the database icon. Choose Delete from the shortcut menu.



Deleting a Damaged Database

If SQL Server detects that a database is damaged and cannot be recovered or reused, SQL Server Manager displays an additional confirmation dialog box after you choose Yes in the initial delete confirmation.

Privileges	Only the owner of a damaged database can delete it.
------------	---

This second confirmation dialog box identifies the database as damaged and requests a final confirmation. Confirm the deletion by clicking Yes.

Displaying and Modifying Database Information

After creating a database, you can examine or modify it with the Database Properties dialog box. This section describes how to examine database details and how to make the following types of changes to a database that you can make in the Database Properties dialog box:

- Extending storage allocations
- Modifying log cache properties
- Transferring database ownership

Examining Database Details

To open the Database Properties dialog box:

1. Select the icon of the database to display.
2. From the Database menu, choose Properties...

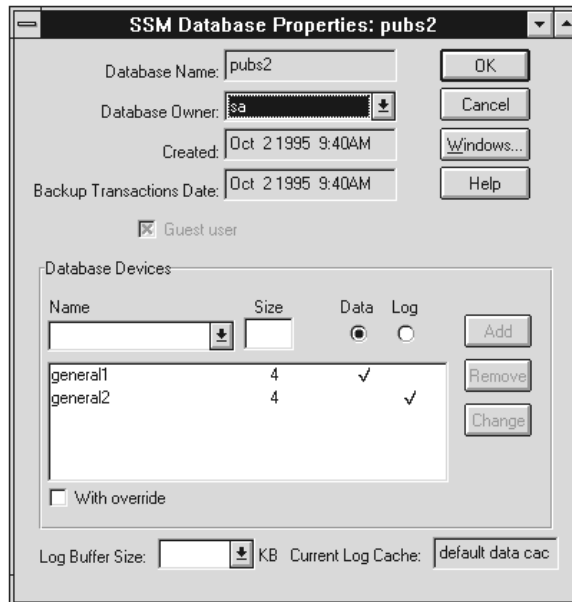


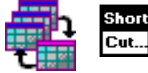


Figure 7-2: Database Properties dialog box

Shortcuts

	Double-click the database icon.
	Select the database icon; then, select the Properties toolbar button.
	Click the right mouse button over the database icon; then, choose Properties... from the shortcut menu.

Modifying Database Storage Allocations

As a database grows, you may need to increase the storage allocated to it. After allocating storage space, you cannot de-allocate it; you can only add to it.

Privileges	Only a System Administrator can change storage allocations.
------------	---

You can allocate additional space on the device where you made the initial allocation or on a different device.

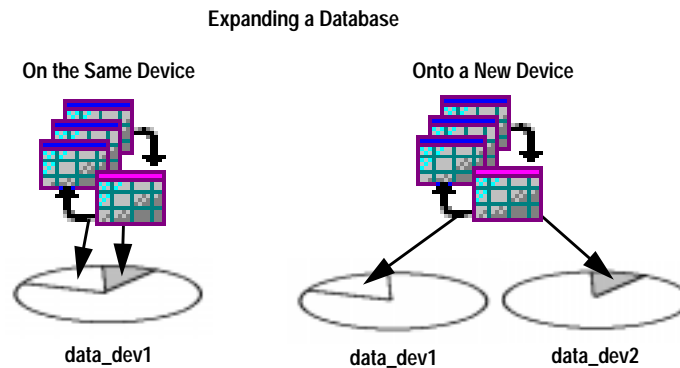


Figure 7-3: Expanding a database

To expand a database, open the Database Properties dialog box as described in “Examining Database Details” on page 7-8. Then:

- To allocate additional space on an existing database device, follow the steps in “Allocating Space on the Same Device” on page 7-10.
- To allocate space on a different database device, follow the steps in “Allocating Space on a New Device” on page 7-10.

► **Note**

To reduce the size of a database, you must back up all the data, delete the database, recreate it with smaller storage allocations, and then load the data back in.

Allocating Space on the Same Device

To allocate space on the same device:

1. Enter the following information in the boxes above the list in the Database Devices group box:
 - Select the name of the current database device from the Name drop-down list.
 - Enter the size of the additional allocation on the device in the Size edit box, in megabytes.
2. Click the Add button to move the data from the entry fields into the list box.
3. Click OK. SQL Server increases the allocation on the selected database device to the sum of the original allocation and the additional allocation you specified.

Allocating Space on a New Device

To allocate space on a new device:

1. Enter the following information in the boxes above the list in the Database Devices group box:
 - Select the name of a new database device from the Name drop-down list.
 - Enter the size of the allocation on the new device in the Size edit box, in megabytes.

2. Click the Add button to move the data from the entry fields into the list box.
3. Click OK.

When the original device is full, any additional database expansion takes place on the new device you specified.

Examining Database Storage Allocations

To see the current storage allocation of a database, its transaction log, and any indexes it contains, display the Database Allocations dialog box:

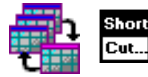
1. Select the icon of the database you want to display.
2. From the Database menu, choose Allocations... The Database Allocations dialog box opens.

This dialog box shows how the total amount of space allocated to the database is distributed, in megabytes. For database data, indexes, and transaction log, the dialog box shows how much space is currently used, how much is reserved but unused, and how much is free.

Data	Index	Log	Total (MB)	
0.168	0.102	0.326	0.596	Used
0.357	0.555	0.000	0.912	Reserved, but unused
			0.492	Free
			2.000	Allocated

Figure 7-4: Database Allocations dialog box

Shortcut



Click the right mouse button over the database icon; then, choose Allocations... from the shortcut menu.

Transferring Database Ownership

After creating a database, a System Administrator can transfer ownership to a different user. This allows the control of database creation and resource allocation to remain centralized in the hands of the System Administrator, while permitting database administration to rest with the principal database users.

When you change ownership of a database, the new owner must have a SQL Server login but must not be a user of the database or have an alias in it.

Privileges	Only a System Administrator can transfer database ownership.
------------	--

To change database ownership:

1. Open the Database Properties dialog box as described in “Displaying and Modifying Database Information” on page 7-7.
2. Select the SQL Server login of the new owner from the Database Owner list.
3. To transfer the permissions and aliases of the current owner to the new owner, select the Transfer Alias and Permission box. This box is hidden until you enter a login name in the Database Owner box.
4. Click OK. Database ownership and the privileges associated with it transfer to the new owner.



Changing Log Buffer Size

The log buffer size determines the size of the I/Os that SQL Server can perform to the memory buffer in the cache designated for transaction log I/Os. (The cache used for transaction log I/Os is shown in the Log Cache box of the Database Properties dialog box.) The default buffer size is 2K, indicating that SQL Server performs transaction log I/Os one data page at a time. When you are in a database in a release 11.0 SQL Server, you can change the log buffer size for the database's transaction log.

To change the log buffer size:

1. Open the Database Properties dialog box as described in "Displaying and Modifying Database Information" on page 7-7.
2. Enter a value in the Log Buffer Size box. The value must be a power of 2 in the range of 2K to 16K.
3. Click OK.

Generating and Executing Database DDL Scripts

Data Definition Language (DDL) consists of Transact-SQL™ commands combined into scripts that capture the schema definition of a SQL Server or database and its objects. You can run these scripts to recreate a database in its original location or on a different SQL Server.

Privileges	Only a System Administrator can generate and execute database DDL scripts.
------------	--

SQL Server Manager allows you to generate DDL scripts automatically and to execute them. Use any text editor to customize DDL scripts to reflect the environmental variations of a different SQL Server than the one on which you generated them.

When you generate or execute DDL for a database, SQL Server Manager creates or executes the scripts for the database and its objects.

Generating Database DDL

To generate database DDL scripts:

1. Select the icon of the database for which to generate scripts.
2. From the Database menu, choose Generate DDL... The Database Generate DDL dialog box opens.

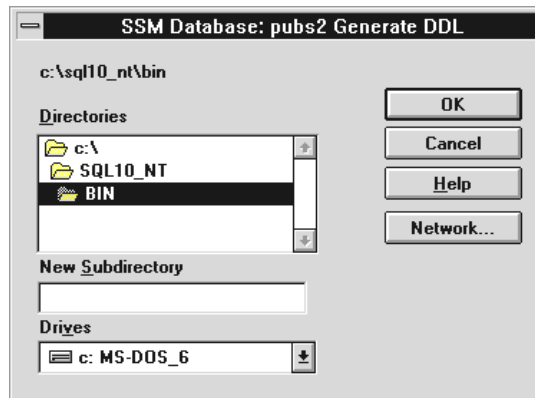
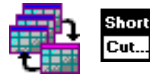


Figure 7-5: Database Generate DDL dialog box

3. Enter the following information:
 - For Directories, select the location in the directory hierarchy where you want SQL Server Manager to create the top-level DDL script directory.
 - For New Subdirectory, type the name of the top-level directory in which you want SQL Server Manager to create database DDL scripts. The default directory is the *ddl* subdirectory of the root directory of your Sybase installation.
 - For Drives, select the letter of the disk drive on which to store DDL scripts. To access an unconnected drive, click the Network... button to display the Connect Network Drive dialog box.
4. Click the OK button. SQL Server Manager creates the appropriate database DDL script hierarchy in the directory location you specified. For a list of database DDL script files, see “Examining and Editing DDL Script Files” on page 7-16.

Shortcut



Click the right mouse button over the database icon. Choose Generate DDL... from the shortcut menu.

Executing Database DDL

To execute database DDL:

1. Select the icon of the database you want.
2. From the Database menu, choose Run DDL... The Database Run DDL dialog box opens.

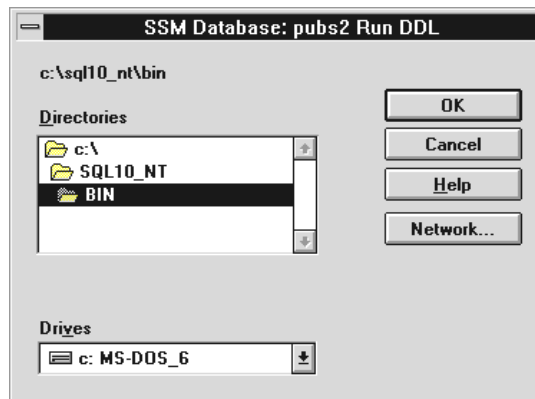


Figure 7-6: Database Run DDL dialog box

3. Enter the following information:
 - From the Drives list, select the letter of the disk drive where the DDL scripts are stored. To access an unconnected drive, click Network... and enter information in the Connect Network Drive dialog box.
 - In the Directories list, select the directory named for the SQL Server where you generated the database DDL scripts. This directory is one level below the directory you specified as New Directory in the Database Generate DDL dialog when generating the scripts.

4. Click OK. SQL Server Manager executes the appropriate SQL Server DDL scripts. For a list of SQL Server DDL script files, see “Examining and Editing DDL Script Files” on page 7-16.

Shortcut



Click the right mouse button over the database icon. Choose Run DDL... from the shortcut menu.

Examining and Editing DDL Script Files

When SQL Server Manager generates DDL scripts for a database, it places the scripts in the directory you specify in the Database DDL Generate dialog box. Within this directory, it places scripts in a subdirectory named *srv*, where *srv* is the name of the SQL Server in which the database resides.

When SQL Server Manager generates database-level scripts, it creates a set of files for each database and places them in a subdirectory named for the database. The files in the *srv* subdirectory and any database subdirectories below it carry a *.ddl* extension.

The following figure illustrates the directory hierarchy and shows the files created in each directory for a top-level DDL directory called *dbddl*, a SQL Server called *SSM_1002*, and a database called *epics*.

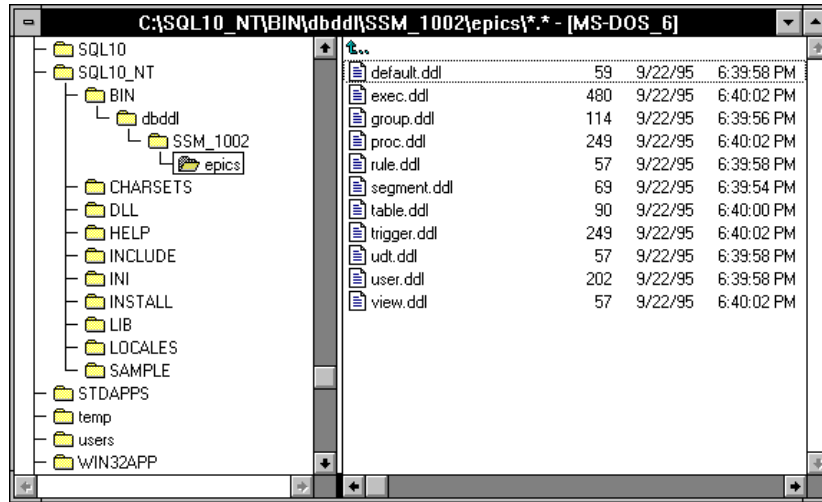


Figure 7-7: Database DDL directory and file structure

The generated DDL scripts are plain text files. To view or edit them, use any text editor—for example, Windows or NT Notepad.

Review the DDL files for applicability to the environment in which you plan to execute them. The following table lists the files in each *db* subdirectory of the *srv* subdirectory, where *db* is the name of a database for which DDL scripts were generated.

Table 7-1: Files created during generation of database-level DDL

File Name	What the File Contains
<i>exec.ddl</i>	A list of all other DDL files for <i>db</i> in the order they should be executed. SQL Server Manager uses this file at DDL execution time to determine which files to execute and in what order.
<i>segment.ddl</i>	DDL to create all segments and thresholds defined for <i>db</i> . The segments are created on the same logical devices as on <i>srv</i> .
<i>udt.ddl</i>	DDL to create all user-defined datatypes in <i>db</i> and to bind defaults and rules to datatypes. Ownership of the datatypes matches ownership in <i>db</i> on <i>srv</i> .

Table 7-1: Files created during generation of database-level DDL (continued)

File Name	What the File Contains
<i>user.ddl</i>	DDL to add users to <i>db</i> .
<i>group.ddl</i>	DDL to create all groups in <i>db</i> .
<i>table.ddl</i>	DDL to create all indexes and tables in <i>db</i> , set permissions on tables, and bind defaults and rules to table columns. Tables are created on the same segments on which they existed on <i>srv</i> . Table ownership matches ownership on <i>srv</i> .
<i>view.ddl</i>	DDL to create all views in <i>db</i> .
<i>default.ddl</i>	DDL to create all defaults defined in <i>db</i> on <i>srv</i> .
<i>rule.ddl</i>	DDL to create all rules defined in <i>db</i> on <i>srv</i> .
<i>proc.ddl</i>	DDL to create all user-defined stored procedures in <i>db</i> and to set permissions on them.
<i>trigger.ddl</i>	DDL to create all triggers defined on tables in <i>db</i> .

Changing Database Options

Database options enable you to customize the behavior of user databases (you cannot change the option settings on the *master* database).

Privileges	A database owner can change option settings.
------------	--

A new user database takes its default option settings from the *model* database. To control the default settings of all new user databases, update the *model* database option settings. Only the System Administrator can change the *model* database.

This section describes how to change database options and gives a brief description of each option. For more detailed option information, see the *System Administration Guide*.

Instructions

To change database option settings for a user database:

1. Select the icon of the database.
2. From the Database menu, choose Options... The Database Options dialog box opens.

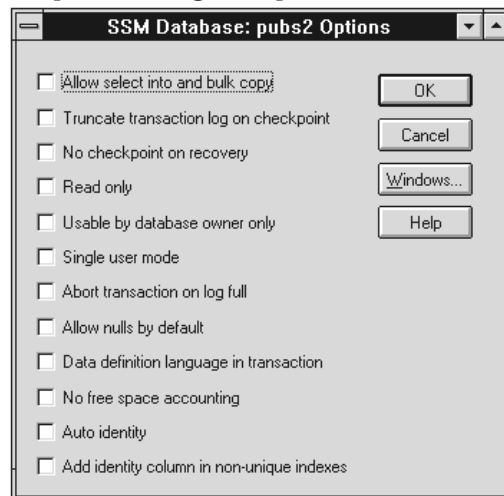


Figure 7-8: Database Options dialog box

3. Select or clear each option you want to change by clicking its checkbox. (Selecting an option sets its status to “ON” in SQL Server; clearing it sets the status to “OFF.”)
4. Click OK. SQL Server resets the options for the database.

Shortcut



Click the right mouse button over the database icon. Choose Options... from the shortcut menu.

List of Options

A description of each option follows:

- **Allow Select Into and Bulk Copy**—allows users to perform non-logged operations. Non-logged operations include `select into` for permanent tables, the bulk copy utility `bcp`, and the `writetext` utility.

You do not need to select this option to allow `select into` for temporary tables or to run `bcp` on a table with indexes because inserts are logged.

Also, attempting to dump the transaction log in a database after unlogged changes have been made to the database with `select only` or bulk copy produces an error message instructing you to use `dump database` instead.

- **Truncate Transaction Log on Checkpoint**—truncates the transaction log (removes committed transactions) every time the database is checkpointed.

If you select this option, you cannot dump the transaction log. It may be useful to turn this option on during development work when backups of the transaction log are typically not needed.

► **Note**


If you select **Truncate Transaction Log on Checkpoint** for development purposes, clear it periodically and dump the transaction log. If you never dump the transaction log, it continues to grow, and eventually you run out of space in the database.

- **No Checkpoint on Recovery**—used only under special circumstances involving the maintenance of multiple copies of a database. See the *System Administration Guide* for details.
- **Read Only**—prevents modification of any data in the database.
- **Usable by Database Owner Only**—restricts database access to the database owner.
- **Single User Mode**—allows only one user at a time to use the database.
- **Abort Transaction on Log Full**—decides how SQL Server treats active transactions when the database's log becomes critically low on space:
 - To cancel all user queries that need to write to the transaction log until space in the log has been freed, select this option.

- To suspend transactions and awaken them when space has been freed, clear this option.
- Allow Nulls by Default—affects the ability of columns in newly created database tables to accept NULL values:
 - If you select this option, columns in newly-created tables allow null values unless the column definitions explicitly state “not null.”
 - If this option is cleared, nulls are not allowed unless the column definitions explicitly permit them.
- Data Definition Language in Transaction—allows users to include DDL syntax within their transactions.

Generally, avoid using Data Definition Language commands inside transactions. For more information about this option, see the *SQL Server Reference Manual*.
- No Free Space Accounting—allows users to suppress free space accounting and execution of threshold actions for the non-log segments. Selecting this option speeds recovery time because the free-space counts are not recomputed for those segments.

Privileges	A System Security Officer can change the “No Free Space Accounting” option.
------------	---

- Auto Identity—automatically adds a 10-digit IDENTITY column in a new table when a user creates the table without specifying a primary key, a unique index, or an IDENTITY column.
-  Add Identity Column in Non-Unique Indexes—automatically includes an IDENTITY column in a table’s index keys, so that all indexes created on the table are unique. This option is available when you are connected to a SQL Server at release 11.0 or later.

Issuing a Database Checkpoint

A **checkpoint** is an automatic mechanism to guarantee that data pages changed by completed transactions are regularly written from the cache in memory to the database device. Each time it issues a checkpoint, SQL Server does the following:

- Freezes all current data modification transactions
- Writes pages that have been modified in memory, but not on disk, since the last checkpoint, to the database device
- Writes a checkpoint to the transaction log
- Unfreezes the current transactions

Through this mechanism, SQL Server regularly synchronizes the database and its transaction log, thereby shortening the recovery time in the event of a system failure.

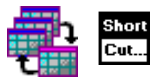
SQL Server uses the *recovery interval* configuration variable to determine when to issue a checkpoint automatically. A System Administrator or database owner can also issue a checkpoint manually.

Privileges	A database owner can issue a manual checkpoint.
------------	---

To issue a manual checkpoint:

1. Select the icon of the database you want.
2. From the Database menu, choose Checkpoint.

Shortcut



Click the right mouse button over the database icon. Choose Checkpoint from the shortcut menu.

Using Database Segments

Segments are named subsets of the database devices on which a particular database is stored. A segment is a label that points to one or more database devices. Within a particular database, you can define segments for the database devices already allocated to that database.

Each database can contain up to 32 segments. Every database contains segments called *system*, *logsegment*, and *default*. You can define additional segments, as needed.

Defining segments for a database lets you control placement of the objects that consume the most storage—tables and indexes. This gives you several performance and control advantages:

- By placing large tables on segments that span multiple physical devices, you can increase I/O throughput.
- By separating tables and their nonclustered indexes on different physical devices, you can also increase I/O throughput.
- By placing a table on a segment of a specific size, you can control space usage, since a table cannot grow larger than its segment allocation.

To monitor the use of space on a segment so that you can take action before a segment becomes full, you can define **threshold** values. Thresholds allow you to automatically trigger actions such as notification or dumping a transaction log when a segment is filled to the level of the threshold.

Creating and Deleting Segments

This section describes how to create and delete a segment.

Creating a Segment

Before you create a segment, the database device on which you create it must exist, and the database you create it for must have space allocated on the device.

Privileges	A database owner can create a segment.
------------	--

To create a segment:

1. Select the icon of the database on which to create the segment.
2. From the Database menu, choose Create; then, choose Segment from the cascading menu. The Create Segment dialog box opens.

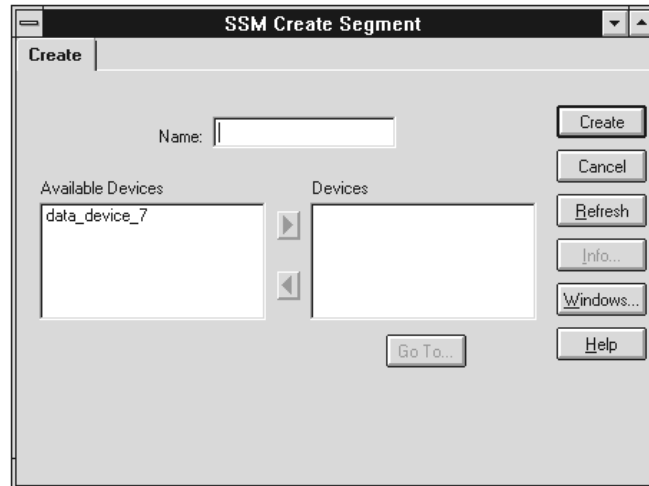
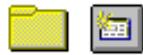


Figure 7-9: Create Segment dialog box

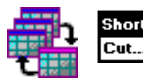
3. Enter the segment name in the Name edit box.
4. Use the arrow buttons to move the names of the devices to which the segment will point from the Available Devices list to the Devices list:
 - Select a device name in the Available Devices list.
 - Click the right-pointing arrow button to copy the name into the Devices list.
 - To remove a device from the Devices list, select it and then click the left-pointing arrow.
5. When you are satisfied with the Devices list, click Create. SQL Server creates the database segment on the specified database devices, and a new segment icon appears in the right pane of the Voyager window.

Shortcuts

Click the Segments container icon in the appropriate database; then, select the Create toolbar button.



Click the right mouse button over the Segments container icon in the appropriate database; then, choose Create from the shortcut menu.



Click the right mouse button over the appropriate database icon. Choose Create; then, choose Segment.

Deleting a Segment

To delete a segment:

1. Select the icon of the segment to delete.
2. From the Segment menu, choose Delete.
3. Confirm the deletion by clicking Yes in the confirmation dialog box.

Shortcuts

Select the icon of the segment to delete; then, select the Delete toolbar icon.



Click the right mouse button over the segment to delete; then, choose Delete from the shortcut menu.

Displaying and Modifying Segment Information

SQL Server Manager enables you to examine the mapping of database segments to database devices and the distribution of database objects on those segments. You can also extend an existing segment onto additional database devices. All of these activities are available through the Segment dialog box.

Examining Segment Details

Use the Properties tab of the Segment dialog box to examine the mapping of a database segment to database devices. This dialog box also enables you to navigate to a specific database device.

1. Select the icon of the segment you want to examine.
2. From the Segment menu, choose Properties...

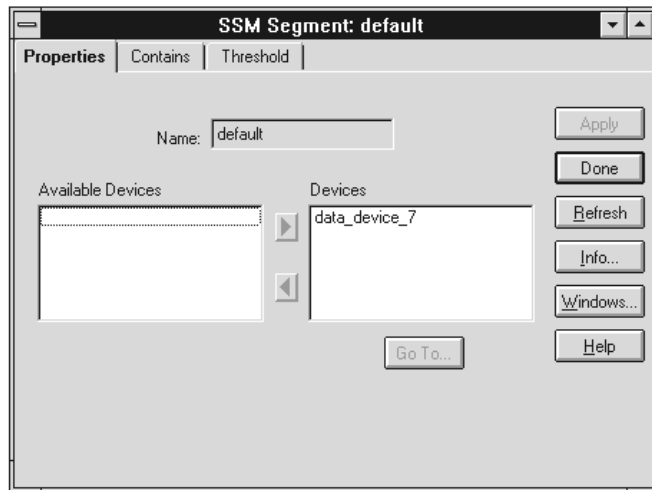


Figure 7-10: Properties tab of Segment dialog box

Shortcuts

Double-click the segment icon.



Select the segment icon; then, select the Properties toolbar button.



Click the right mouse button over the segment icon; then, choose Properties... from the shortcut menu.



Navigating to a Database Device

To display the Database Device dialog box for any database device mapped to the segment:

1. Select the device by clicking its row in the Devices list.
2. Click the Go To... button. The dialog box for the selected database device opens.

Shortcut

Double-click the device row in the Devices list.



Extending a Segment

To extend a segment onto another database device on which the database has been allocated:

1. Open the Segment dialog box as described in “Examining Segment Details” on page 7-26.

Privileges	A database owner can extend a segment.
------------	--

2. Use the arrow buttons to move the name of the new device from the Available Devices list to the Devices list:

- Select a device name in the Available Devices list.
 - Click the right-pointing arrow button to copy the name into the Devices list.
 - To remove a device from the Devices list, select it and click the left-pointing arrow.
3. Click Apply to update the segment.

Examining Objects Stored on a Segment

When you add an index or table to the database, you can specify the segment on which to place the object. From the Segment dialog box, you can examine and navigate to the indexes and tables that use the segment to map to a particular database device. To examine objects on a segment:

1. Open the Segment dialog box as described in “Examining Segment Details” on page 7-26.
2. Click the Contains button to display the Contains tab.

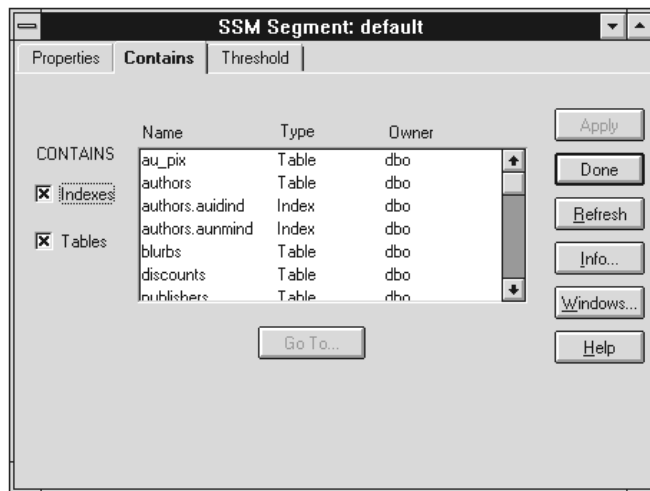


Figure 7-11: Contains tab of the Segment dialog box

This tab lists the indexes and tables whose current growth is on the segment, along with the names of their owners. Use the Indexes and Tables checkboxes under the Contains label to specify whether the list box displays indexes, tables, or both.

Navigating to an Index or Table on a Segment

To display the Index or Table Dialog box for any index or table stored on the segment, indicate whether to display indexes or tables in the list box by checking the Indexes or Tables checkbox. Then:

1. Select the object by clicking its row in the list box.
2. Click the Go To... button. The dialog box for the selected index or table opens.

Shortcut

Double-click the row of the index or table to display.



Using Segment Thresholds

Thresholds provide a way to monitor and preserve free space in a database. A threshold is a value specified as a number of pages of free space. Each threshold value is associated with a stored procedure. (For example, you can associate a threshold with a procedure that dumps the transaction log.) When free space on a segment falls below the threshold value, the stored procedure automatically executes.

You can define thresholds on segments that hold either data or transaction logs. A database can have as many as 256 thresholds. Using SQL Server Manager, you can create, delete, or modify a threshold, and you can create the stored procedure that executes when a threshold is reached.

Privileges	A System Administrator or database owner can create, delete, or modify a threshold.
------------	---

Threshold Concepts: Last-Chance Threshold

If a transaction log is on its own segment (that is, if a segment is mapped to a device on which a database has allocated log-only space), SQL Server automatically creates a **last-chance threshold** on the segment, along with any other thresholds that you add. This threshold is an estimate of the number of free log pages that would

be required to back up the transaction log. SQL Server automatically adjusts the last-chance threshold as you allocate more space to the log segment.

Threshold Concepts: Hysteresis Value

To keep threshold procedures from firing continuously in response to minor fluctuations in space usage, SQL Server uses the **hysteresis** value. This server-wide value, stored in the system variable `@@thresh_hysteresis`, is specified in data pages. Before a threshold that has been activated and has triggered a stored procedure can be activated again, the space usage on the segment must decrease by at least the hysteresis value. Similarly, any two thresholds on a segment must be at least two hysteresis values apart.

For example, if one threshold is set at .2MB and the hysteresis value for the SQL Server is 64 pages (.128MB), the closest you could place the next threshold would be $.2 + (2 * .128)$, or .456MB.

Creating a Threshold

To create a threshold:

1. Select the icon of the segment on which to create the threshold.
2. Open the Segment dialog box as described in “Examining Segment Details” on page 7-26.

3. Click the Thresholds control to display the Thresholds tab:

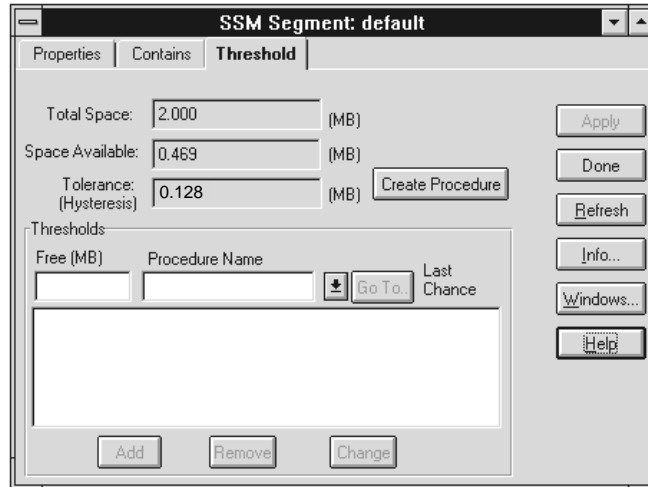


Figure 7-12: Thresholds tab of Segment dialog

4. In the Free (MB) box, enter the amount of free space for the threshold to guard, in megabytes. The stored procedure you associate with the threshold executes when free space on the segment falls below this amount.

The Total Space (MB), Space Available (MB), and Tolerance (Hysteresis) boxes provide current reference information to help you decide on an appropriate free space value. All of these boxes display information in megabytes.

5. In the Procedure Name list, select the name of the stored procedure to associate with the threshold:

- To review the definition of a selected procedure, click the Go To... button. The Properties tab of the selected procedure opens. When ready, click Done to close the Properties tab.
- To create a new stored procedure, click the Create Procedure... button above the Thresholds group box. The Create Procedure dialog box opens. After completing the definition and clicking the Create button, return to the Thresholds tab. The new stored procedure is available for selection in the Procedure Name list.

► **Note**

For instructions on creating a stored procedure, see “Creating a Procedure” on page 9-29.

6. Click Add. The threshold definition moves into the threshold list. To remove a threshold from the list, select the definition in the list and click Remove. To change the definition of a threshold in the list, select it. Update the values in the Free (MB) or Procedure Name boxes, and click Change to update the values in the list.
7. Click Apply.

Deleting a Threshold

To delete a threshold:

1. Display the segment’s Thresholds tab as described in “Creating a Threshold” on page 7-30.
2. Select the definition of the threshold to delete in the Thresholds list.

► **Note**

You cannot delete a last-chance threshold (marked with a check in the Last Chance column).

3. Click Remove. The definition is deleted from the Thresholds list.
4. Click Apply.

Modifying a Threshold

In user-defined thresholds, you can change the free space value and the stored procedure with which the threshold is associated. In a last-chance threshold, you can change only the stored procedure; SQL Server automatically maintains the free space value. To modify a threshold:

1. Display the segment’s Thresholds tab as described in “Creating a Threshold” on page 7-30.
2. Select the definition of the threshold to modify in the Thresholds list. The definition is copied into the Free (MB) and Procedure Name boxes.

3. Modify the values in the Free (MB) and Procedure Name boxes as desired.
4. Click Change to update the values in the Thresholds list.
5. Click Apply.

Preparing for Backup and Recovery

Backup and recovery are two of the most critical activities a system or database administrator performs. While SQL Server has automatic recovery procedures to protect you during power outages and computer failures, your best protection against media failure is regular and frequent database backups.

Before performing backup and recovery operations, you should address the following issues:

- Develop a backup and recovery plan.
- Make sure you can connect to the Backup Server from each SQL Server you administer.
- Decide on the backup media you will use, and create dump devices that identify your physical backup media to SQL Server.

Developing a Backup and Recovery Plan

To develop a reliable backup and recovery plan, you must understand how SQL Server handles backups and recovery. To get started, read the chapter of the *System Administration Guide* on developing a backup and recovery plan. The *System Administration Guide* also has a detailed discussion of the commands and options used to perform backup and recovery operations.

The following are good practices to include in your backup and recovery procedures:

- Make frequent backups of the *master* database.
- Truncate the log of the *master* database frequently.
- Keep a current backup of the *model* database.
- Make frequent database and transaction log dumps for all databases.
- Keep statistics on how long it takes to back up and restore databases and how much space is required.

Identifying the Backup Server

SQL Server backups are performed by Backup Server, an Open Server™-based program that runs on the same host as SQL Server. Before backing up a database, make sure that the Backup Server for your SQL Server is up and running and that it is correctly identified in the interfaces file. Also, the login of the person who starts Backup Server (usually “sybase”) must have write permission for the physical dump device on which the backup is created.

Managing Dump Devices

Before backing up a database, you must have a **dump device** available to receive the backup. A dump device is a tape or disk device for storing a backup copy of a database or its transaction log. This section describes how to create, delete, and examine the properties of a dump device.

Creating a Dump Device:

Privileges	Only a System Administrator can create a dump device.
------------	---

To create a dump device

1. Click the icon for the SQL Server on which to create a dump device.
2. From the Server menu, choose Create; then, choose Dump Device. The Create Dump Device dialog box opens.

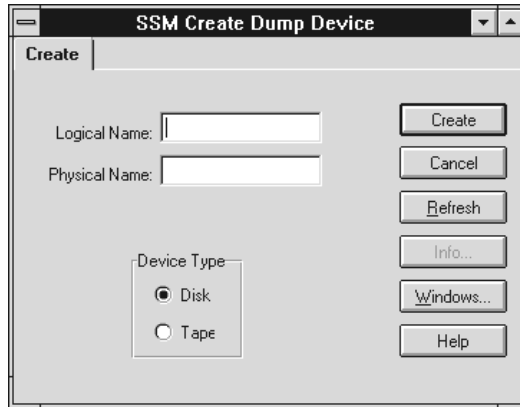
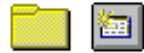


Figure 7-13: Create Dump Device dialog box

3. Enter the following information:
 - For Logical Name, enter your name for the device.
 - For Physical Name, enter the name of the physical device in your environment to map the logical name to. For a **tape dump device**, it must be the name of a tape drive. For a **disk dump device**, it must be the name of an operating system file, and you must specify the full path name.
4. Specify whether the dump device is a disk or tape device by selecting the Disk or Tape option button.
5. If the device is a tape device, enter its capacity in the Size (MB) box, in megabytes. This box is visible only if you select the Tape option button.
6. Click Create. SQL Server creates the dump device, and a new dump device icon appears in the Voyager window.

Shortcuts



Click the Dump Device container icon in the appropriate SQL Server; then, select the Create toolbar button.



Click the right mouse button over the Dump Device container icon in the appropriate SQL Server; then, choose Create from the shortcut menu.



Click the right mouse button over the appropriate SQL Server icon. Choose Create; then, choose Dump Device.

Deleting a Dump Device

To delete a dump device:

Privileges	Only a System Administrator can delete a dump device.
------------	---

1. Select the icon of the device to delete.
2. Choose Delete from the Dump Device menu.
3. Confirm the deletion by clicking Yes in the confirmation dialog box.

Shortcuts



Select the icon of the dump device to delete; then, select the Delete toolbar icon.



Click the right mouse button over the dump device to delete; then, choose Delete from the shortcut menu.

Examining Dump Device Properties

To display a Dump Device dialog box:

1. Select the icon of the dump device.
2. Choose Properties... from the Dump Device menu. The Dump Device dialog box opens.

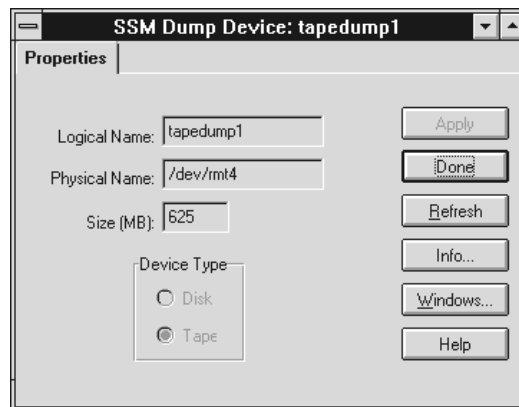


Figure 7-14: Properties tab of the Dump Device dialog box

Shortcuts

Double-click the dump device icon.



Select the dump device icon; then, select the Properties toolbar button.



Click the right mouse button over the dump device icon; then, choose Properties... from the shortcut menu.



Backing up a Database

SQL Server Manager enables you to back up a database and its transaction log. This section describes how to use the Database Backup dialog box to specify backup options and execute a database backup.

Scheduling Database Backups

The frequency of your database backups determines how little (or how much) work you will lose in the event of a media failure. This section presents some guidelines about when to schedule backups of user and system databases.

Guidelines for Backing up User Databases

Back up a user database after the following events:

- After creating it, to develop a baseline backup. Thereafter, back up each database on a fixed schedule. Daily backups of the transaction log and weekly database backups are the minimum recommended. Many installations with large and active databases make transaction log dumps every half hour or hour and database dumps every day.
- After creating a new index. The transaction log records the create index transaction but does not log the filling of index pages with information.
- After executing the following unlogged transactions:
 - Unlogged writetext
 - select into on a permanent table
 - Bulk copy in “fast” mode (bcp into a table with no triggers or indexes)
- After truncating the transaction log without making a backup copy.

Guidelines for Backing up the *master* Database

Back up the *master* database each time you change it. Each operation that affects disks, storage, databases, or segments makes changes to the system tables in the *master* database. For example, back up *master* after you:

- Create, modify, or delete a database, database device, database segment, dump device, or procedure
- Add or change disk mirroring specifications
- Change SQL Server configuration variables

Using the Database Backup Dialog Box

Privileges	Operators can back up any database. Database owners can back up databases they own.
------------	---

To back up a database:

1. Select the icon of the database to back up.
2. From the Database menu, choose Backup... The Database Backup dialog box opens.

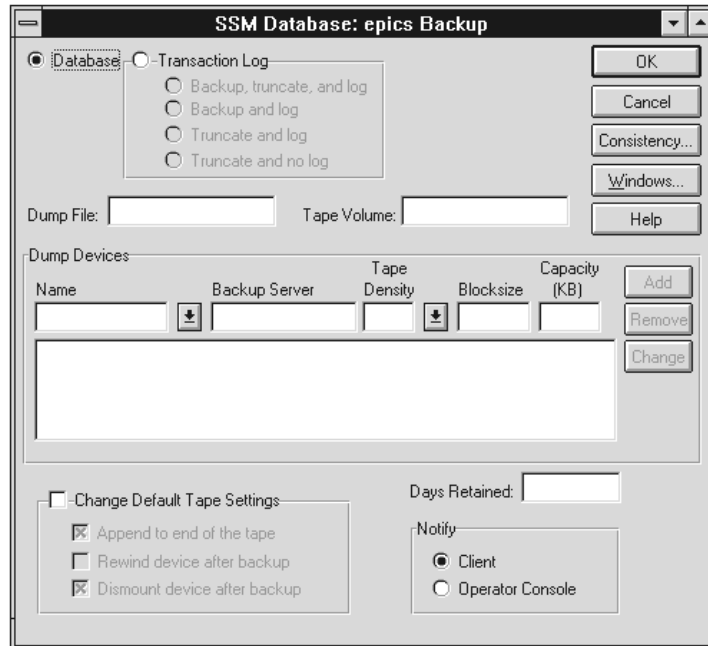


Figure 7-15: Database Backup dialog box

3. Specify whether to back up a database's data or its transaction log by selecting the Database or Transaction Log option in the upper left corner.
4. If you are backing up the transaction log, choose one of the following options in the Transaction Log group box:
 - Backup, Truncate, and Log—back up the transaction log, remove the inactive portion of the log, and create a new transaction log entry recording the backup.
 - Backup and Log—back up the transaction log and create a new transaction log entry recording the backup. This option retains the transaction log's entries.
 - Truncate and Log—remove the inactive portion of the transaction log without backing it up to a dump device, and create a new transaction log entry recording the dump.
 - Truncate and No Log—remove the inactive portion of the transaction log without backing it up to a dump device and

without creating a new transaction log entry recording the dump. This option is useful if the transaction log grows so large that there is no room to dump it normally.

5. Optionally, enter the name of the dump file in the Dump File edit box. The name cannot exceed 17 characters and must conform to operating system conventions for file names. If you do not enter a file name, Backup Server generates a default name.
6. If the dump device is a tape, enter the volume name in the Tape Volume edit box.
7. If the backup option is Database or Backup, Truncate, and Log or Backup and Log, in the Dump Devices group box, specify the dump device or devices to use for the backup. For each dump device, provide the following information:
 - For Name, select the name of the physical or logical dump device from the drop-down list. Enter the absolute pathname for a physical device.
 - When backing up to a device on a remote system, enter the name of the Backup Server running on the remote system.
 - Optionally, if the dump device is a tape device, select an override to the default tape density from the Tape Density drop-down list.
 - Optionally, if the dump device is a disk device, enter an override to the default blocksize in the Blocksize edit box, in bytes. Using the default blocksize is recommended.
 - For Capacity (KB), enter the maximum amount of data that the device can write to a single tape volume, in kilobytes.

To add a set of dump device specifications to the Database Devices list, click Add. To change an entry, highlight it, make changes, and click Change. To delete an entry, highlight it and click Remove.

8. Optionally, enter the number of days for Backup Server to prevent you from overwriting a dump in the Days Retained edit box. To indicate that a dump can be overwritten immediately, enter 0. If you do not enter a number, Backup Server uses the value of the `tape retention` SQL Server configuration variable.
9. Optionally, if the dump device is a tape, select Change Default Tape Settings to indicate that you want to change the way SQL Server handles the tape backup by default. If you leave this box unselected, SQL Server adds the backup after the last end-of-

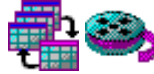
tape mark on the device and dismounts the tape device when the backup is complete.

When you select Change Default Tape Settings, the following options become available. Indicate the options you want by selecting or clearing the corresponding check boxes:

- Append to End of the Tape—Adds the backup after the last end-of-tape mark on the device
 - Rewind Device After Backup—Rewinds the tape when the backup is complete
 - Dismount Device After Backup—Dismounts the tape device when the backup is complete
10. Specify where you want to receive backup messages by selecting one of the following radio buttons in the Notify group box:
- Client—Messages display in a SQL Server Manager dialog box.
 - Operator Console—Messages display on the console of the host where Backup Server is running.
11. Click OK to start the backup.

Shortcuts

Drag the icon of a database onto the icon of a dump device.



Click the right mouse button over the database icon, and choose Backup... from the shortcut menu.



Responding to Volume Change Prompts

When a backup event requires operator intervention, Backup Server sends a message to your workstation or the operator console of the Backup Server host (as specified in the Notify box of the Database Backup dialog box). For example, when a tape volume becomes full, Backup Server sends the following message to signal that a new tape volume must be mounted:

```
Mount the next volume to write.
```

After mounting the tape, the operator must notify Backup Server that the intervention is complete. To do this, use the Volume Change dialog box, which SQL Server Manager displays when a message comes from Backup Server:

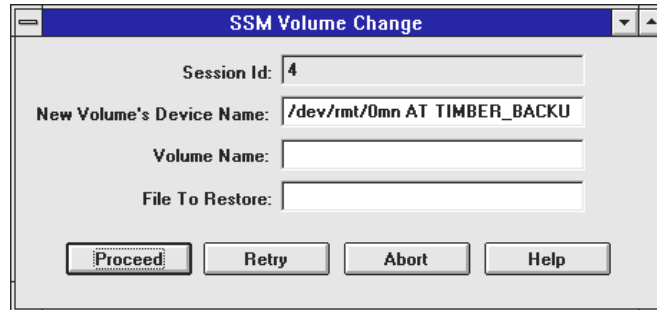


Figure 7-16: Volume Change dialog box

The information to enter in the dialog box, and the command button to select, vary with the contents of the message. The following messages are the most common:

- If Backup Server reaches the end of the current tape volume, it sends the following message:

`Mount the next volume to write.`

- If you specified that the backup should be appended to an existing volume, and Backup Server cannot find the end-of-file mark on that volume, it sends the message:

`Mount the next volume to search.`

To Proceed with the Backup

1. Mount a new tape volume.
2. If the device on which the new tape volume is mounted differs from the device of the original tape volume, enter the new device name in the New Volume's Device Name box.
3. Optionally, enter the name of the new tape volume in the Volume Name box. If you do not enter a name, SQL Server Manager uses the name specified in the Tape Volume box of the Database Backup dialog box. If the Tape Volume box is blank, Backup Server leaves the ANSI tape label blank during the backup and then does not check the ANSI tape label value during a restore.

4. Click Proceed. Backup Server checks the tape and then continues the backup.

To Abort the Backup

Click Abort. Backup Server stops the backup and returns control to SQL Server Manager.

To Get Information on Backup Server Messages

SQL Server Manager passes the information you supply in the Volume Change dialog box to SQL Server, which executes the `sp_volchanged` system procedure. For information about additional Backup Server messages and how to respond to them, see the description of `sp_volchanged` in the *SQL Server Reference Manual*.

Restoring a Database

SQL Server Manager enables you to restore a database and its transaction log from database and transaction log backups. This section describes how to use the Database Restore dialog box to specify recovery options and execute a database recovery.

Deciding How to Restore

When your installation experiences media failure, the strategy for recovery depends on the state of your database and transaction log backups. Generally, you need to:

1. Restore the transaction log to capture the transactions that have occurred since the most recent transaction log backup.
2. Restore the database from the most recent complete database backup.
3. Apply each transaction log backup sequentially from the time of the most recent database backup through the time of media failure.

Recovery Considerations

In planning how to recover, consider the following:

- You can load a database backup into a pre-existing database, or you can create a new database with the `for load` option. For details on creating a database for load, see “Creating a Database” on page 7-3. With this method, you can move a database to a different SQL Server. However, you cannot load a database backup that was created on a different operating system or with an earlier release of SQL Server.
- To prevent users from making changes from the time you begin restoring a database until the time you finish applying the last transaction log backup, select the following database options in the Database Options dialog box:
 - Single-User Mode
 - No Checkpoint On Recovery
 - Read Only
 - Usable by Database Owner Only

When recovery is complete, clear these options to allow users to resume updating the database.

- For additional information about recovery, see the *System Administration Guide* and the *Sybase Troubleshooting Guide*.

Using the Database Restore Dialog Box:

Privileges	Operators can restore any database. Database owners can restore databases they own.
------------	---

To restore a database:

1. Select the icon of the database to restore.
2. From the Database menu, choose Restore... The Database Restore dialog box opens.

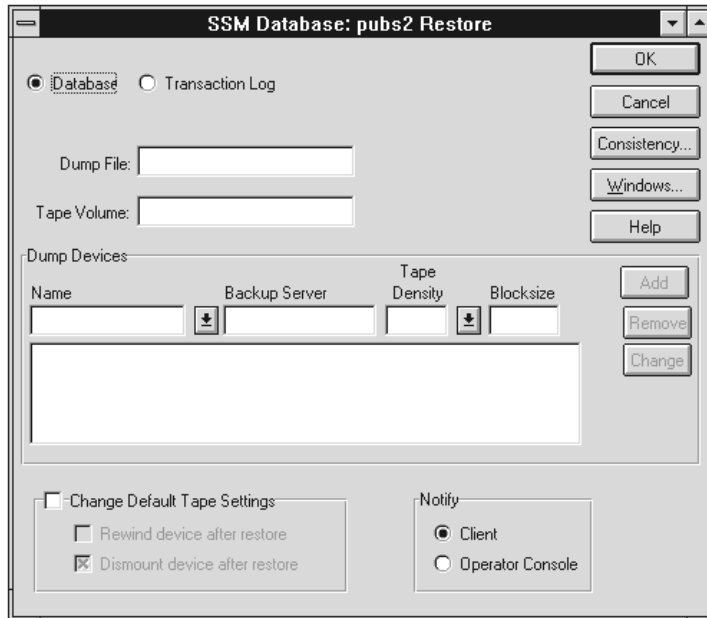


Figure 7-17: Database Restore dialog box

3. Select Database to specify restoring the database, or select Transaction Log to specify restoring its transaction log.
4. In the Dump File edit box, enter the file name of the backup from which you are restoring.
5. If the dump device is a tape, enter the volume name in the Tape Volume edit box.
6. In the Dump Devices group box, specify the dump device or devices to use for the restoration. For each dump device, provide the following information:
 - For Name, select the name of the physical or logical dump device from the drop-down list. Enter the absolute pathname for a physical device.
 - For a remote Server, enter the name of the Backup Server for the SQL Server on which the database or transaction log resides.

- Optionally, if the dump device is a tape device, select an override to the default tape density in the Tape Density drop-down list.
- Optionally, if the dump device is a disk device, enter an override to the default blocksize in the Blocksize edit box, in bytes. Using the default blocksize is recommended.

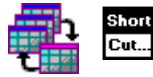
To add a set of dump device specifications to the Database Devices list, click Add. To delete an entry, highlight it and click Remove. To change an entry, highlight it, make changes, and click Change.

7. Optionally, if the dump device is a tape, select Change Default Tape Settings to indicate that you want to change the way SQL Server handles the tape restore by default. If you leave this box unselected, SQL Server dismounts the tape device when the restore is complete and does not rewind the device.

When you select Change Default Tape Settings, the following options become available. Indicate the options you want by selecting or clearing the corresponding check boxes:

- Rewind Device After Restore—Rewinds the tape when the restore is complete
 - Dismount Device After Restore—Dismounts the tape device when the restore is complete
8. Specify where you want to receive restoration messages by selecting one of the following radio buttons in the Notify group box:
 - Client—Messages appear in a SQL Server Manager dialog box.
 - Operator Console—Messages appear on the console of the host where Backup Server is running.
 9. Click OK to start the restore.

Shortcut



Click the right mouse button over the database icon, and choose Restore... from the shortcut menu.

Responding to Volume Change Prompts

When a restore event requires operator intervention, Backup Server sends a message to your workstation or the operator console of the Backup Server host (as specified in the Notify box of the Database Backup dialog box). For example, when Backup Server finishes reading the current tape volume, it sends the following message to signal that a new tape volume must be mounted:

```
Mount the next volume to read.
```

After mounting the tape, the operator must notify Backup Server that the intervention is complete. To do this, use the Volume Change dialog box, which SQL Server Manager displays when a message comes from Backup Server:

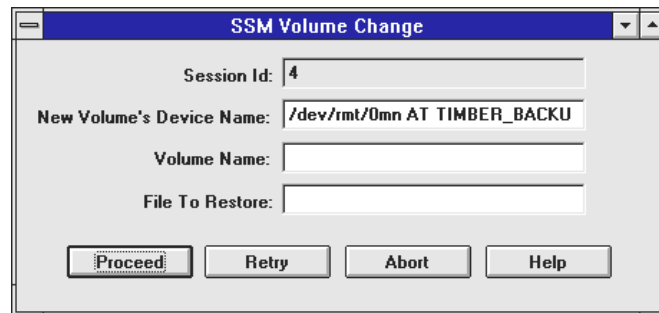


Figure 7-18: Volume Change dialog box

The information to enter in the dialog box and the command button to select vary with the contents of the message. The following messages are the most common:

- When Backup Server is ready to read the next section of the dump file from a multi-volume dump, it sends the following message:

Mount the next volume to read.

- If Backup Server cannot find the specified file on a multi-file medium, it sends the message:

Mount the next volume to search.

To Abort the Restore

Click Abort. Backup Server stops the restore and returns control to SQL Server Manager.

To Proceed with the Restore

1. Mount a new tape volume.
2. If the device on which the new tape volume is mounted differs from the device of the original tape volume, enter the new device name in the New Volume's Device Name box.
3. Optionally, enter the name of the new tape volume in the Volume Name box. Backup Server uses this name to confirm that the correct tape volume has been mounted. If you do not enter a name, Backup Server uses the name specified in the Tape Volume box of the Database Restore dialog box. If the Tape Volume box is blank, Backup Server does not check the ANSI tape label before continuing the restore.
4. Optionally, enter the name of the file to restore in the File to Restore box. If you do not enter a name, Backup Server uses the name specified in the Dump File box of the Database Restore dialog box. If the Dump File box is blank, Backup Server restores the first file on the tape.

► **Note**

If you enter a tape volume name in the Volume Name box, you must also enter a filename in the File to Restore box.

5. Click Proceed. Backup Server checks the tape and then continues the restore.

To Get Information on Backup Server Messages

SQL Server Manager passes the information you supply in the Volume Change Notification dialog box to SQL Server, which executes the `sp_volchanged` system procedure. For information about additional Backup Server messages and how to respond to them, see the description of `sp_volchanged` in the *SQL Server Reference Manual*.

Checking Database Consistency

SQL Server Manager enables you to run the Database Consistency Checker (`dbcc`), a set of utility commands that check the logical and physical consistency of a database. Use the `dbcc` commands as follows:

- As part of regular database maintenance (periodic checks run by a System Administrator or database owner). These checks can detect, and often correct, errors before they affect a user's ability to use SQL Server.
- To determine the extent of possible damage after a system error has occurred.
- Before backing up a database.
- When you suspect that a database is damaged. For example, if using a particular table generates the message "Table corrupt," use `dbcc` to determine if other tables in the database are also damaged.

This section discusses the `dbcc` command options that deal with overall database consistency. It gives instructions for executing the commands and then describes each command.

Execute these commands from the Database menu. Additional database command options enable you to check on the consistency of tables and indexes, and in SQL Server 11.0, network I/O tasks; these are described in Chapter 8, "Controlling Access" and Chapter 4, "Managing the SQL Server Operating Environment."

Instructions:

Privileges	A System Administrator or database owner can execute database <code>dbcc</code> commands.
------------	---

To check database consistency

1. Select the icon of the database to check.
2. From the Database menu, choose Consistency... The Database Consistency Check dialog box opens.

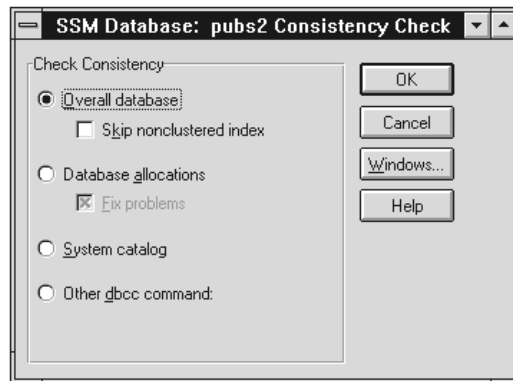
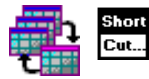


Figure 7-19: Database Consistency Check dialog box

3. Select the option button for the type of consistency check to perform and enter any other specifications that apply to that type. For consistency check details, see “Consistency Check Options” on page 7-52.
4. Click OK. SQL Server executes the `dbcc` command. When it completes, SQL Server Manager displays the results in the Output dialog box. The command option descriptions that follow include sample report output.

Shortcuts



Click the right mouse button over the database icon and choose Consistency... from the shortcut menu.



Click the Consistency... button in the Backup dialog box for the database.

Consistency Check Options

The following `dbcc` command options are available for consistency checking at the database level:

- Overall Database
- Database Allocations
- System Catalog
- Other `dbcc` Command

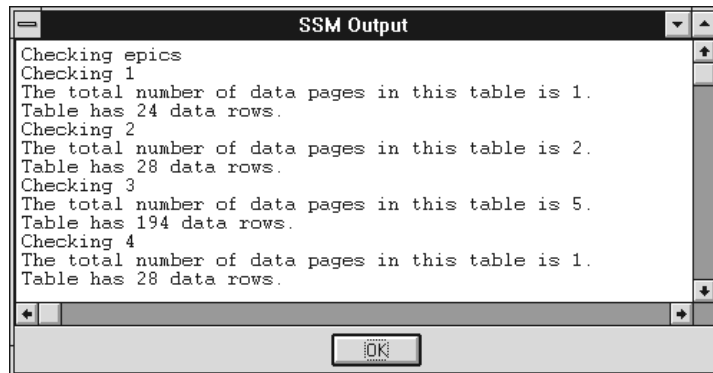
Overall Database Option

This option corresponds to the `dbcc checkdb` command option. This option checks each table in the selected database to see that:

- Index and data pages are correctly linked
- Indexes are in properly sorted order
- All pointers are consistent
- Information on internal allocation pages is synchronized with data rows

To skip checking nonclustered indexes on user tables, select the Skip Nonclustered Index box. If you leave the box unselected, the `dbcc` command checks all indexes on all tables.

The report for each undamaged table shows the number of data pages used and the number of data rows for the table. You can copy and paste this text to a word processor or text editor.



```
SSM Output
Checking epics
Checking 1
The total number of data pages in this table is 1.
Table has 24 data rows.
Checking 2
The total number of data pages in this table is 2.
Table has 28 data rows.
Checking 3
The total number of data pages in this table is 5.
Table has 194 data rows.
Checking 4
The total number of data pages in this table is 1.
Table has 28 data rows.
```

Figure 7-20: dbcc checkdb output

Database Allocations Option

This option corresponds to the `dbcc checkalloc` command option. This option checks the selected database to see that:

- All pages are correctly allocated
- No page is allocated that is not used
- No page is used that is not allocated

By default, `dbcc checkalloc` fixes allocation errors as it executes. SQL Server Manager automatically places the database in single-user mode while executing `dbcc checkalloc` and then returns the database to multiple-user mode when processing is complete. To generate a report without fixing allocation errors, clear the Fix Problems box.

The report shows the amount of space allocated and used by each database table. The output consists of a block of data for each table, including the system tables, and the indexes on each table. For each table or index, it reports the number of pages and extents (8-page blocks of allocated space) used. You can copy and paste this text to a word processor or text editor.

► **Note**

Because this **dbcc** option uses a large amount of overhead, run it in periods of low activity.

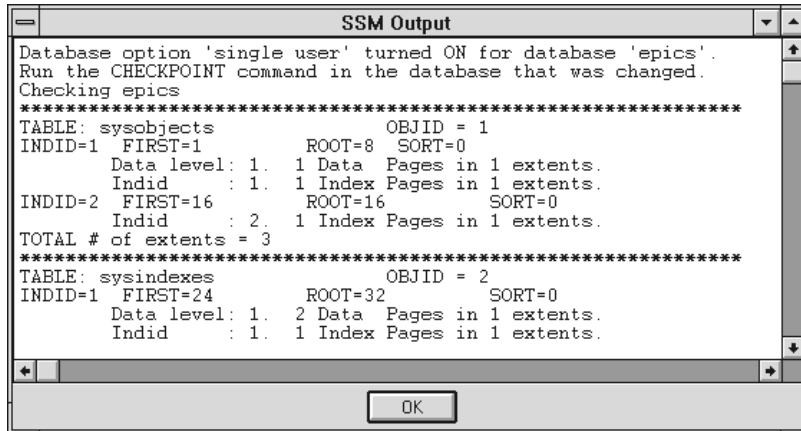


Figure 7-21: dbcc checkalloc output

System Catalog Option

This option corresponds to the **dbcc checkcatalog** command option. This option checks for consistency within and between the system tables found in a database. The report created by this option lists the segments defined for use by the database. You can copy and paste this text to a word processor or text editor.

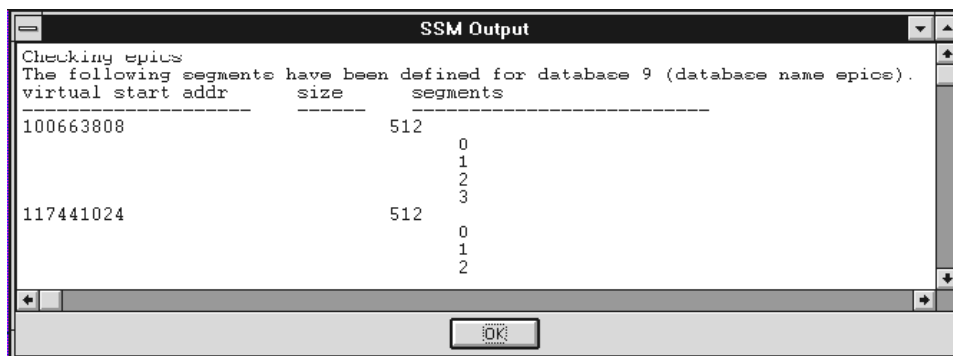


Figure 7-22: dbcc checkcatalog output

Other dbcc Command Option

This option allows you to enter the text of other dbcc command options. When you select Other dbcc Command, an edit box appears below the option button. Enter the text of the entire dbcc command option to execute, including the dbcc keyword.



Setting a Database Online

Restoring a database or transaction log requires that the database in which it is performed be set offline. When connected to SQL Server 11.0, SQL Server Manager automatically sets the database offline. When the restore operation is complete, you must manually return the database to online status.

Similarly, it is possible that database administration activities performed outside SQL Server Manager can leave a database in the offline state. If you receive an error message in SQL Server Manager indicating that the requested database is not online, you can set it back online by using a menu command.

To set a database online:

1. Select the icon of the database.
2. From the Database menu, choose Set Online. The following confirmation message appears:

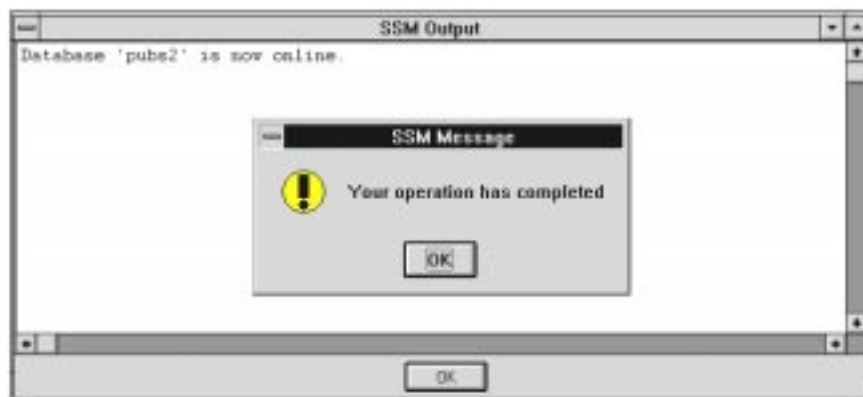


Figure 7-23: Confirmation message: setting a database online

Shortcut

Click the right mouse button over the database icon. Choose Set Online from the shortcut menu.

8

Controlling Access

What's in This Chapter

Access to SQL Server, databases, and database objects is controlled through privileges that users receive automatically and through permissions that are assigned explicitly. Automatic privileges come with the role associated with the user's login or come with database or object ownership. Explicit permissions are granted by authorized users.

For access to SQL Server, a SQL Server Manager user must have a SQL Server login. For access to a database or database objects, a SQL Server Manager user must be a user in the database. Unless a database user has automatic privileges in the database because of role assignment or database or object ownership, the user must be granted explicit permissions. Therefore, to set up a new SQL Server Manager user to use a database, do the following:

- Create a new SQL Server login for the user and, optionally, assign roles or aliases to the login.
- Add the user to the database, and, optionally, add the user to a group.
- Grant permissions to the user to use commands and to use database objects.

This chapter describes how to create, modify, and delete SQL Server logins, users, and groups. It also describes how to control access through roles, shared logins (aliases), command permissions, and object permissions. Table 8-2 on page 8-41 summarizes permissions.

About Access Privileges and Permissions

Access privileges and permissions provide the ability to use SQL Server commands to create, view, and modify SQL Server and database objects. A user has access to commands and objects depending on the following:

- SQL Server administrative role
- Status as a user (database owner or object owner)
- Group membership
- Specific command permissions granted

- Specific object permissions granted

Role assignment takes place at the SQL Server login level. Users have special user status when they become owners of a database or database object. Authorized SQL Server Manager users can grant command and object permissions to a database user directly, to a group to which the user belongs, or to another user whose login (alias) the user can use. This section describes access based on each of these categories of privilege or permission.

Access Based on Administrative Role

Three SQL Server roles help with the division of administrative responsibilities. A user can have multiple roles simultaneously. The roles are as follows:

- **System Administrator**—a user with the System Administrator role performs administrative tasks and has privileges for almost every SQL Server Manager function. Privileges acquired by a user with the System Administrator role are:
 - Managing disk space
 - Creating, deleting, and modifying logins
 - Granting or revoking the System Administrator role
 - Creating remote servers
 - Creating user databases and transferring ownership of them
 - Creating and managing the membership of groups
 - Granting permissions to SQL Server users
 - Executing the `dbcc` utility
 - Changing SQL Server configuration parameters
 - Shutting down SQL Server or killing SQL Server processes
 - Monitoring recovery

Additionally, SQL Server treats users with the System Administrator role as database owners in any database.

- **System Security Officer**—a user with the System Security Officer role performs security-related tasks. Privileges acquired by a user with the System Security Officer role are:
 - Creating and modifying logins and passwords
 - Setting the SQL Server password expiration interval

- Setting the remote server option that controls whether a connection is maintained in the absence of login activity
- Granting and revoking administrative roles
- Assigning aliases
- **Operator**—a user with the Operator role can back up and restore all databases and transaction logs.

To assign roles to a login by using SQL Server Manager, use the Create Login dialog box or the Properties tab of the Login dialog box. For details, see “Creating a SQL Server Login” on page 8-8 and “Modifying Login Information” on page 8-11.

Access Based on Special User Status

Some access privileges apply to users who assume a special status because they own a database or database object.

- **Database owner**—users who own databases have authority over the databases they own. Database owners have full permissions on objects inside their databases. Within their own databases, database owners can:
 - Issue a database checkpoint
 - Execute database-level dbcc commands
 - Delete a database
 - Create database users
 - Create database objects
 - Grant and revoke permission to create tables, views, defaults, procedures, and rules
 - Grant and revoke permissions on database objects
 - Back up and load the database and its transaction log

The creator of a database is its initial owner. Only a user with the System Administrator role can create a database. The System Administrator can then transfer database ownership to another user. See Chapter 7, “Managing Databases and Segments.”

- **Object owner**—a user who creates a database object owns the object and automatically has all object permissions on it. For a description of object permissions, see “Access Based on Object Permissions” on page 8-5. Users other than the object owner, including the owner of the database, must explicitly be granted each type of permission on the object. Along with object permissions, the owner of an object has the ability to:
 - Grant and revoke object permissions
 - Delete the object

The owner of a database table has the following additional permissions:

- Adding columns to the table
- Creating indexes and triggers on the table
- Executing the `update statistics` command

To become an object owner, a user:

- Must be the owner of a database, or
- Must be granted permission to create an object by a database owner or a user with the System Administrator role.

Access Based on Group Membership

Users who are members of a group take on the permissions explicitly granted to the group. Group permissions on an object override permissions granted to individual users on the same object.

Access Based on Command Permissions

To create databases or database objects, users must either have privileges based on role, special user status, or group membership, or be granted explicit permission to execute the corresponding `create` command.

A user with the System Administrator role can grant or revoke permission to create a database.

A database owner can grant to and revoke from other users or groups the permission to use the commands that create database tables, defaults, rules, procedures, and views.

In SQL Server Manager, use the following dialog boxes to grant or revoke command permissions:

- To grant permissions to or revoke permissions from a user, use the Command Permissions tab of the User dialog box. For details, see “Granting and Revoking Command Permissions for a User” on page 8-27.
- To grant permissions to or revoke permissions from a group, use the Command Permissions tab of the Group dialog box. For details, see “Granting and Revoking Command Permissions for a Group” on page 8-39.

Access Based on Object Permissions

To obtain access to database tables, views, and procedures, users must either have privileges based on role, special user status, or group membership, or be granted explicit permission for each type of access. Database and object owners can grant or revoke permission on objects they own.

For each permission granted, the grantor can specify that the recipient can grant the permission to another user. This form of granting permission is called granting **with grant**. When revoking permission, the revoker can specify that the permission be revoked from all users to whom the recipient granted it. In SQL Server Manager, this form of revoking permissions is called revoking **with cascade**.

You can use SQL Server Manager to grant or revoke the following object permissions:

- Select—select data from a table or view
- Insert—insert a row in a table or view
- Delete—delete a row in a table or view
- Update—update a row in a table or view
- Execute—execute a procedure
- Reference—create referential constraints on a table or view

When assigning permissions, you can update users or groups, or you can update the object to which permissions apply. For example, when creating a new user, you can assign all of the user’s permissions from a single user dialog box. When creating a new table, you can assign permissions to each authorized user from the

Table dialog box. The following table shows which dialog boxes to use for granting or revoking object permissions:

Table 8-1: Object permission dialog boxes

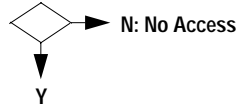
Method for assigning permissions	Dialog Box tab
Update a user	User dialog box Object Permissions tab
Update a group	Group dialog box Object Permissions tab
Update a table	Table dialog box Permissions tab
Update a view	View dialog box Permissions tab
Update a procedure	Procedure dialog box Permissions tab

Access Example

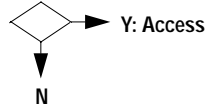
The following example illustrates several ways a user can have access to a database object:

User Homer wants to create a table in the Epics database.

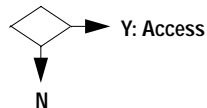
Is Homer a user in Epics?



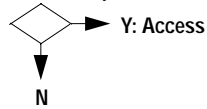
Is Homer associated with a login that has the System Administrator role?



Is Homer the owner of Epics?



Is Homer a member of a group that has been granted permission to create a table in Epics?



Has Homer been granted permission to create a table in Epics?

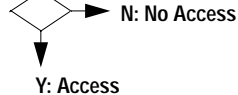


Figure 8-1: Obtaining access to a database object

Creating and Deleting SQL Server Logins

Creating a SQL Server login is the first step in giving a SQL Server Manager user access to SQL Server. This section describes how to create and delete a login.

Creating a SQL Server Login

Privileges Only a System Security Officer can create a SQL Server login.

To create a SQL Server login:

1. Select the icon of the SQL Server for which to create the login.
2. Choose Login from the Create menu that cascades from the Server menu. The Create Login dialog box opens:

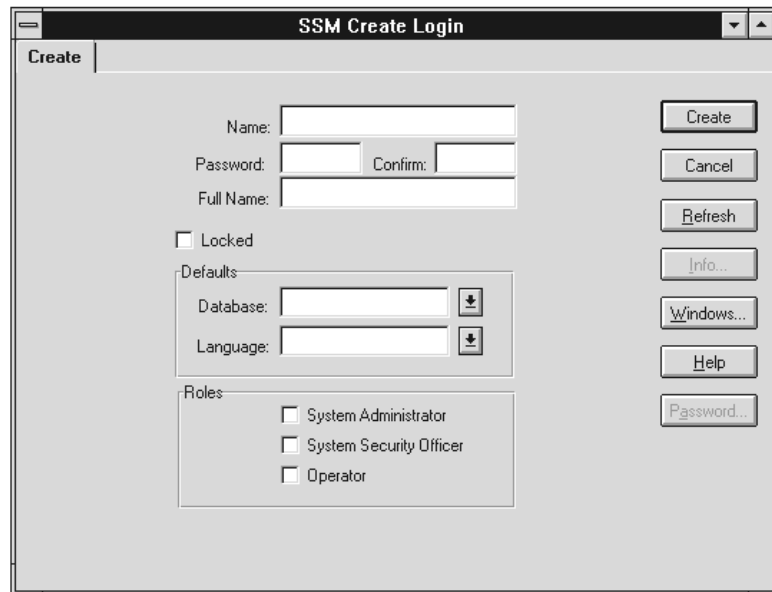


Figure 8-2: Create Login dialog box

3. Enter the login in the Name edit box.
4. Enter the login password in the Password edit box.
5. Re-enter the password in the Confirm box.
6. Enter the full name of the person represented by the login in the Full Name edit box.
7. To lock a login (that is, to prevent it from obtaining any SQL Server access), check the Locked check box. For information on why you might want to lock a login, see “Locking a SQL Server Login” on page 8-12.

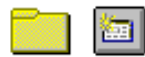
8. Select the database that the login points to automatically on connecting to SQL Server from the list in the Database box.

► **Note**

If you do not specify a default database, SQL Server assigns *master* as the default. Allowing the database to default to *master* is not recommended.

9. Choose a language for the login from the list in the Language box.
10. To assign one or more roles to the login, check the appropriate boxes in the Roles group box. For a summary of the privileges conferred by each role, see “Access Based on Administrative Role” on page 8-2.
11. Click Create to add the new login.

Shortcuts



Click the Login container icon in the appropriate SQL Server; then, select the Create toolbar button.



Click the right mouse button over the Login container icon in the appropriate SQL Server; then, choose Create from the shortcut menu.



Click the right mouse button over the appropriate SQL Server icon. Choose Create; then, choose Login.

Deleting a SQL Server Login

Before you can delete a login, you must do the following for each user with which the login is associated:

- Revoke the user’s command and object permissions with cascade. Revoking permissions with cascade revokes them from users to whom this user has granted them.
- Re-grant the revoked permissions to the other users, if appropriate.
- Delete the user’s objects.

- Delete the user from all databases.

Privileges	Only a System Administrator can delete a login.
------------	---

To delete a login:

1. Select the icon of the login to delete.
2. Choose Delete... from the Login menu.
3. Respond Yes to the confirmation dialog box prompt.

Shortcuts

Select the login icon. Select the Delete toolbar button.



Click the right mouse button over the login icon. Choose Delete from the shortcut menu.



Displaying and Modifying Login Information

Once a login has been created, users with the System Security Officer role can lock or unlock the login or change the defaults, the password, and the user role. System Security Officers can also assign additional users (aliases) to the login.

Also, users can navigate to databases owned by the login and to users for whom the login is an alias.

Modifying Login Information

To change the fullname, defaults, or role assignments of a login:

1. Select the login icon you want to modify.

Privileges A System Security Officer can modify a SQL Server login.

2. From the Login menu, choose Properties...

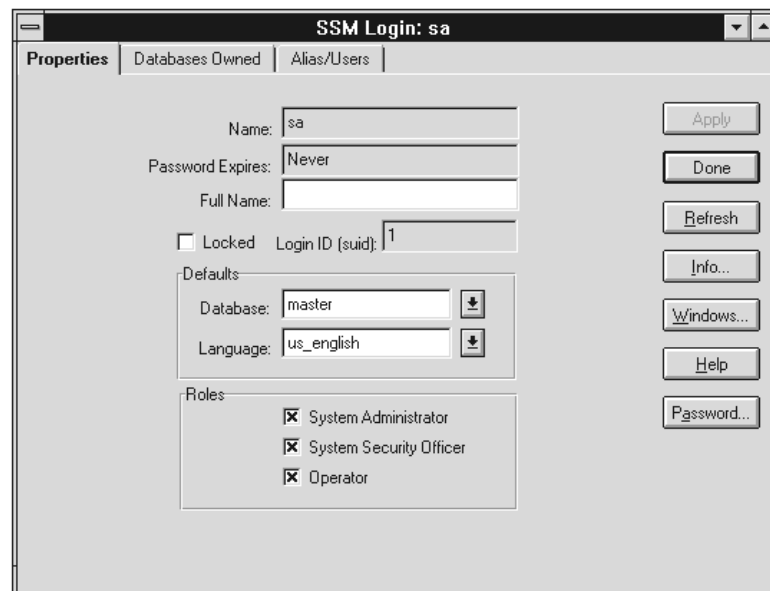





Figure 8-3: Login Properties tab

3. To change the person's name, enter the new name in the Full Name box.
4. To change the database that the login points to automatically on connecting to SQL Server, select the new database name from the list in the Database box.
5. To change the language for the login on connecting to SQL Server, select the new language from the list in the Language box.
6. To change the login's role assignment, select or clear the appropriate boxes in the Roles group box. For a summary of the

privileges assumed by each role, see “Access Based on Administrative Role” on page 8-2.

7. Click Apply to change the login.

Shortcuts

	Double-click the login icon.
	Select the login icon, and select the Properties toolbar button.
	Click the right mouse button over the login icon. Choose Properties... from the shortcut menu.

Locking a SQL Server Login

Locking a SQL Server login is a way to disable it without removing it from the *master* database. It may be easier to lock a login rather than delete it for the following reasons:

- You cannot delete the login if a user exists for that login in any database.
- You cannot delete a user from a database if the user owns any objects in the database or has granted permissions on objects to other users.
- SQL Server could compromise accountability by reusing a deleted login server user ID (suid) when the next login is created.
- You cannot delete the last remaining System Security Officer or System Administrator login.

Another common reason to lock a login is to implement roles. When you first install SQL Server, the default “sa” login has all privileges associated with every role. At many installations, after the “sa” login makes the initial role assignments, a user with the System Administrator or System Security Officer role locks the “sa” login, so that no single SQL Server Manager user has every available privilege and so that actions can be traced.

To lock or unlock a login:

1. Display the Login dialog box, as described in “Modifying Login Information” on page 8-11.
2. Select or clear the Locked check box.
3. Click Apply to lock or unlock the login.

► **Note**

You can lock an account that is logged in—the user is not locked out until he or she logs out. You can lock an account that owns a database and objects in databases.

Changing the Login Password

To change the login password:

1. Display the Login dialog box, as described in “Modifying Login Information” on page 8-11.
2. Click Password.... The Change Password dialog box opens.

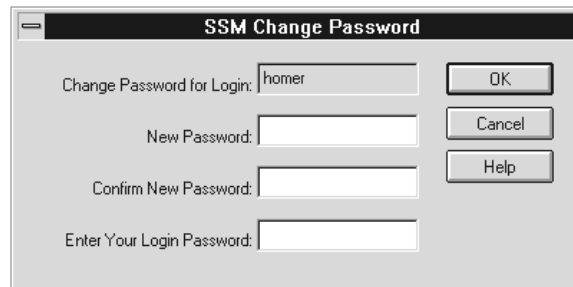


Figure 8-4: Change Password dialog box

3. Enter a new password in the New Password box.
4. Re-enter the password in the Confirm New Password box.
5. Enter your own login password in the Enter Your Login Password box.
6. Click OK to change the password.

Displaying Databases Owned by a Login

To display the databases a login owns and navigate to them:

1. Display the Login dialog box, as described in “Modifying Login Information” on page 8-11.
2. Select Databases Owned to display the Databases Owned tab.

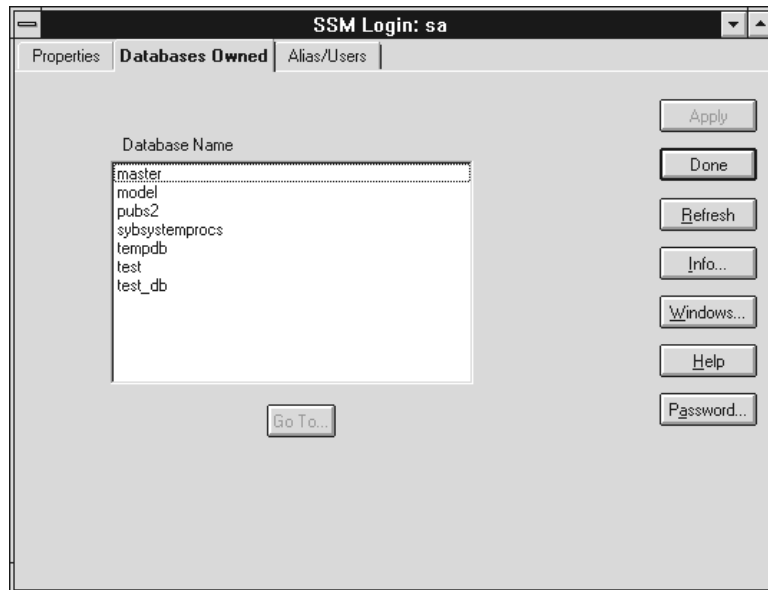


Figure 8-5: Login Databases Owned tab

3. To open the Properties tab for the database, select a database in the list of databases owned and click Go To...

Sharing Logins by Using Aliases

Using aliases enables more than one person to share identical privileges within a database. For example, to give several logins the automatic privileges associated with ownership of a database, you can associate those logins with the database owner, thus making each login an alias to the database owner. This section describes how to add aliases to a login and how to navigate to aliased users.

Privileges	A System Security Officer can update the aliases mapped to a login.
------------	---

Adding an Alias to a Login

To add an alias for a login:

1. Display the Login dialog box, as described in “Modifying Login Information” on page 8-11.
2. Select Alias/Users to display the Alias/Users tab.

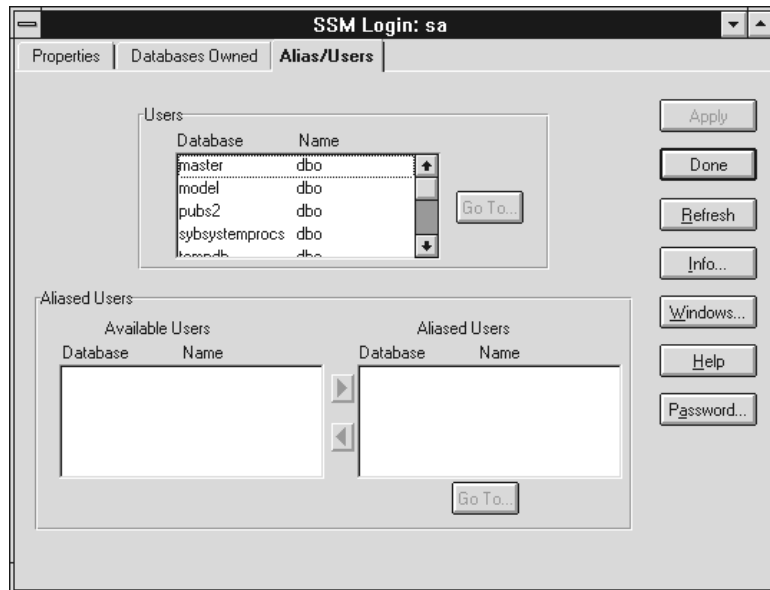


Figure 8-6: Login Alias/Users tab


This tab contains three scrolling lists of users and their databases:

- The list in the Users group box shows the users assigned to the login during user creation. In each database, only one user is assigned to the login in this way.
 - The Available Users list in the Aliased Users group box shows the users not yet aliased to the login. Users in the databases shown in the Users list are excluded from the Available Users list.
 - The list in the Aliased Users group box shows the users for whom the login is now an alias.
3. Select a database and user row in the Available Users list.


4. Click the right-pointing arrow to move the user to the Aliased Users list.
5. Click Apply to add the alias to the login.

Shortcuts


Double-click the login icon.



Select the login icon, and select the Properties toolbar button.



Click the right mouse button over the login icon. Choose Properties... from the shortcut menu.



Deleting an Alias from a Login

To delete an aliased user from the login:

1. Display the Login dialog box, as described in “Modifying Login Information” on page 8-11.
2. Select Alias/Users to display the Alias/Users tab.
3. Select a database and user row in the Aliased Users list.
4. Click the left-pointing arrow to move the user to the Available Users list.
5. Click Apply to delete the alias from the login.

Navigating to Users

From the Alias/Users tab of the Login dialog box, you can navigate to users associated with the login in two ways:

- Users that were assigned to the login when each user was created
- Users who are aliases for the login

First, display the Alias/Users tab:

1. Display the Login dialog box, as described in “Modifying Login Information” on page 8-11.
2. Select Alias/Users. The dialog box changes to the Alias/Users tab.

Then, perform the steps in “Navigating to Users Assigned at User Creation” or “Navigating to Aliased Users” as follows:

Navigating to Users Assigned at User Creation

1. Select a user name in the Users list at the top of the Alias/Users tab.
2. Click the Go To... button to display the Properties tab for the user.

Navigating to Aliased Users

1. Select a user name in the Aliased Users scrolling list in the Aliased Users group box.
2. Click the Go To... button to display the Properties tab for the user.

Creating and Deleting Users

A common practice for setting users up in a database is to:

1. Add the user to the database.
2. Grant the user specific permissions on commands and objects.

This section describes how to create and delete a database user.:

Privileges	Database owners can add and delete users in the databases they own.
------------	---

Creating a User

To create a new user:

1. From the Database menu, choose Create; then, choose User from the cascading menu. The Create User dialog box opens.

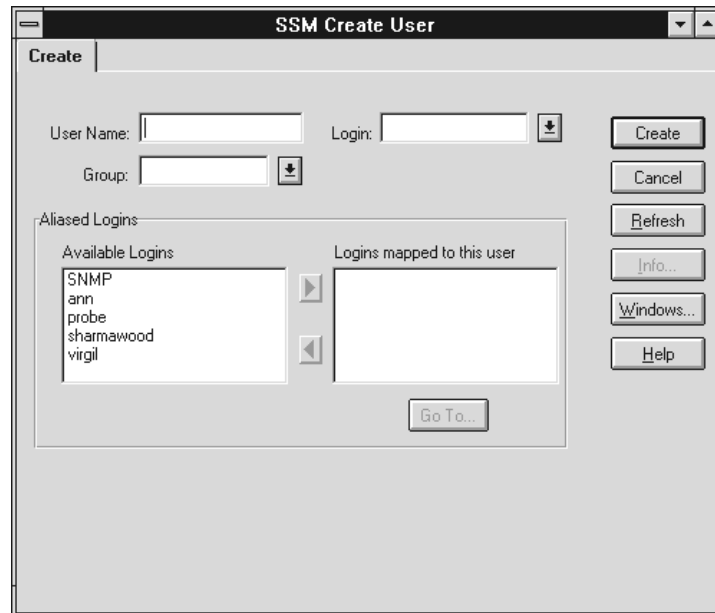


Figure 8-7: Create User dialog box





2. Enter the user's name in the User Name box.
3. Select the user's login from the list in the Login box.
4. To assign the user to a group, select a group from the drop-down Group list box. If you do not assign a group to the user, the user is created as a member of only the "public" group, which has minimal database privileges.

► **Note**

A user can be a member of one assigned group or the default "public" group. The group permissions override the user's permissions on the same objects.

5. To associate the user as an alias with one or more logins:
 1. Select a login in the Available Logins list.
 2. Click the right-pointing arrow to move the login to the Logins Mapped to This User box.
 3. Repeat these three steps for each login you want to assign to the new user.
6. Click Create to create the user.

Shortcuts

	<p>Select the Users container icon in the appropriate database, and select the Create toolbar button.</p>
	<p>Drag the login icon that corresponds to the user onto the database icon.</p>
	<p>Click the right mouse button over the Users container icon in the appropriate database, and choose Create from the shortcut menu.</p>
	<p>Click the right mouse button over the Database icon, and choose Create from the shortcut menu.</p>

Deleting a User

You cannot delete a user who owns objects. Since there is no command to transfer ownership of objects, you must delete objects owned by a user before you can delete the user. Also, you cannot delete a user who has granted permissions to other users without first revoking the permissions with cascade. If appropriate, re-grant the permissions to the other users.

Locking a login is a simple alternative to deleting a user. For details, see “Locking a SQL Server Login” on page 8-12. If you decide to delete a user rather than lock a login, you must do the following before you can delete the user:

- Revoke the user’s command and object permissions with cascade.

- Re-grant the permissions to the other users, if appropriate.
- Delete the user's objects.

To delete a user:

1. Open the User container icon for the user you want to delete.
2. Click the user icon. The menu bar changes to the User menu.
3. Choose Delete from the menu.
4. Click Yes in the confirmation dialog box.

Shortcuts



Select the icon of the user to delete, and select the Delete toolbar button.



Click the right mouse button over the user icon, and choose Delete from the shortcut menu.

Displaying and Modifying User Information

You can display and modify information about users and about their group membership, logins, permissions, and objects owned. This section describes how to:

- Display user properties
- Modify a user's group and alias associations
- Display and navigate to owned objects
- Grant and revoke object and command permissions

Displaying User Properties

To display information about how the user is defined:

1. Select the icon of the user.
2. From the User menu, choose Properties...

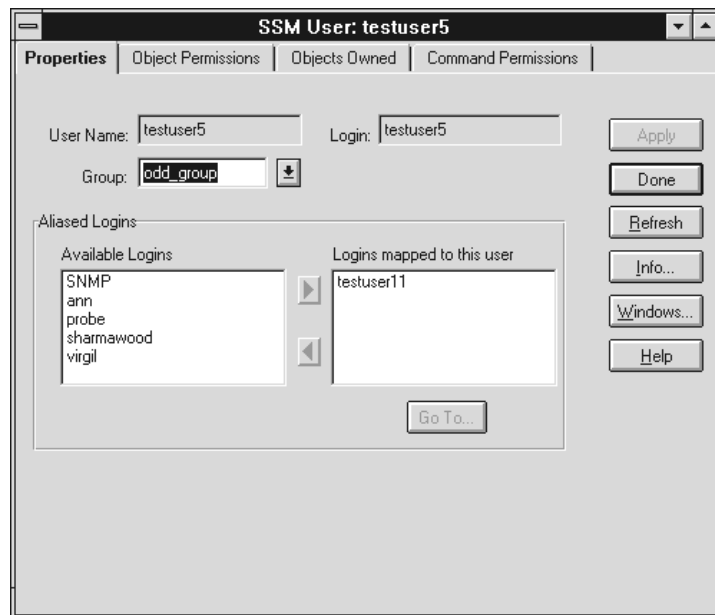


Figure 8-8: User Properties tab

The Properties tab shows the user's name and login, as well as the group (if any) to which the user belongs, and any aliased logins assigned to the user.

Modifying Group Membership and User Aliases

The user properties that you can change are the user's group membership and aliased logins.

Privileges	Database owners can modify a user's group and aliases in databases they own.
------------	--

To modify these properties:

1. Display the User dialog box, as described in “Displaying User Properties” on page 8-21.
2. To change the user’s group, select a group in the Group list. The user is added to the new group and removed from the previous group.
3. To add an alias:
 1. Select a login in the Available Logins list.
 2. Click the right-pointing arrow to move the login to the Logins Mapped to This User list.
4. To delete an alias:
 1. Select a login in the Logins Mapped to This User list.
 2. Click the left-pointing arrow to move the login to the Available Logins list.
5. Click Apply to update the user.

Navigating to Logins (Aliases)

To navigate to one of the user’s aliased logins:

1. Select a login in the Logins Mapped to This User list.
2. Click Go To... to display the Properties tab for the login.

Displaying Objects Owned by the User

To see a list of the objects a user owns and navigate to specific objects:

1. Display the User dialog box, as described in “Displaying User Properties” on page 8-21.
2. Select Objects Owned. The dialog box changes to the Objects Owned tab.

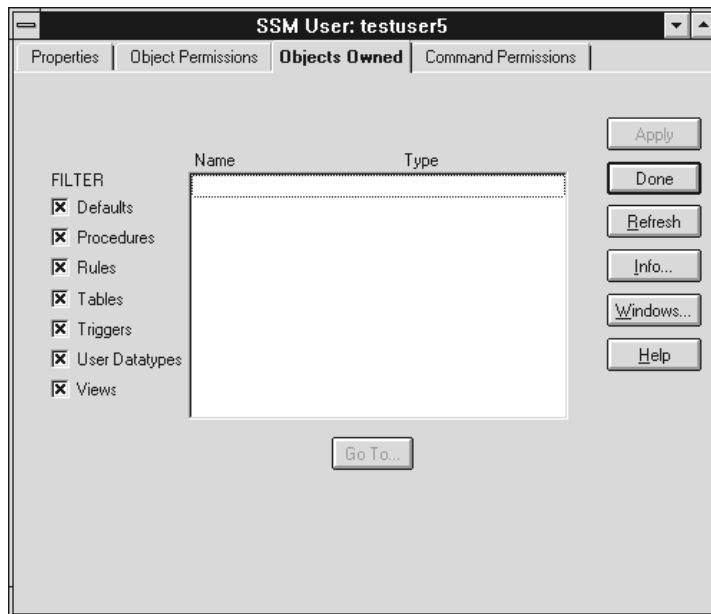


Figure 8-9: User Objects Owned tab

3. To limit the list of displayed objects to specific object types, use the check boxes under the Filter label. By default all object types are selected. Clear the check box for each type of object to exclude from the display.

Navigating to an Owned Object

To navigate to one of the objects in the list:

1. Select an object in the list.
2. Click Go To... to display the Properties tab for the object.

Granting and Revoking Object Permissions for a User

Object permissions govern access to database objects. When assigning permissions to a user on an object, object owners can update either the user or the object. This section describes assigning permissions by updating database users. For information on assigning permissions by updating database objects, see the description of each object in Chapter 9, “Managing Database Objects.”

Privileges Object owners can assign privileges on the objects they own.

To grant or revoke object permissions:

1. Display the User dialog box, as described in “Displaying User Properties” on page 8-21.
2. Select Object Permissions. The dialog box changes to the Object Permissions tab.

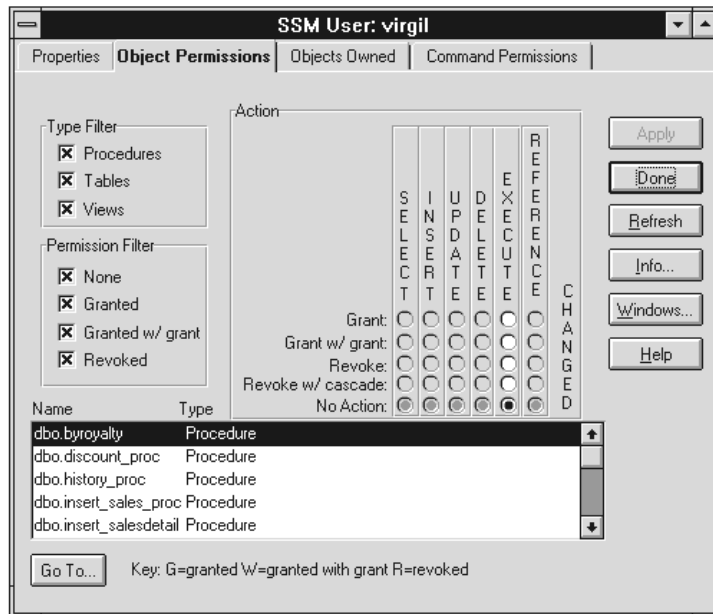


Figure 8-10: User Object Permissions tab

3. Initially, the dialog box shows all possible objects. To limit the displayed list:
 - In the Type Filter group box, clear the types of objects you want to filter out of the display.
 - In the Permission Filter group box, clear the options for the objects you do not want to see:
 - None—objects on which the user has no permissions
 - Granted—objects on which the user has been granted at least one type of permission
 - Grant w/Grant—objects on which the user has been granted at least one type of permission with grant option
 - Revoked—objects on which the user has had at least one type of permission revoked
4. Select the object on which you want to change the user's permissions.
5. To grant a permission on the object, select the Grant button for that permission in the Action group box. To grant the permission with the **grant option** (that is, to allow the user to extend the same permission to other users), select the Grant w/Grant option button for that permission.

On procedures, you can grant permission to execute the procedure. On tables and views, you can grant the following types of permissions:

 - Select—permission to select a table or view
 - Insert—permission to insert a row
 - Update—permission to update a row
 - Delete—permission to delete a row
 - Reference—permission to create referential constraints

When you make a permission change for the user, the Changed column displays a check mark (√), indicating that you have altered the original settings.
6. To revoke a permission on the object, select the Revoke option button for that permission in the Action group box. To revoke the permission **with cascade** (that is, to revoke permission from any users to whom this user has extended the same permission), select the Revoke w/Cascade option button for that permission.

When you make a permission change for the user, the **Changed** column displays a check mark (√), indicating that you have altered the original settings.

7. Click **Apply** to update the user's permissions in the database. SQL Server Manager updates the list with one of the following codes in each column that corresponds to the type of permission you changed:
 - G—granted
 - W—granted with grant
 - R—revoked
 - Blank—no permission explicitly assigned

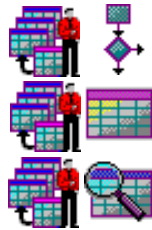
► **Note**

SQL Server Manager reports on permissions explicitly granted or revoked. It does not show permissions that users obtain by being members in a group or by having an administrative role, such as System Administrator, associated with their logins.

8. Repeat Steps 4 through 7 for each object on which you want to update permissions.
9. Click **Done** to exit the dialog box.

Shortcuts

Drag and drop the user icon onto the object icon.



Navigating to Object Permissions Information

To navigate to an object from the User dialog box:

1. Select Object Permissions on the User dialog box. The dialog box changes to the Object Permissions tab.
2. To limit the displayed list:
 - In the Type Filter group box, clear the types of objects you want to filter out of the display.
 - In the Permission Filter group box, clear the options for the objects you do not want to see:
 - None—objects on which the user has no permissions
 - Granted—objects on which the user has been granted at least one type of permission
 - Grant w/Grant—objects on which the user has been granted at least one type of permission with grant
 - Revoked—objects on which the user has had at least one type of permission revoked
3. Select the object in the Name list, then click Go To... to display the Properties tab for the object.

Granting and Revoking Command Permissions for a User

Command permissions govern the ability to execute create commands for a database or database objects.

Privileges	System Administrators and database owners can grant command permissions. Only a System Administrator can grant permission to create a database. Database owners can grant permission to create database objects in the databases they own.
------------	--

► **Note**

Permission to create a database applies only to users of the *master* database.

To manage command permissions:

1. Display the User dialog box, as described in “Displaying User Properties” on page 8-21.
2. Select Command Permissions. The dialog box changes to the Command Permissions tab.

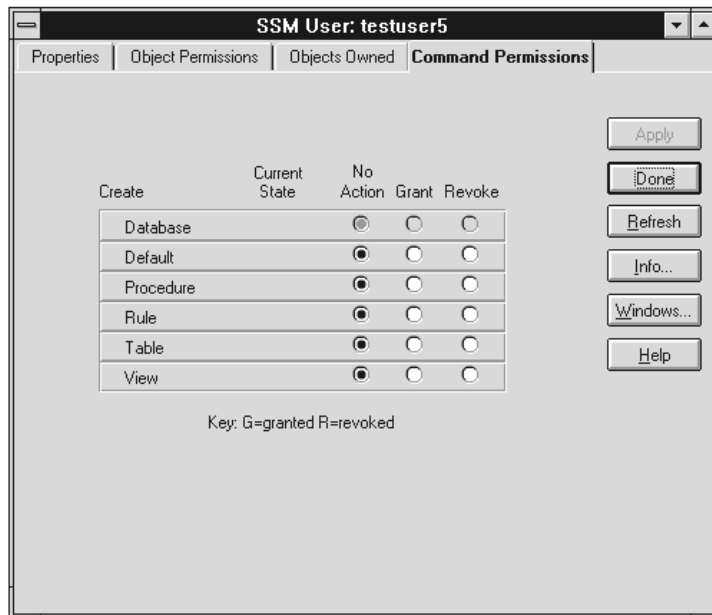


Figure 8-11: User Command Permissions tab

3. To grant permission, select the option button in the Grant column corresponding to the permission.
4. To revoke permission, select the option button in the Revoke column corresponding to the permission.
5. Click Apply to change the permissions in the database. SQL Server Manager updates the Current State column to reflect the new level of permission:
 - G—granted
 - R—revoked
 - Blank—no permission explicitly assigned

► Note

SQL Server Manager reports on permissions explicitly granted or revoked. It does not show permissions that users obtain by being members in a group or by having an administrative role, such as System Administrator, associated with their logins.

Creating and Deleting Groups

Groups provide a convenient way to grant and revoke permissions to more than one user at once. Each user can be a member of one assigned group or the “public” group (a default group with minimal privileges).

Privileges	Database owners can create and delete groups in databases they own.
------------	---

It may be convenient to create groups before adding users to a database so you can assign new users to existing groups.

Creating a Group

To create a group:

1. Select the icon of the database in which to create the group.
2. From the Database menu, choose Create; then, choose Group from the cascading menu. The Create Group dialog box opens.

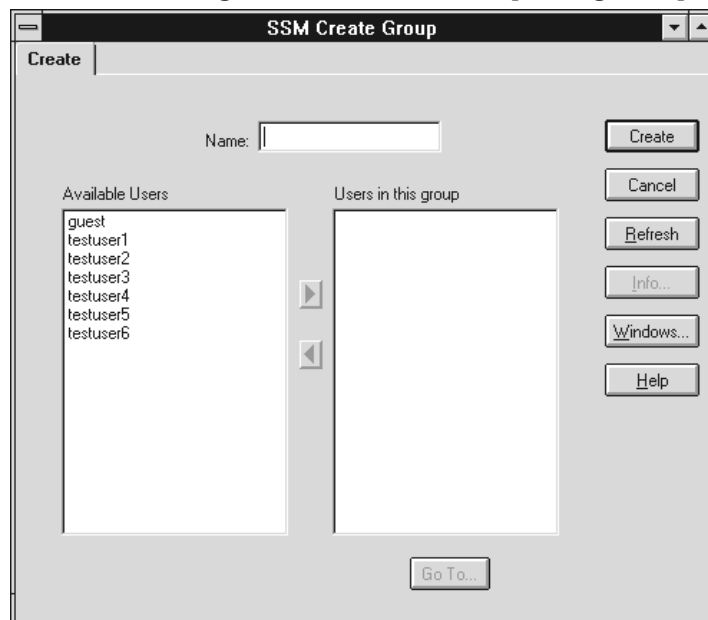
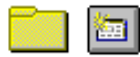


Figure 8-12: Create Group dialog box

3. Enter the group's name in the Name box.
4. Select a user in the Available Users list and click the right-pointing arrow to add the user to the group. If the user was in another group, the user is removed from the old group.
5. Repeat Step 3 for each user you want to add to the group.
6. Click Create to create the group.

Shortcuts



Select the Group container icon in the appropriate database, and select the Create button.



Click the right mouse button over the Group container icon. Choose Create from the shortcut menu.



Click the right mouse button over the database icon. Choose Create from the shortcut menu; then, choose Group.

Deleting a Group

You cannot delete a group that has members. Before deleting a group, you must first remove all of its users.

Removing Users from a Group

Privileges	Database owners can modify group membership in databases they own.
------------	--

To remove the users from a group:

1. Click the icon of the group to modify.
2. From the Group menu, choose Properties...

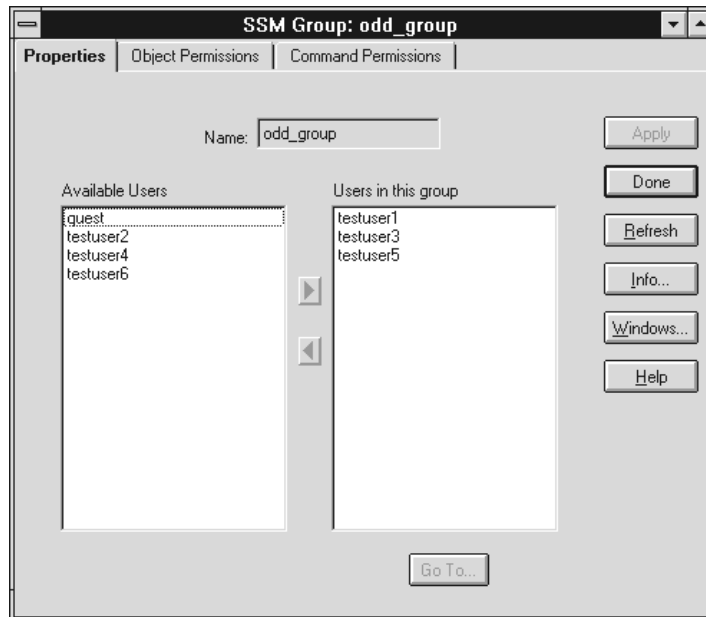






Figure 8-13: Group Properties tab

3. Select a user in the Users in This Group list and click the left-pointing arrow to remove the user from the group.
4. Repeat Step 3 for each user in the group.
5. Click Apply.

Deleting an Empty Group

1. Select the icon of the group to delete.
2. Choose Delete from the Group menu.
3. Click Yes in the confirmation dialog box.

Shortcuts

		Select the group icon; then, select the Delete toolbar button.
		Click the right mouse button over the group icon, and choose Delete from the shortcut menu.

Displaying and Modifying Group Information

You can display and modify information about groups, their membership, permissions, and objects owned. This section describes how to:

- Examine and update group membership
- Navigate to the users in a group
- Assign object and command permissions for a group

Displaying Group Properties

To display information about how the group is defined:

1. Click the icon of the group to display.
2. From the Group menu, choose Properties...

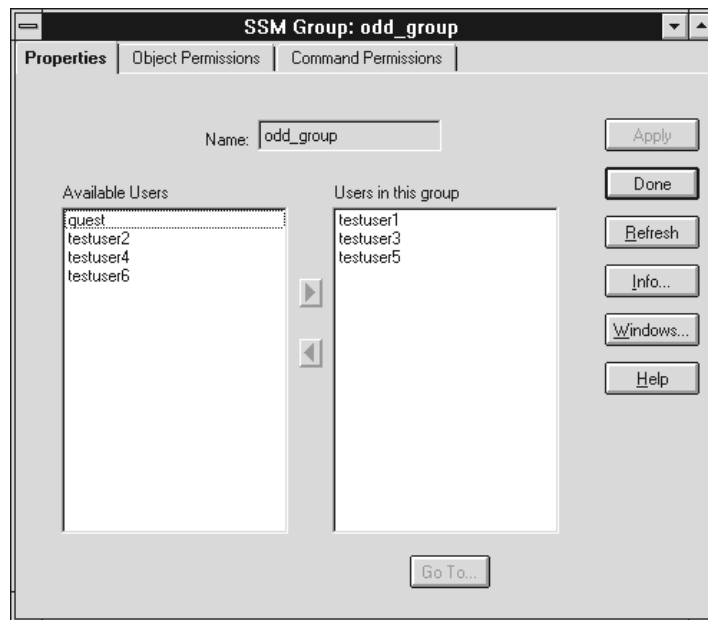





Figure 8-14: Group Properties tab

The Properties tab shows the group's name and users in the group, as well as all users in the database.

Shortcuts

	Double-click the group icon.
	Select the group icon, and select the Properties toolbar button.
	Click the right mouse button over the group icon; then, choose Properties... from the shortcut menu.

Navigating to Users

To navigate to one of the users:

1. Select a user in the Users in This group list.
2. Click Go To... to display the user's Properties tab.

Modifying Group Membership:

Privileges	Database owners can modify group membership in databases they own.
------------	--

To modify the membership of a group

1. Display the Group dialog box as described in "Displaying Group Properties" on page 8-34.
2. To add a user to the group, select a user in the Available Users list and click the right-pointing arrow to add the user to the Users in This Group list.
3. To remove a user from the group, select a user in the Users in This Group list and click the left-pointing arrow.
4. Repeat Step 2 or 3 for each user you want to add or remove.
5. Click Apply to update the group.

Granting and Revoking Object Permissions for a Group

Object permissions govern access to database objects. When assigning permissions to a group on an object, Object owners can either update the group or the object. This section describes assigning permissions by updating database groups. For information on assigning permissions by updating database objects, see the description of each object in Chapter 9, “Managing Database Objects.”

Privileges	Object owners can assign object permissions on groups they own.
------------	---

To grant or revoke object permissions:

1. Display the Group dialog box, as described in “Displaying Group Properties” on page 8-34.
2. Select Object Permissions. The dialog box changes to the Object Permissions tab.

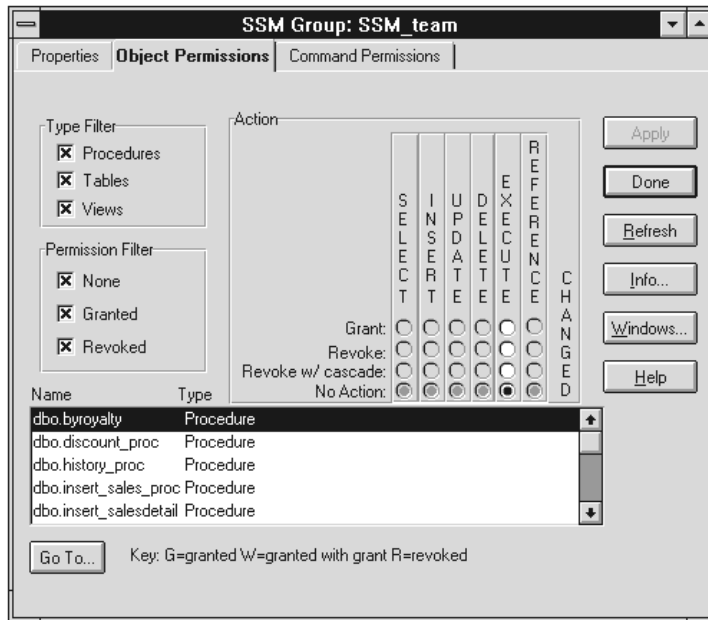


Figure 8-15: Group Object Permissions tab

3. Initially, the dialog box shows all possible objects. To limit the displayed list:
 - In the Type Filter group box, clear the types of objects you want to filter out of the display.
 - In the Permission Filter group box, clear the options for the objects you do not want to see:
 - None—objects on which the group has no permissions
 - Granted—objects on which the group has been granted at least one type of permission
 - Revoked—objects on which the group has had at least one type of permission revoked
4. Select the object on which you want to change the group's permissions.
5. To grant a permission on the object, select the Grant option button for that permission in the Action group box.

On procedures, you can grant permission to execute the procedure. On tables and views, you can grant the following types of permissions:

 - Select—permission to select a table or view
 - Insert—permission to insert a row
 - Update—permission to update a row
 - Delete—permission to delete a row
 - Reference—permission to create referential constraints

When you make a permission change for the group, the Changed column displays a check mark (√), indicating that you have altered the original settings.
6. To revoke a permission on the object, select the Revoke option button for that permission in the Action group box. To revoke the permission **with cascade** (that is, to revoke permission from any groups to whom this group has extended the same permission), select the Revoke w/Cascade option button for that permission.

When you make a permission change for the group, the Changed column displays a check mark (√), indicating that you have altered the original settings.
7. Click Apply to update the group's permissions in the database. SQL Server Manager updates the list with one of the following

codes in each column that corresponds to the type of permission you changed:

- G—granted
- R—revoked
- Blank—no permission explicitly assigned

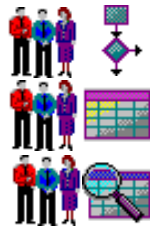
► **Note**

SQL Server Manager reports on permissions explicitly granted or revoked. It does not show permissions that users obtain by being members in a group or by having an administrative role, such as System Administrator, associated with their logins.

8. Repeat Steps 4 through 7 for each object on which you want to update permissions.
9. Click Done to exit the dialog box.

Shortcut

Drag and drop the group icon onto the object icon.



Navigating to Object Permissions Information

To navigate to an object from the group dialog box:

1. Display the Group dialog box, as described in “Displaying Group Properties” on page 8-34.
2. Select Object Permissions. The dialog box changes to the Object Permissions tab.
3. To limit the displayed list:
 - In the Type Filter group box, clear the types of objects you want to filter out of the display.

- In the Permission Filter group box, clear the objects you do not want to see:
 - None—objects on which the group has no permissions
 - Granted—objects on which the group has been granted at least one type of permission
 - Revoked—objects on which the group has had at least one type of permission revoked
- 4. Select the object in the Name list, then click Go To... to display the Properties tab for the object.

Granting and Revoking Command Permissions for a Group

Command permissions govern the ability to execute create commands for a database or database objects.

Privileges	Only a System Administrator can grant permission to create a database. Database owners can assign command permissions to groups in databases they own.
------------	--

► **Note**

Permission to create a database applies only to users of the *master* database.

To manage command permissions:

1. Display the Group dialog box, as described in “Displaying Group Properties” on page 8-34.
2. Select Command Permissions. The dialog box changes to the Command Permissions tab.

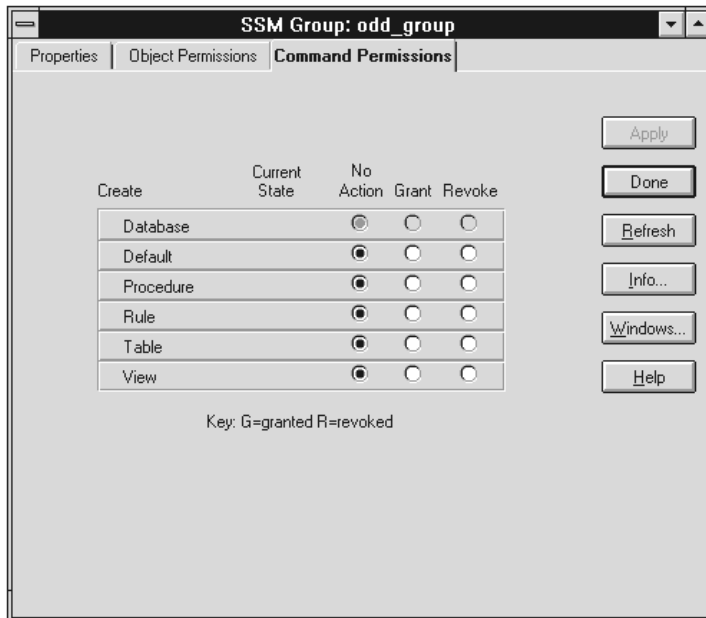


Figure 8-16: Group Command Permissions tab

3. To grant permission, select the option button in the Grant column corresponding to the permission.
4. To revoke permission, select the option button in the Revoke column corresponding to the permission.
5. Click Apply to change the permissions in the database. SQL Server Manager updates the Current State column to reflect the new level of permission:
 - G—granted
 - R—revoked
 - Blank—no permission explicitly assigned

► **Note**

SQL Server Manager reports on permissions explicitly granted or revoked. It does not show permissions that users obtain by being members in a group or by having an administrative role, such as System Administrator, associated with their logins.

Summary of Access Privileges and Permissions

The following table summarizes the SQL Server Manager protection system as implemented in SQL Server Manager. The type of user listed as the one to whom the task defaults is the lowest level of user to which the permission is automatically granted. This user can grant the permission to other users or revoke it from other users, if it is transferable.

Table 8-2: Privilege and permission summary

Task description	Defaults to					Can be granted or revoked	
	System Administrator	System Security Officer	Operator	Database owner	Object owner	Yes	No
Add an alias	•			•			•
Add columns to a table					•		•
Bind a default to a table column or user datatype					•		•
Bind a rule to a table column or user datatype					•		•
Change database device default status	•						•
Change database options	•			•			•
(1) Transferred with database ownership (2) Public can create temporary tables, no permission required (3) If a view, permission defaults to view owner (4) Defaults to stored procedure owner (5) All users can change their own passwords							

Table 8-2: Privilege and permission summary (continued)

Task description	Defaults to					Can be granted or revoked	
	System Administrator	System Security Officer	Operator	Database owner	Object owner	Yes	No
Change database ownership	•			•			•
Change database storage allocations	•					(1)	
Change group membership	•			•			•
Change login defaults or fullname	•						•
Change login password (5)	•	•					•
Create a named cache	•					•	
Create a database	•					•	
Create a database device	•						•
Create a default				•		•	
Create a dump device	•						•
Create a group	•			•			•
Create an index					•		•
Create a login		•					•
Create a procedure				•		•	
Create a remote server	•						•
Create a rule				•		•	
Create a segment	•			•			•
Create a table		(2)		•		• (2)	
Create a trigger					•		•
Create a user	•			•			•
(1) Transferred with database ownership (2) Public can create temporary tables, no permission required (3) If a view, permission defaults to view owner (4) Defaults to stored procedure owner (5) All users can change their own passwords							

Table 8-2: Privilege and permission summary (continued)

Task description	Defaults to					Can be granted or revoked	
	System Administrator	System Security Officer	Operator	Database owner	Object owner	Yes	No
Create a view				•		•	
Define remote login mappings		•					•
Delete an alias	•			•			•
Delete a database or dump device	•						•
Delete a database object					•(3)		•
Delete a group	•			•			•
Delete a login	•						•
Delete a remote server	•						•
Delete a segment	•						•
Delete a user	•			•			•
Delete a user datatype	•			•			•
Disable disk mirroring	•						•
Dump a database			•	•			•
Dump a transaction log			•	•			•
Enable disk mirroring	•						•
Execute dbcc commands				•			•
Execute a procedure					•(4)	•	
Extend a segment	•			•			•
Grant permission on a database object					•	•	
(1) Transferred with database ownership (2) Public can create temporary tables, no permission required (3) If a view, permission defaults to view owner (4) Defaults to stored procedure owner (5) All users can change their own passwords							

Table 8-2: Privilege and permission summary (continued)

Task description	Defaults to					Can be granted or revoked	
	System Administrator	System Security Officer	Operator	Database owner	Object owner	Yes	No
Grant permission to create a database object				•		•	
Grant roles to logins	•	•					•
Insert a row in a table					• (3)	•	
Issue a database checkpoint				•			•
Issue an update statistics command					•		•
Kill a user process	•						•
Load a database from a backup			•	•			•
Load a transaction log from a backup			•	•			•
Lock a login	•	•					•
Move the transaction log to a different device	•		•				•
Place new table or view allocations on a segment	•			•	•		•
Reset SQL Server configuration options	•						•
Re-enable disk mirroring on an inactive mirror device	•						•
Revoke permission on an object					•		•
(1) Transferred with database ownership (2) Public can create temporary tables, no permission required (3) If a view, permission defaults to view owner (4) Defaults to stored procedure owner (5) All users can change their own passwords							

Table 8-2: Privilege and permission summary (continued)

Task description	Defaults to					Can be granted or revoked	
	System Administrator	System Security Officer	Operator	Database owner	Object owner	Yes	No
Revoke permission to create an object				•			•
Select rows in a table					• (3)	•	
Shut down SQL Server	•						•
Unbind a default from a table column or user datatype					•		•
Unbind a rule from a table column or user datatype					•		•
Update rows in a table					• (3)	•	
(1) Transferred with database ownership (2) Public can create temporary tables, no permission required (3) If a view, permission defaults to view owner (4) Defaults to stored procedure owner (5) All users can change their own passwords							

Summary of Transferable Privileges and Permissions

The following table summarizes transferable access privileges, the administrative role or user who grants and revokes access, and the dialog box and tab to use for entering the information.

Table 8-3: Access granted through tabs and dialog boxes

Privilege	Grantor	Dialog box and tab where granted	
Change database storage allocations	System Administrator	Database (permission transfers with database ownership)	Properties
Create a database	System Administrator	Group	Command Permissions
		User	Command Permissions
Create a default	Database owner	Group	Command Permissions
		User	Command Permissions
Create a procedure	Database owner	Group	Command Permissions
		User	Command Permissions
Create a rule	Database owner	Group	Command Permissions
		User	Command Permissions
Create a table	Database owner	Group	Command Permissions
		User	Command Permissions
Create a view	Database owner	Group	Command Permissions
		User	Command Permissions
Delete a table or view	Object owner	Group	Object Permissions
		Table or View	Object Permissions
		User	Object Permissions
Define referential constraints on a table	Table owner	User	Object Permissions
		Group	Object Permissions
		Table	Object Permissions
Execute a procedure	Procedure owner	Group	Object Permissions
		Procedure	Object Permissions
		User	Object Permissions

Table 8-3: Access granted through tabs and dialog boxes (continued)

Privilege	Grantor	Dialog box and tab where granted	
Grant permission to transfer permission to other users	Any authorized grantor	Group	Object Permissions
		Table or View	Object Permissions
		User	Object Permissions
Insert rows in a table or view	Object owner	Group	Object Permissions
		Table or View	Object Permissions
		User	Object Permissions
Select rows in a table or view	Object owner	Group	Object Permissions
		Table or View	Object Permissions
		User	Object Permissions
Update rows in a table or view	Object owner	Group	Object Permissions
		User	Object Permissions
		Table or View	Object Permissions

9

Managing Database Objects

What's in This Chapter

This chapter describes the database objects and how to create, display, modify, navigate between, and delete them. Some procedures are identical regardless of the object involved; some procedures are unique to the object.

After an overview of procedures common to all objects, the chapter discusses the following database objects in detail:

- Defaults
- Indexes
- Procedures
- Rules
- Triggers
- User Datatypes
- Views

The following objects are covered in other chapters:

- Segments—Chapter 7, “Managing Databases and Segments”
- Tables—Chapter 10, “Managing User Tables”
- Users and groups—Chapter 8, “Controlling Access”

Operations Common to All Objects

For all types of database objects, use the same operations to create, delete, and display detailed information about them.

Creating an Object

Privileges	Only a database owner or a user or group with create privileges can create an object.
------------	---

To create an object:

1. Select the icon of the database where you want to create the object.
2. From the Database menu, choose Create.
3. From the cascading menu, choose the type of object you want to create. The Create dialog box for the object opens.

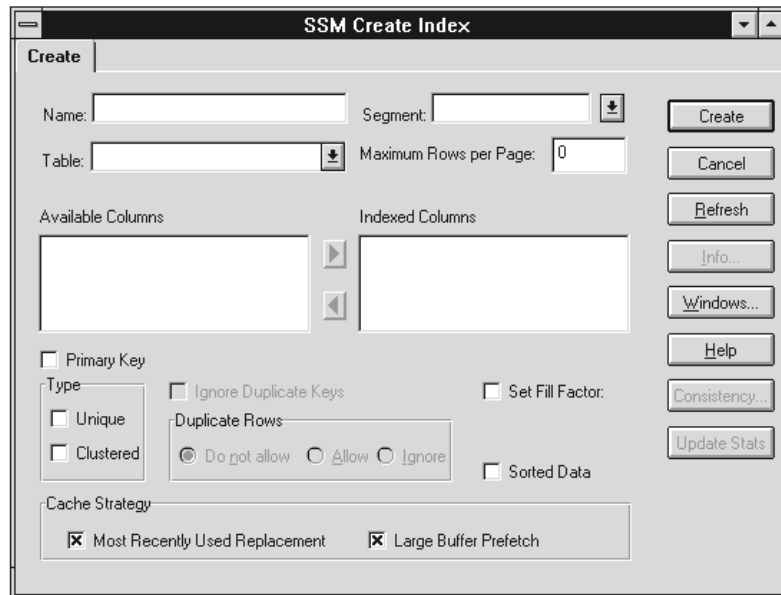
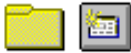




Figure 9-1: Create dialog box for an index

4. Define the object on the Create dialog box by filling in the appropriate attributes.
5. When you have finished defining the object, click Create to create the object. The Create dialog box is replaced by the object dialog box with the Properties tab displayed. In the place of the Create button is an Apply button so you can modify the object. Select the tabs appropriate to the object to further define bindings, permissions, and so on.

Shortcuts

	<p>Select the container icon for the type of object to create. Select the Create toolbar button.</p>
	<p>Click the right mouse button over the container icon for the object to create, and choose Create from the shortcut menu.</p>
	<p>Click the right mouse button over the database icon. Choose Create from the shortcut menu; then, choose the command for the object you want.</p>

Closing an Object's Dialog Box

When you have finished applying changes to an object by clicking the Apply button, click Done to close the dialog box. If you have made changes but not applied them, a confirmation dialog box opens asking if you want to apply the changes. Click Yes, No, or Cancel.

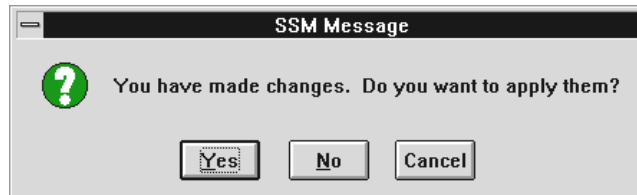


Figure 9-2: Confirmation dialog box

Displaying Details About an Object

To see detailed information about an object, use different methods depending on the information you want. You can display the following:

- The object's properties
- The SQL Server and database in which the object exists
- Information about objects that relate to the object

Displaying Object Properties

To see the properties of an object, including those that go into the SQL command to create the object's schema definition:

1. Select the icon of the object you want to see. The menu for the object appears in the menu bar.
2. From the context-sensitive menu, choose Properties... The dialog box opens with the Properties tab displayed.

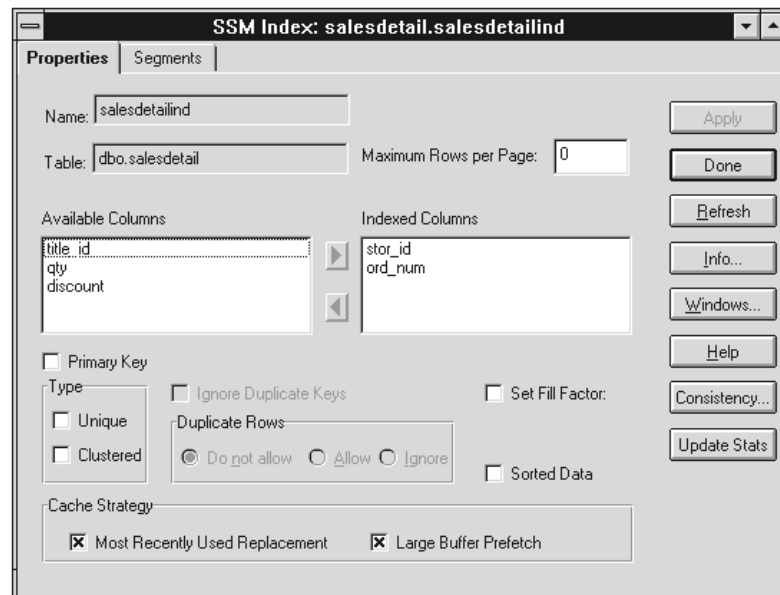


Figure 9-3: Properties tab for an index

Shortcuts

Double-click the object icon.



Select the object icon, and select the Properties toolbar button.



Click the right mouse button over the object icon; then, choose Properties... from the shortcut menu.



Displaying SQL Server and Database Objects

The Voyager window shows all objects in the SQL Server and database hierarchy in which they belong. In addition, the Contents bar above the right pane of the window displays the name of the SQL Server and database for the selected object. The following illustration of the Voyager window shows both of these display features.

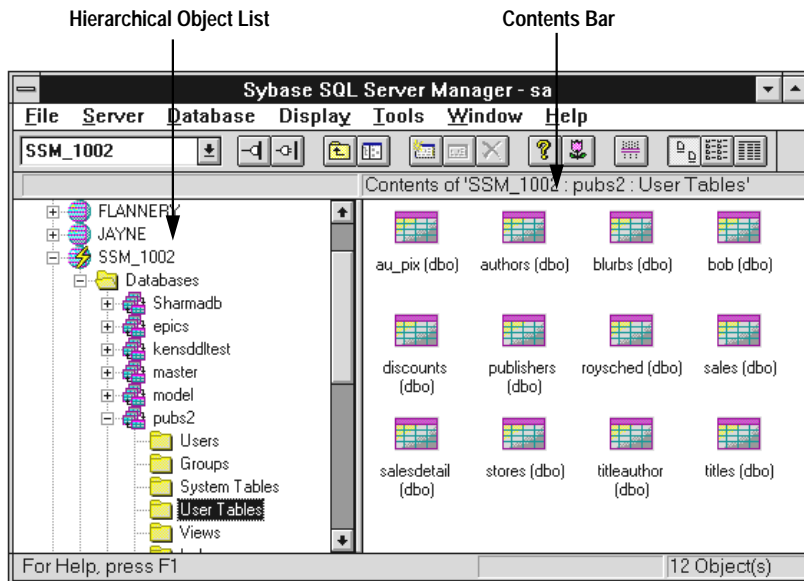


Figure 9-4: Voyager window showing object hierarchy

Displaying an Object Relative to Other Objects

To see information on objects that relate to the current object:

1. Open the object dialog box as described in “Displaying Object Properties.”
2. Select the tab button appropriate to the information you want. For example, the view dialog box has a tab button for Dependencies and Permissions.

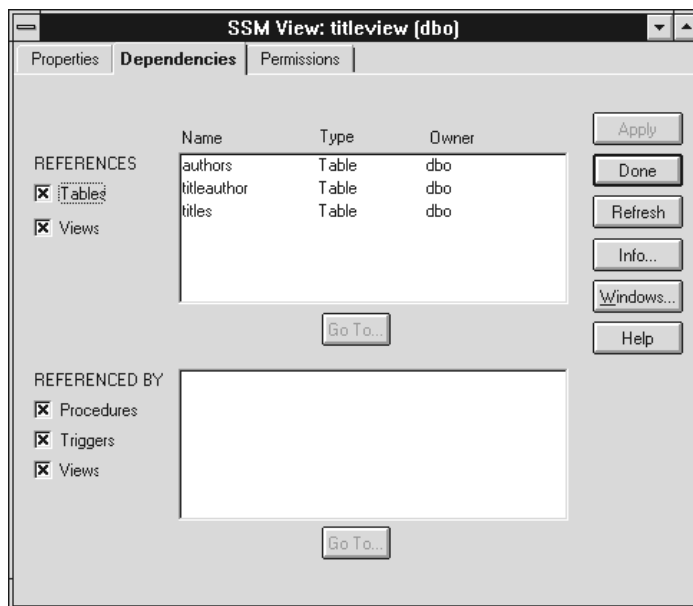


Figure 9-5: Dependencies tab for a view

Deleting an Object

To delete an object:

1. Select the icon of the object. Its context-specific menu appears in the menu bar.
2. Choose Delete... from the menu. A confirmation dialog box opens.



Figure 9-6: Delete confirmation dialog box

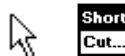
3. Click Yes to delete the object.

Shortcuts

Select the object; then, select the Delete toolbar button.



Click the right mouse button over the object icon; then, choose Delete from the shortcut menu.



Managing Defaults

A **default** is a value you specify that SQL Server inserts for a table column when no value is provided. SQL Server Manager supports the following ways of creating a default:

- As part of a table column definition. As you create or add columns to a table, you can specify default values that apply only to those columns.
- As a database object. Once you create a default, you can bind it to multiple table columns and to user datatypes.

The advantage of the first method is that you specify the table column and its default value in the same step. The advantage of the second is that you can define the default once and apply it as needed to multiple table columns. Either method is available whenever you create a table column.

This section describes the method in which you create a default as a database object. For information on creating a default as part of a table column definition, see “Creating a Table” on page 10-1 or “Adding Columns to a Table” on page 10-13.

This section covers the following:

- Creating a default
- Displaying default properties
- Binding and unbinding a default
- Displaying dependencies
- Navigating to objects with dependencies on a default

Creating a Default:

Privileges	Only a database owner or a user with create default permission can create a default.
------------	---

To create a default:

1. Select the icon of the database in which to create the default.
2. From the Database menu, choose Create; then, choose Default from the cascading menu. The Create Default dialog box opens.

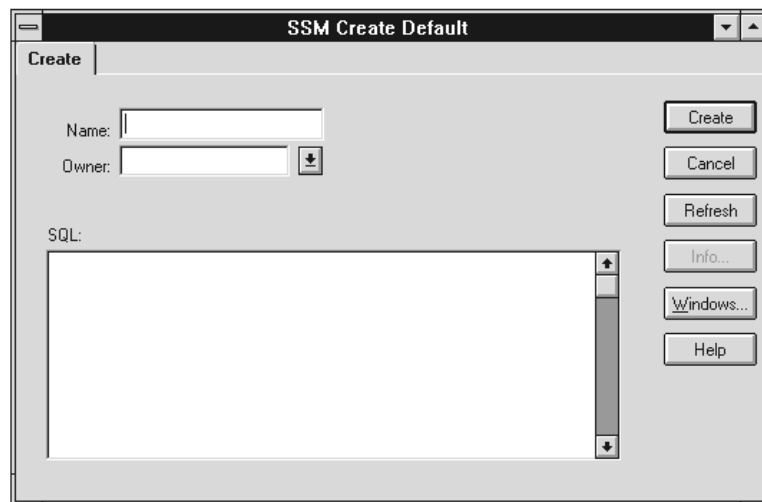





Figure 9-7: Create Default dialog box

3. In the Name edit box, enter a name for the default.
4. From the list in the Owner box, choose an owner for the default.
5. In the SQL box, enter an expression representing the default value. SQL Server Manager uses this expression to generate a create statement for the default in the correct Transact-SQL format.
6. Click Create.

Shortcuts

	<p>Select the Default container icon, and select the Create toolbar button.</p>
	<p>Click the right mouse button over the Default container icon; then, choose Create from the shortcut menu.</p>
	<p>Click the right mouse button over the appropriate database icon. Choose Create from the shortcut menu; then, choose Default.</p>

Example

You want to create a default, owned by user “virgil,” to bind to a column that holds a state abbreviation. The default, named “statedflt,” should have the value “MA.” In the Create Default dialog box, enter the values shown in the following figure:

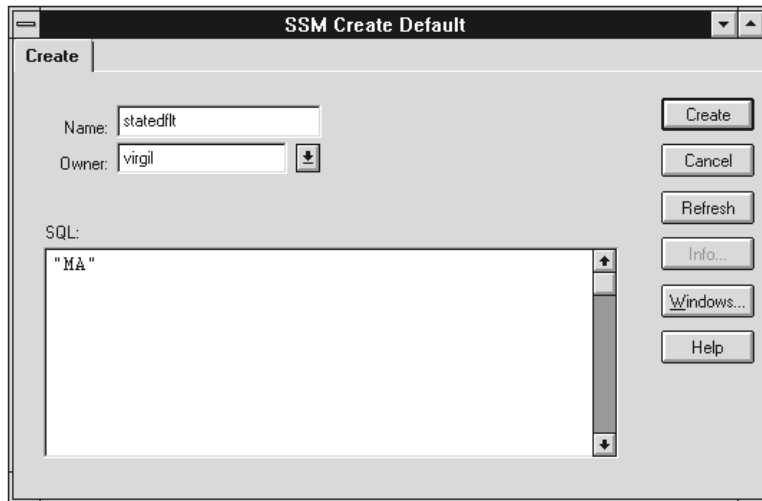


Figure 9-8: Creating the statedflt default

SQL Server Manager constructs the appropriate SQL query from the data you provide and creates the default shown in the following figure.

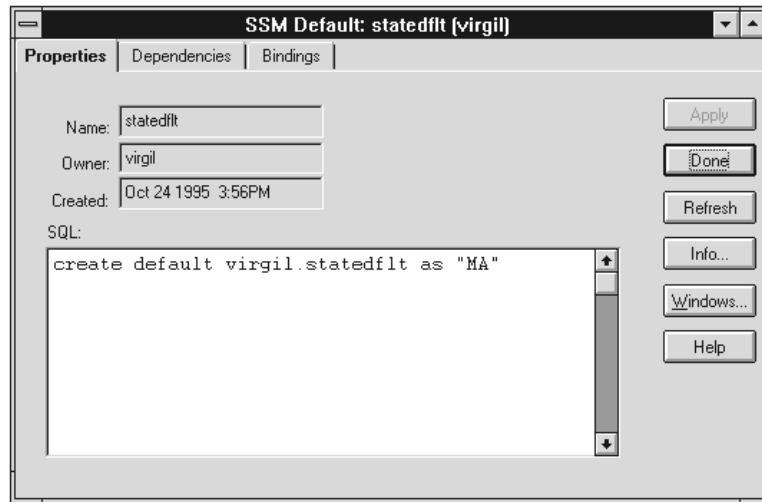


Figure 9-9: statedflt default as created by SQL Server Manager

Displaying Default Properties

To display the Properties tab:

1. Select the icon of the default.
2. From the Default menu, choose Properties...

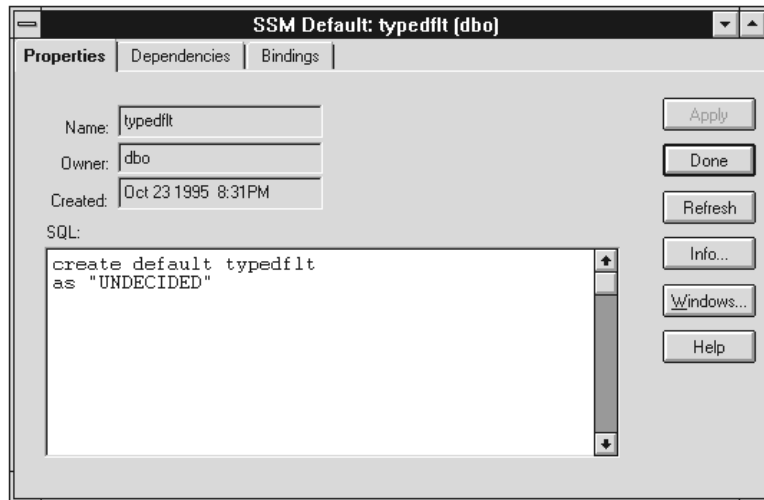


Figure 9-10: Default Properties tab

Shortcuts

Double-click the default icon.



Select the default icon, and select the Properties toolbar button.



Click the right mouse button over the default icon, and choose Properties... from the shortcut menu.



Displaying Default Dependencies

To display dependencies for a default:

1. Display the default Properties tab as described in “Displaying Default Properties” on page 9-12.
2. Click Dependencies. The display changes to the Dependencies tab.

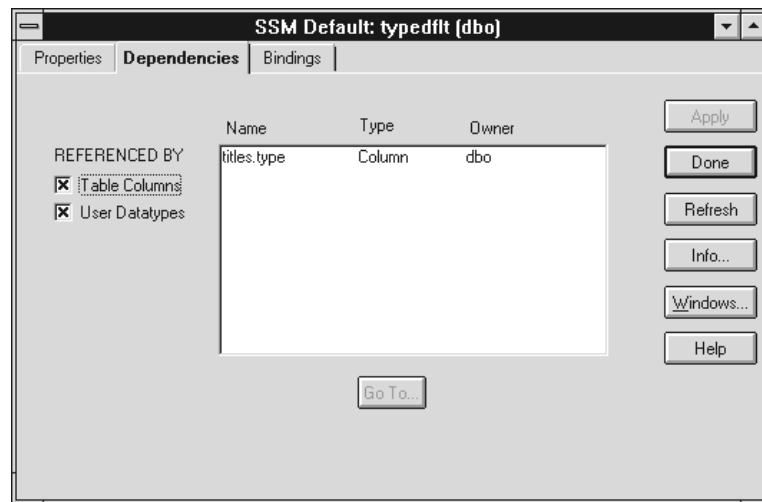


Figure 9-11: Default Dependencies tab

3. To display only columns that reference the default, clear User Datatypes. To display only user datatypes, clear Table Columns.

Navigating to Objects with Dependencies on a Default

To navigate to objects that reference a default:

1. Display the default Dependencies tab as described in “Displaying Default Properties” on page 9-12.
2. To display only table columns that reference the default, clear User Datatypes. To display only user datatypes, clear Table Columns.
3. To display the Properties tab for an object, select the object in the list. Then, click Go To...

Binding a Default

You can bind a default to one or more table columns, or to all columns having a given user datatype. The column has the default value if no other value is provided.

Before binding a default, verify that:

- The column is large enough for the default.
- The default value is allowed by any rules that are bound to the column or user datatype.
- Columns that do not allow null values have a default bound to them. Otherwise, SQL Server produces an error message whenever anyone inserts a row and fails to make an entry for that column.

► **Note**

A default bound to a column overrides a default bound to a user datatype.

Binding a Default to a Table Column

To bind a default to a table column:

1. Display the default Properties tab as described in “Displaying Default Properties” on page 9-12.
2. Click Bindings. The display changes to the Bindings tab.

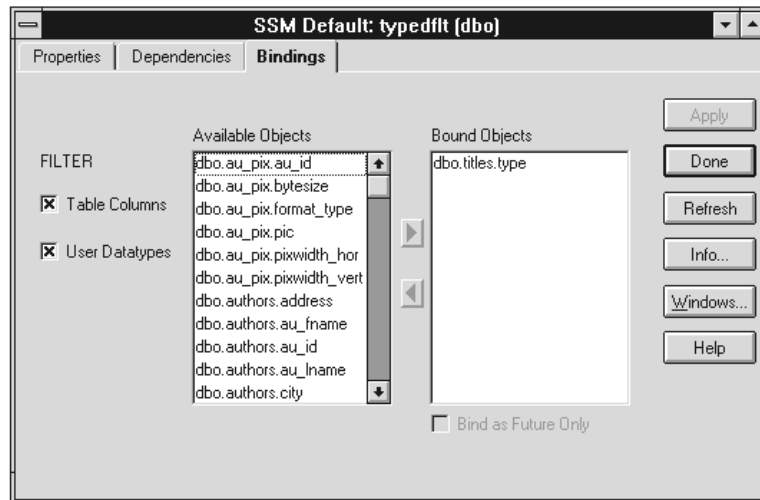


Figure 9-12: Default Bindings tab

3. To display only table columns, clear User Datatypes.
4. To bind objects to the default, move them from the Available Objects list to the Bound Objects list. To move an object, select the object in the list and click the right-pointing arrow.
5. Click Apply.

Shortcut

Drag and drop the default icon onto the table icon to display the Bindings tab.



Binding a Default to a User Datatype

To bind a default to a user datatype:

1. Display the default Properties tab as described in “Displaying Default Properties” on page 9-12.
2. Click Bindings. The display changes to the Bindings tab.
3. To display only user datatypes, clear Table Columns.

4. To bind objects to the default, move them from the Available Objects list to the Bound Objects list. To move an object, select the object in the list and click the right-pointing arrow.
5. To keep existing columns of the user datatype from inheriting the default, check Bind as Future Only. To let current columns of the user datatype inherit the default, leave the box blank.
6. Click Apply to set the bindings.

Shortcut



Drag and drop the default icon onto the user datatype icon to display the Bindings tab.

Deleting a Binding

To unbind a default:

1. Display the default Properties tab as described in “Displaying Default Properties” on page 9-12.
2. Click Bindings. The display changes to the Bindings tab.
3. To display only user datatypes, clear Table Columns. To display only columns, clear User Datatypes.
4. To unbind an object, move it from the Bound Objects list to the Available list. To move an object, select the object in the list and click the left-pointing arrow.
5. To keep existing columns of the user datatype from losing the default, select Bind as Future only. To unbind the default from the user datatype and from all columns of that type, leave the box unselected.
6. Click Apply to complete the deletion of the binding.

Shortcut



Drag and drop the default icon onto the table icon or user datatype icon to display the Bindings tab.

Managing Indexes

An **index** is created on one or more table columns and points to the place where the column data is stored on disk. Indexes speed up data retrieval and are useful for enforcing referential integrity. A table can have more than one index.

The index attributes you can define when you create an index with SQL Server Manager are:

- Clustered or nonclustered
- Unique or nonunique

A **clustered index** is sorted on an ongoing basis so that the physical order of rows is the same as the logical order. The bottom or leaf level of the index contains the actual data pages of the table. Finding data using a clustered index is faster than using a nonclustered index.

Create a clustered index before creating nonclustered indexes because nonclustered indexes are rebuilt when a clustered index is created. A table can have only one clustered index.

A **nonclustered index** stores key values and pointers to data pages rather than the data itself. The physical order of the rows is not the same as the indexed order. A table can have up to 249 nonclustered indexes.

A **unique index** is one in which no two rows can have the same index value. A unique index is useful as an integrity check on unique data.

A **nonunique index** is one in which two or more rows can have the same index value.

You can combine these characteristics; for example, you can create a unique, nonclustered index.

Once you create an index, you can change the segment on which it is allocated and you can check index consistency. When you modify the

index, the old index is deleted from the database and replaced with the new version.

This section describes the following:

- Creating an index
- Displaying index properties
- Placing an index on a segment
- Navigating to a segment
- Checking index consistency
- Updating statistics

Creating an Index

Privileges	Only a table owner or a user with create index permission on a table can create an index.
------------	--

To create a new index:

1. Select the icon of the database in which to create the index.
2. From the Database menu, choose Create; then, choose Index from the cascading menu. The Create Index dialog box opens.

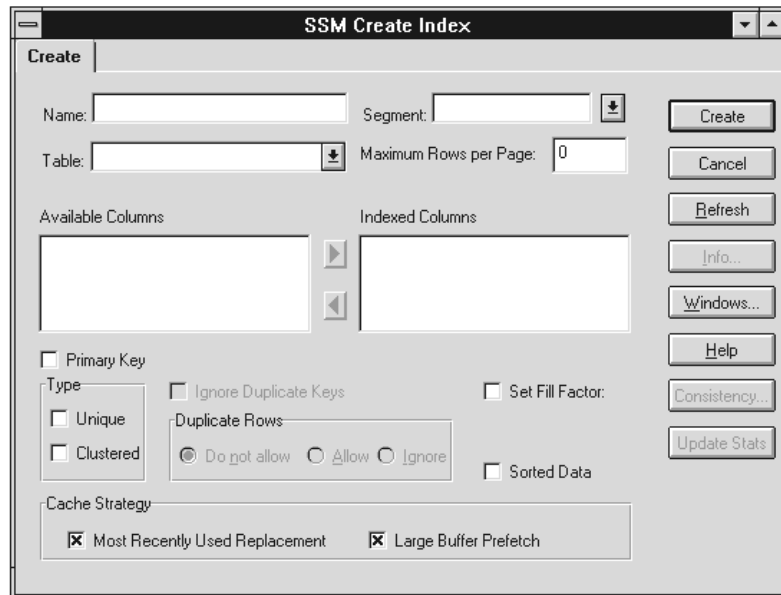



Figure 9-13: Create Index dialog box

3. In the Name box, enter a name for the index.
4. From the Segment box, choose a segment on which to create the index. A clustered index must be created on the same segment as its table; otherwise, the table moves to the new segment. To check the segment on which a table is placed:
 1. Display the Table Properties tab by selecting the table icon and choosing Properties... from the Table menu.
 2. To display the Segments tab, which shows the table's current disposition on a segment in the Current box, click Segments.
5. From the Table list, select the table on which to create the index.
6.  Optionally, to override the default assigned by SQL Server for the maximum number of index rows to store on a leaf level page, enter a number in the Maximum Rows Per Page box. This option is available in SQL Server release 11.0 or later and is mutually exclusive with the Fill Factor option. Also, unlike the Fill Factor option, you can change the Maximum Rows Per Page value at any time.

The default value, 0, causes SQL Server to create clustered indexes with full pages and nonclustered indexes with full leaf pages.

For a clustered index, values between 0 and 256 are valid. For a nonclustered index, the largest allowable Maximum Rows Per Page value depends on the size of the index key. To approximate the largest allowable value, subtract 32 from the page size and divide the resulting number by the index key size.

7. Specify the columns to index by moving them from the Available Columns list to the Indexed Columns list. To move a column to the Indexed Columns list, select the column in the Available Columns list and click the right-pointing arrow.
8. To create the index as a primary key, select the Primary Key box. SQL Server Manager automatically sets the controls that support primary key creation. Optionally, to specify that the index is clustered as well as unique, select Clustered. Skip to Step 12.
9. From the Type group box, select Unique or Clustered:
 - Unique—specifies that the index is unique and no two rows can have the same key (index) value. When this feature is checked, the system checks for duplicate key values when the index is created and each time data is added with an insert or update command. You cannot create a unique index on a column that includes duplicate values or more than one null value. This kind of index makes sense when the data itself is unique.
 - Clustered—specifies that the index is clustered and has data, rather than pointers, in its leaf pages. A table can have only one clustered index.
10. To specify how SQL Server should handle an attempt to insert a row with a duplicate key, use the Ignore Duplicate Keys box:
 - To specify that SQL Server should reject the row with the duplicate key, select Ignore Duplicate Keys.
 - To specify that SQL Server should allow the insertion of the row with the duplicate key, clear Ignore Duplicate Keys.

This option is available only on a unique, nonclustered index.
11. To specify how SQL Server should handle a transaction that would create a duplicate row on the table, select one of the following options in the Duplicate Rows box:
 - Do Not Allow—SQL Server rejects the update.
 - Allow—SQL Server allows the update.

- Ignore—SQL Server rejects the duplicate row and issues an error message but continues to process the transaction.

This option is available only on a nonunique, clustered index. You cannot create an index that has this option if a unique index already exists on any column in the table.

◆ **WARNING!**

Use the Duplicate Keys and Duplicate Rows features with caution. An update that creates a duplicate key row or a duplicate row can result in data being overwritten.

12. To specify how full each index page should become, select the Set Fill Factor box. Then, in the edit box below the Set Fill Factor box, enter a percentage. The default fill factor is determined by the value of the fill factor SQL Server configuration parameter. There is seldom a reason to change the fill factor parameter.


If the default value is 0, you can enter a percentage from 0 to 100. If the default value is other than 0, only percentages from 1 to 100 are valid. A fill factor of 0 does not mean the pages are empty, but means there is some room for additional data. Use a fill factor of 100 only for read-only tables to which no additional data will **ever** be added. Small fill factors cause each index to take more storage space and result in index pages that are not completely full.

► **Note**

The fill factor percentage is relevant only at the time an index is created. As the data changes, the pages are not maintained at any particular level of fullness.

13. To indicate that data has been presorted for faster index building, check Sorted Data. This option works only if the table data has been loaded in presorted format with the bulk copy utility **and** the index is one of the following:

- Clustered
- Unique, nonclustered
- Nonunique, nonclustered, and there are no duplicate keys

14.  If you are connected to a release 11.0 SQL Server, you can select a cache strategy for the index. This selection overrides the

default strategy, determined by the SQL Server optimizer, for reading data pages from an index into the buffers in data cache. The following selections are available:

- Most Recently Used Replacement (MRU)—This selection specifies that SQL Server uses the most recently used strategy for determining where in cache to place data pages when reading in new data.

If you clear the check box, SQL Server reads new pages into the MRU end of the chain of buffers in cache. Subsequent reads move the pages along the chain towards the least recently used (LRU) end until they are flushed out by new reads at the MRU end. If you select Most Recently Used Replacement, SQL Server reads new pages into the LRU end. They are used and immediately flushed when a new page enters the MRU end.

This strategy is advantageous when a page is needed only once for a query. It tends to keep such pages from flushing out other pages that can potentially be reused while still in cache.

- Large Buffer Prefetch—This selection applies if one or more large buffer pools is defined in the default cache or, if the index is bound to a named cache, in the named cache. A large buffer pool is one that has buffers larger than the 2K default, as specified in the Cache Properties dialog box. If you select Large Buffer Prefetch, the SQL Server optimizer can fetch data in I/Os of as many as eight 2K data pages at a time instead of the default of one page at a time.

This strategy is advantageous for data that is stored and accessed sequentially; for example, it can improve performance for queries that scan the leaf level of a nonclustered index.

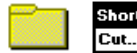
15. Click Create.

Shortcuts

Select the Index container icon, and select the Create toolbar button.



Click the right mouse button over the Index container icon; then, choose Create from the shortcut menu.



Click the right mouse button over the Database icon; then, choose Create from the shortcut menu and Index from the cascading menu.



Displaying Index Properties

To display index properties:

1. Select the icon of the index to display.
2. From the Index menu, choose Properties...

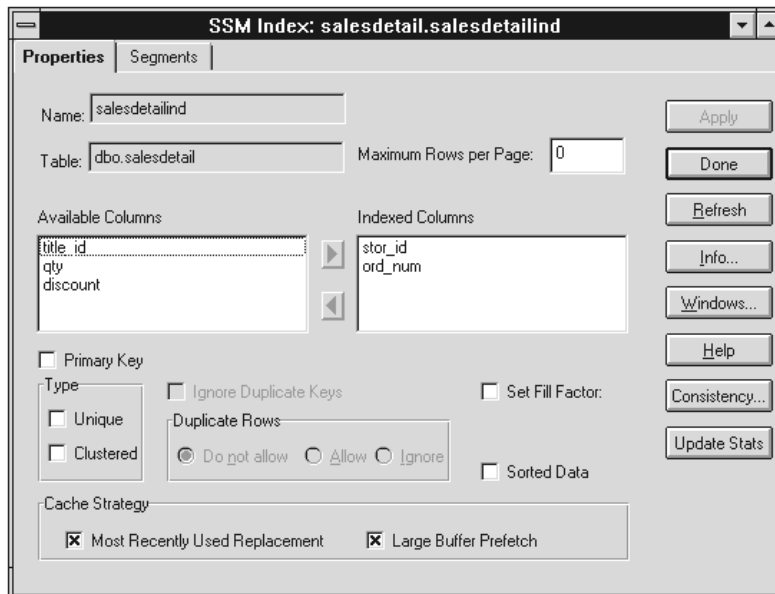


Figure 9-14: Index Properties tab

Shortcuts

Double-click the index icon.



Select the index icon, and select the Properties toolbar button.



Click the right mouse button over the index icon, and choose Properties... from the shortcut menu.



Placing an Index on a Segment

Using a segment to put an index on a specific database device can improve SQL Server performance and can give increased control over placement, size, and space usage of database objects. For example:

- If you put a table on one device, and its nonclustered indexes on a device on another disk controller, the time required to read or write to the disk can be reduced.
- If you put tables and indexes only on specific segments, those objects cannot grow beyond the space available on the devices represented by the segments.
- If you put tables and indexes in their own segments that map to reserved devices, other objects cannot contend for space with them.

To place an index on a segment:

1. Display the Index Properties tab, as described in “Displaying Index Properties” on page 9-23.
2. Click Segments. The display changes to the Segments tab.

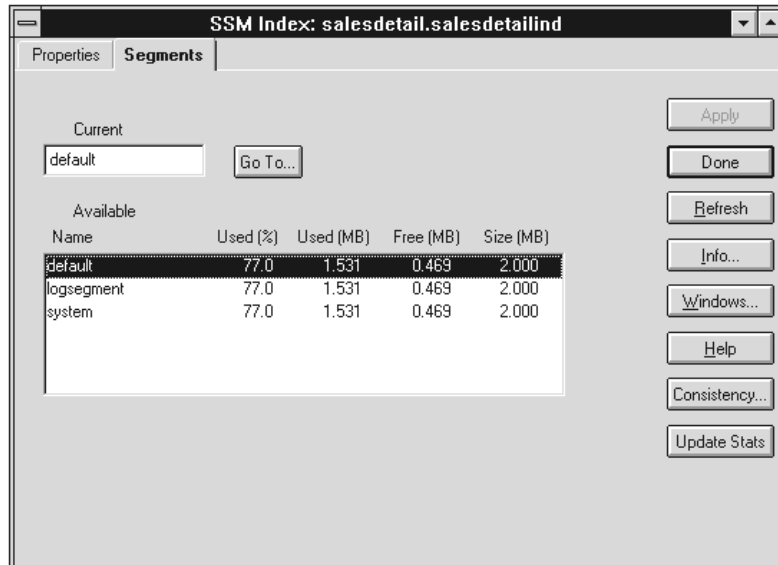
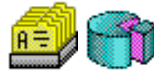


Figure 9-15: Index Segments tab

3. In the Current box, enter the name of a segment from the Available list of all the segments in the database. The value you enter is the segment on which the index puts new data.
4. Click Apply to select the segment. Future growth of the index now occurs on this segment.

Shortcut



Drag and drop the index icon onto the segment icon to display the Segments tab.

Navigating to a Segment

To navigate to a segment from an index:

1. Select Segments on the Index dialog box. The dialog box displays the Segments tab.
2. Select a segment in the Name list, then click Go To... to display the Properties tab for the segment.

Checking Index Consistency

The Database Consistency Checker (dbcc) is a set of utility commands for checking the logical and physical consistency of a database. The dbcc command has an `indexalloc` option to check allocation structures for an index. You can check the current index to see that all data pages are correctly allocated, that no page is allocated that is not used, and that no page is used that is not allocated.

This section gives general instructions for executing the dbcc command for index consistency and then describes each command option.

Privileges	Only a database owner can check index consistency.
------------	--

Instructions

To check index consistency:

1. Select the icon of the index to check.
2. From the Index menu, choose Consistency... The Index Consistency Check dialog box opens.

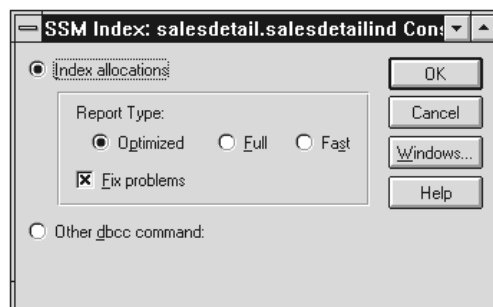


Figure 9-16: Index Consistency Check dialog box

3. Select the appropriate option. For details on available options, see “Consistency Check Options” on page 9-27.
4. Click OK to check index consistency. SQL Server Manager issues the appropriate `dbcc` command and displays the resulting report in the Output dialog box. You can copy and paste report output into a word processor or text editor.

Shortcuts

From the Index dialog box, click Consistency...



Click the right mouse button over the index icon; then, choose Consistency... from the shortcut menu.



Consistency Check Options

The following `dbcc` command options are available at the index level:

- Index Allocations
- Other `dbcc` Command

Index Allocations Option

This option corresponds to the `dbcc indexalloc` command. The following report types are available:

- To report allocation pages listed in the Object Allocation Map pages for the index, select Optimized. (For information on the Object Allocation Map, see the *System Administration Guide*.)
- To report all types of allocation errors, select Full.
- To report pages that are referenced but not allocated in the extent, select Fast.

By default, `dbcc indexalloc` fixes allocation errors as it executes. SQL Server Manager automatically places the database in single-user mode while executing `dbcc indexalloc` and then returns the database to

multiple-user mode when processing is complete. To generate a report without fixing allocation errors, clear the Fix Problems box.

```

SSM Output
INDID=1  FIRST=1      ROOT=8  SORT=0
      Data level: 1.  1 Data Pages in 1 extents.
      Indid       : 1.  1 Index Pages in 1 extents.
TOTAL # of extents = 2
Alloc page 0 (# of extent=1 used pages=2 ref pages=2)
Alloc page 0 (# of extent=1 used pages=3 ref pages=3)
Total (# of extent=2 used pages=5 ref pages=5) in this database
DBCC execution completed. If DBCC printed error messages, contact a user with
System Administrator (SA) role.
Database option 'single user' turned OFF for database 'epics'.
Run the CHECKPOINT command in the database that was changed.
  
```

Figure 9-17: Output from Full option of index consistency check

Other dbcc Command Option

This option allows you to enter the text of other `dbcc` command options. When you select Other `dbcc` Command, an edit box appears below the option button. Enter the text of the entire `dbcc` command option to execute, including the `dbcc` keyword.

Updating Statistics

When you update statistics, SQL Server updates information about the distribution of key values in indexes associated with a table. Update statistics if a great deal of data in an indexed column has been added, changed, or removed. Update statistics after adding new rows to a table whose rows have been deleted with `truncate table`.

Privileges	Only a table owner or a database owner can update index statistics.
------------	---

To update statistics:

1. Select the icon of the index to update.
2. From the Index menu, choose Update Statistics to run `update statistics`.

Shortcuts

In the Index dialog, click the Update Stats button.



Click the right mouse button over the Index icon; then, choose Update Statistics from the shortcut menu.



► Note

After running **update statistics**, recompile the table for which the index is defined. This ensures that the procedures and triggers that use the index will use the new key distribution.

Managing Procedures

A **procedure** is a named collection of SQL statements and flow control statements. Once you create a procedure, you can modify the permission to access it. You can also display the procedure's dependencies and navigate to them.

This section describes the following:

- Creating a procedure
- Displaying procedure properties
- Navigating to objects with dependencies on a procedure
- Updating user and group permissions on a procedure
- Navigating to users and groups

Creating a Procedure

A procedure that performs a select, execute, or data modification command must be owned by the same user as the object acted upon. To create a procedure:

Privileges	Only a database owner or a user or group with create procedure permission can create a procedure.
------------	--

1. Select the icon of the database in which to create the procedure.
2. From the Database menu, choose Create; then, choose Procedure from the cascading menu. The Create Procedure dialog box opens.

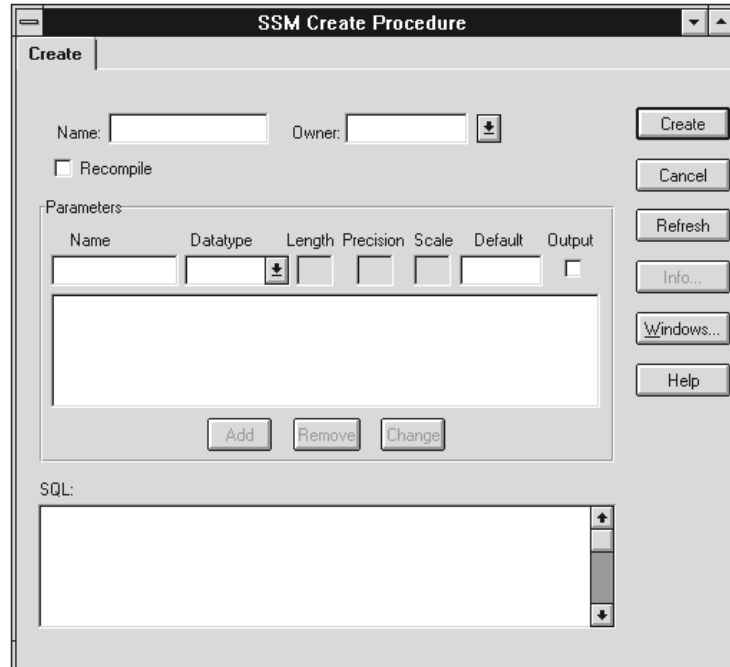
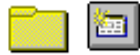


Figure 9-18: Create Procedure dialog box

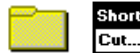
3. In the Name edit box, enter a name for the procedure.
4. Select an owner from the Owner list.
5. To create a new query plan for the procedure each time it is executed, check Recompile. Use this option if you expect that the execution of the procedure may be different each time; for example, if the data passed in its parameters changes so much that a query plan produced at execution would differ greatly from a plan that is stored.
6. Define parameters, if necessary. Parameters are named arguments to the procedure. The value of each parameter is supplied when SQL Server executes the procedure. Parameters are optional in create procedure statements. To define a parameter:

- Enter the parameter name in the Name edit box. The name must begin with the @ symbol and can be a maximum of 30 characters long. If the name of a parameter contains nonalphanumeric characters, it must be enclosed in quotes.
 - Choose the datatype from the Datatype list.
 - Enter the length of the datatype in the Length edit box. This is required if the datatype is *char*, *nchar*, *varchar*, *nvarchar*, *binary*, or *varbinary*.
 - Enter the precision in the Precision edit box. If datatype is *decimal* or *numeric*, enter the maximum number of decimal digits that can be stored in the column. Specify a precision of 1 to 38. If datatype is *float*, enter the number of significant binary digits in the mantissa.
 - If datatype is *decimal* or *numeric*, enter the scale in the Scale edit box. Scale is the maximum number of digits that can be stored to the right of the decimal point. The scale must be less than or equal to the precision. Specify a scale of 0 to 38.
 - Enter an optional default value for the parameter in the Default edit box.
 - To return the parameter value to the execute command, check Output.
7. Click Add to add the parameter. Click Remove or Change if you change your mind **before** you create the procedure. Remove deletes the selected parameter. Change updates the values in the selected parameter with the current values in the data entry fields.
 8. Enter the actions the procedure is to take in the SQL edit box. These are statements that come after the *as* keyword in the create procedure statement. They can include flow control language and any SQL statements except *create view*, *create default*, *create rule*, *create procedure*, *create trigger*, or *use*.
 9. Click Create.

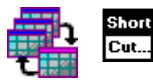
Shortcuts



Select the Procedure container icon in the appropriate database, and select the Create toolbar button.



Click the right mouse button over the Procedure container icon in the appropriate database. Choose Create from the shortcut menu.



Click the right mouse button over the appropriate database icon. Choose Create from the shortcut menu; then, choose Procedure.

Displaying Procedure Properties

To display the Properties tab:

1. Select the procedure icon.
2. From the Procedure menu, choose Properties...

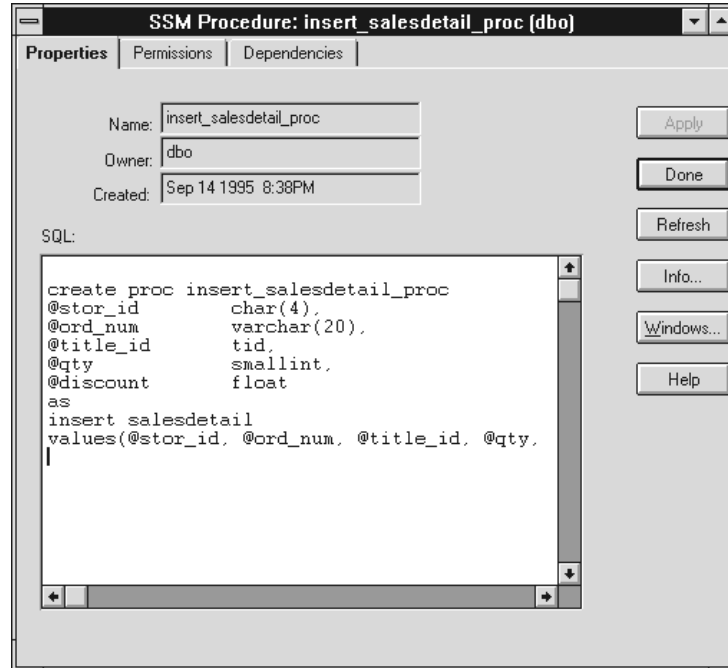








Figure 9-19: Procedure Properties tab

Shortcuts

		Double-click the procedure icon.
		Select the procedure icon, and select the Properties toolbar button.
		Click the right mouse button over the procedure icon, and choose Properties... from the shortcut menu.

Displaying Procedure Dependencies

To display dependencies for a procedure:

1. Display the Procedure Properties tab as described in “Displaying Procedure Properties” on page 9-32.
2. Click Dependencies. The display changes to the Dependencies tab.

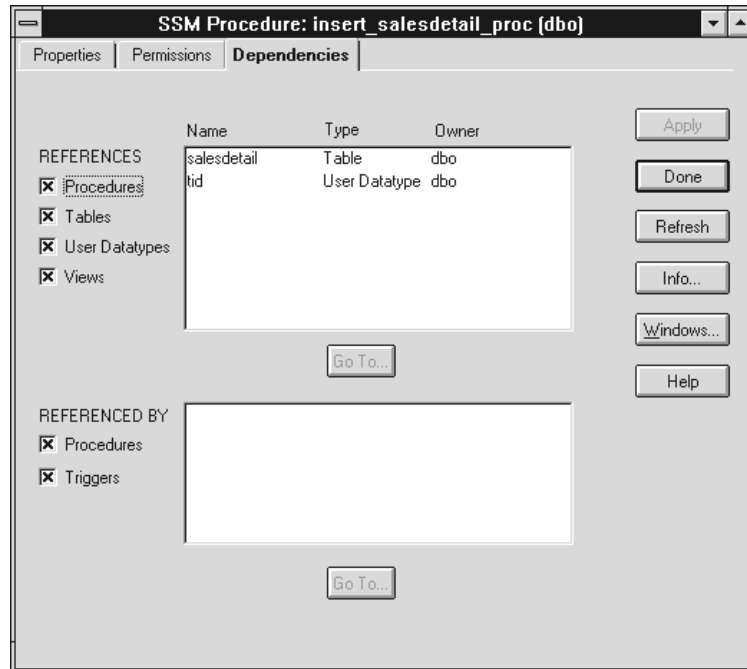


Figure 9-20: Procedure Dependencies tab

3. To display only specific types of objects that the procedure references, clear Procedures, Tables, Views, or User Datatypes.
4. To display only specific types of objects that reference the procedure, clear Procedures or Triggers.

Navigating to Objects with Dependencies on a Procedure

To navigate to objects that reference a procedure or that the procedure references:

1. Display the Procedure Properties tab as described in “Displaying Procedure Properties” on page 9-32.
2. Click Dependencies to display the Dependencies tab.
3. To display only specific types of objects that the procedure references, clear Procedures, Tables, Views, or User Datatypes.
4. To display only specific types of objects that reference the procedure, clear Procedures or Triggers.

5. To display the Properties tab for an object, select the object in the list and click Go To...

Updating User and Group Procedure Permissions

SQL Server Manager enables you to perform the following permission updates for procedures:

- Grant and revoke `execute` permission on a procedure.
- Grant permission to specific users or groups, or grant it using the `with grant` option so the recipient can also grant the permission to other users.
- Revoke the permission from specific users or groups, or revoke it using the `with cascade` option to revoke it from the named user and all users who acquired permission from the named user, directly or indirectly.

Privileges	A procedure owner can grant and revoke <code>execute</code> permission on the procedure.
------------	--

To update procedure permissions:

1. Display the Procedure Properties tab, as described in “Displaying Procedure Properties” on page 9-32.
2. Click Permissions. The display changes to the Permissions tab.

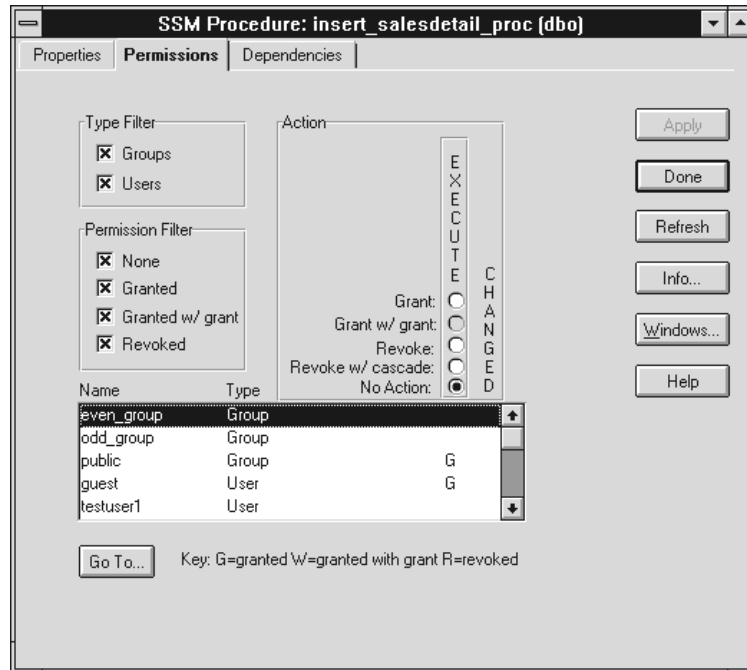
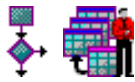


Figure 9-21: Procedure Permissions tab

3. To display only users, clear Groups. To display only groups, clear Users.
4. Initially, the dialog box shows all users or groups and the current state of their permissions. To limit the display, clear the appropriate boxes:
 - None—no permission has been granted on the procedure to the user or group.
 - Granted—permission has been granted on the procedure to the user or group.
 - Grant w/Grant—permission has been granted on the procedure with grant option. The grant option is available only to users, not to groups.

- Revoked—permission on the procedure has been revoked from the user or group.
5. Select the user or group whose permissions you want to change.
 6. To grant execute permission, select the Grant option button in the Execute column of the Action group box. To grant execute permission with the grant option, select the Grant w/Grant option button.
 7. To revoke execute permission, select the Revoke option button in the Execute column of the Action group box. To revoke execute permission with the cascade option, select the Revoke w/Cascade option button.
 8. Repeat Steps 3 through 7 for each user or group.
 9. Click Apply.

Shortcuts



To display the Procedure Permissions tab for users, drag and drop the procedure icon on a user icon.



To display the Procedure Permissions tab for groups, drag and drop the procedure icon on a group icon.

Navigating to Users and Groups

To navigate to users and groups from a procedure:

1. Display the Procedure Properties tab, as described in “Displaying Procedure Properties” on page 9-32.
2. Click Permissions on the Procedure dialog box to display the Permissions tab.
3. To display only users, clear Groups. To display only groups, clear Users.
4. Initially, the dialog box shows all users or groups and the current state of their permissions. To limit the display, clear the appropriate boxes:
 - None—no permission has been granted on the procedure to the user or group.

- Granted—permission has been granted on the procedure to the user or group.
 - Grant w/Grant—permission has been granted on the procedure with grant option. The grant option is available only to users, not to groups.
 - Revoked—permission on the procedure has been revoked from the user or group.
5. To display the Properties tab for a user or group, select the user or group in the list and click Go To...

Managing Rules

A **rule** defines the acceptable values for a table column or a user datatype. For example, the *pubs2* sample database rule, *title_idrule*, allows a column to contain the value “BU” followed by any four digits. Once you create a rule, you can bind it to table columns and to user datatypes.

SQL Server Manager supports the following ways of creating a rule:

- As part of a table definition. As you create or add columns to a table, you can specify rules that apply to all table columns. A rule created in this way is called a **table check constraint**.
- As part of a table column definition. As you create or add columns to a table, you can specify rules that apply only to those columns. A rule created in this way is called a **column check constraint**.
- As a database object. Once you create a rule, you can bind it to multiple table columns and to user datatypes.

The advantage of the first two methods is that you specify the table or column and its rule in the same step. The advantage of the third is that you can define the rule once and apply it as needed to multiple table columns. All methods are available whenever you create a table column.

This section describes the method in which you create a rule as a database object. For information on creating a rule as part of a table or table column definition, see “Creating a Table” on page 10-1 or “Adding Columns to a Table” on page 10-13.

This section describes the following:

- Creating a rule
- Displaying rule properties
- Binding a rule to a column
- Binding a rule to a user datatype
- Deleting a binding
- Displaying dependencies
- Navigating to objects with dependencies on a rule

Creating a Rule

Privileges	Only a database owner or a user or group with create rule permission can create a rule.
------------	--

To create a rule:

1. Select the icon of the database in which to create the rule.
2. From the Database menu, choose Create; then, choose Rule from the cascading menu. The Create Rule dialog box opens.

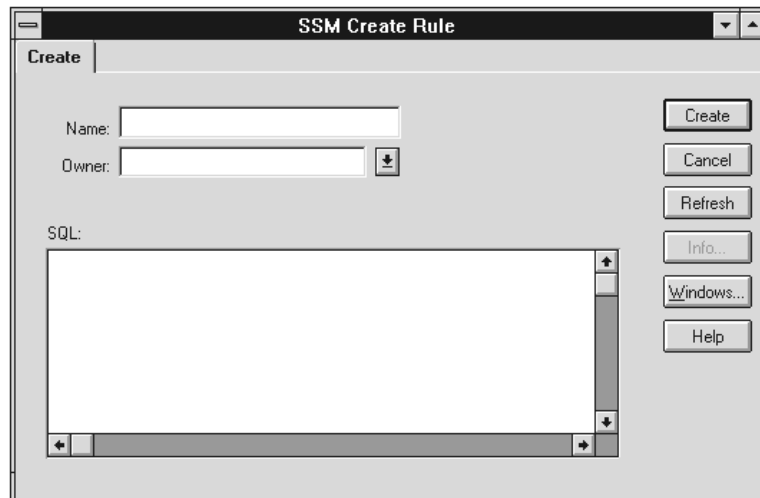


Figure 9-22: Create Rule dialog box

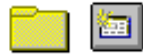
3. Enter a name for the rule in the Name edit box.

4. Choose an owner from the Owner list.
5. Enter the rule in the SQL edit box. Enter the part of the rule that follows the `as` keyword in the `create rule` statement; SQL Server Manager builds the complete statement from the values entered in the dialog box.

The expression portion can be any expression valid in a `where` clause, and can include arithmetic operators and relational operators. The expression takes one argument, prefixed by the `@` symbol, and refers to the value that is entered via the `update` or `insert` command. The expression cannot reference any column or other database object.

6. Click Create.

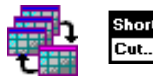
Shortcuts



Select the Rules container icon in the appropriate database, and select the Create toolbar button.



Click the right mouse button over the Rules container icon in the appropriate database; then, choose Create from the shortcut menu.



Click the right mouse button over the appropriate database icon. Choose Create from the shortcut menu; then, choose Rule.

Displaying Rule Properties

To display the Properties tab:

1. Select the icon of the rule for which to display properties.
2. From the Rule menu, choose Properties...

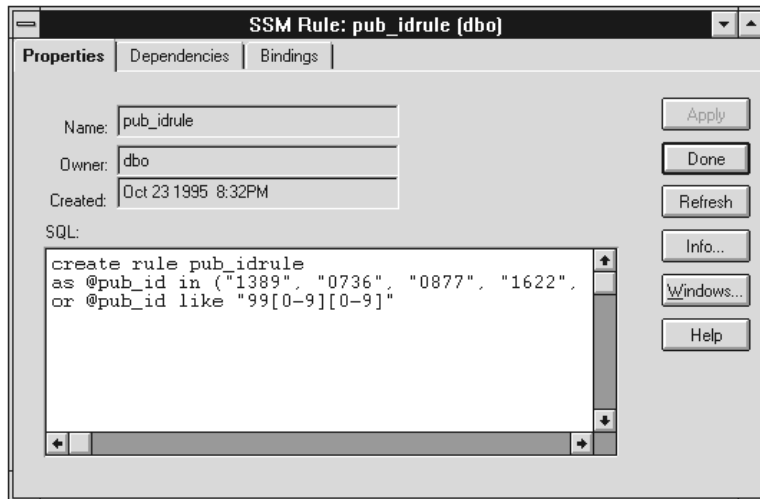





Figure 9-23: Rule Properties tab

Shortcuts

	Double-click the rule icon.
	Select the rule icon, and select the Properties toolbar button.
	Click the right mouse button over the rule icon. Choose Properties... from the shortcut menu.

Displaying Rule Dependencies

To display dependencies for a rule:

1. Display the Rule Properties tab, as described in “Displaying Rule Properties” on page 9-42.
2. Click Dependencies. The display changes to the Dependencies tab.

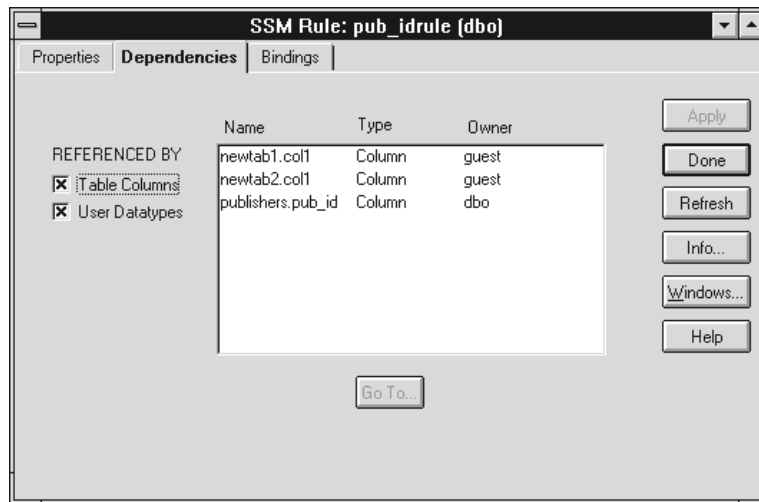


Figure 9-24: Rule Dependencies tab

3. To display only table columns that reference the rule, clear User Datatypes. To display only user datatypes that reference the rule, clear Table Columns.

Navigating to Objects with Dependencies on a Rule

To navigate to objects that reference a rule:

1. Display the Rule Properties tab, as described in “Displaying Rule Properties” on page 9-42.
2. Click Dependencies. The display changes to the Dependencies tab.
3. To display only table columns that reference the rule, clear User Datatypes. To display only user datatypes that reference the rule, clear Table Columns.

4. To display the Properties tab for a table or user datatype, select the object in the list and click Go To...

Binding a Rule

Before binding a rule to a table column or user datatype, consider the following:

- When you bind a rule to a table column or user datatype that already has a rule bound to it, the new rule takes effect and the old one is unbound.
- Rules bound to columns take precedence over rules bound to user datatypes.
- Rules must be compatible with the datatype of the column.
- Rules do not apply to data already in the database when they are created.
- Rules do not override column definitions.
- Make sure that any default value bound to the column or user datatype is compatible with the rule. A default that conflicts with the rule is not inserted.
- You cannot bind a rule to a *text*, *image*, or *timestamp* column.
- Do not assign a value to a user datatype variable that would be rejected by a rule bound to a column of the same datatype.

Binding a Rule to a Table Column

To bind a rule to a table column:

1. Display the Rule Properties tab, as described in “Displaying Rule Properties” on page 9-42.
2. Click Bindings. The display changes to the Bindings tab.

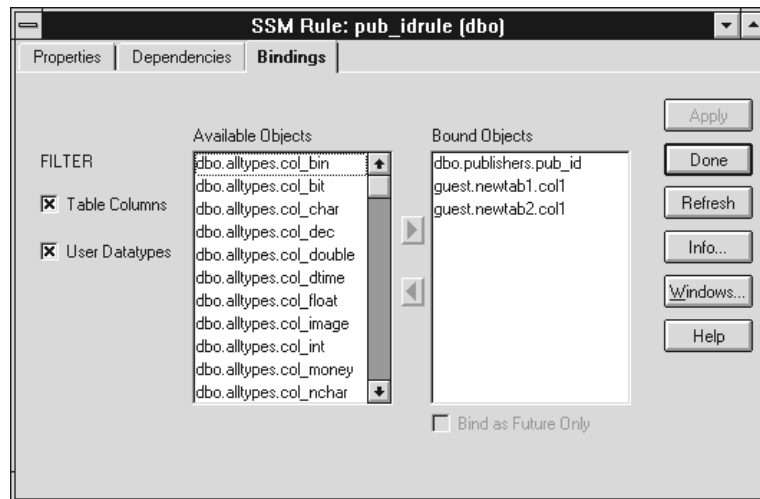


Figure 9-25: Rule Bindings tab

3. To display only table columns, clear User Datatypes.
4. To bind a column to the rule, move the column from the Available list to the Bound Objects list. To move the column, select the column in the list and click the right-pointing arrow.
5. Click Apply.

Shortcut



To display the Bindings tab, drag and drop the rule icon on the table icon.

Binding a Rule to a User Datatype

To bind a rule to a user datatype:

1. Display the Rule Properties tab, as described in “Displaying Rule Properties” on page 9-42.
2. Click Bindings. The display changes to the Bindings tab.
3. To display only user datatypes, clear Table Columns.
4. To bind a datatype to the rule, move it from the Available list to the Bound Objects list. To move the datatype, select the datatype in the list and click the right-pointing arrow.
5. Check Bind as Future Only if the binding affects only future use of the datatype. (SQL Server checks future inserts and updates, not current values, against the rule.) If the binding affects current use, leave Bind as Future Only cleared.
6. Click Apply.

Shortcut



To display the Bindings tab, drag and drop the rule icon on the user datatype icon.

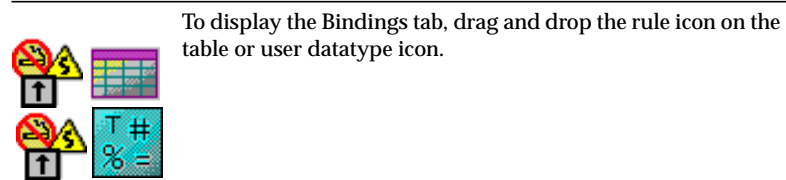
Deleting a Binding

Before creating a new rule with the same name as an existing rule, you must delete the existing rule. Before deleting a rule, you must unbind it from any table columns or user datatypes to which it is bound.

To unbind a rule:

1. Display the Rule Properties tab, as described in “Displaying Rule Properties” on page 9-42.
2. Click Bindings. The display changes to the Bindings tab.
3. To display only table columns, clear User Datatypes. To display only user datatypes, clear Table Columns.
4. To unbind an object, move it from the Bound Objects list to the Available list. To move the object, select the object in the list and click the left-pointing arrow.
5. To keep existing table columns of the specified user datatype from losing the rule, select Bind as Future Only. To unbind the rule from the specified user datatype and from all columns of that type, leave Bind as Future Only cleared.
6. Click Apply.

Shortcut



Managing Triggers

A **trigger** is a special type of procedure attached to a table column. The trigger goes into effect when a user changes the table with an insert, update, or delete command. The trigger executes immediately after the data modification statements are completed. Because triggers are more versatile than rules and referential integrity constraints, they are often used to do the following:

- Enforce referential integrity.
- Cascade changes through related tables in the database, such as deleting a record.
- Enforce restrictions much more complex than those defined with rules. Unlike rules, triggers can reference columns or database objects.

- Perform “what if” analyses, such as comparing a table before and after a data modification, and performing an action based on the results of the comparison.

Only the owner of an object can create a trigger on it. However, a trigger can modify an object owned by another user. If a trigger activates a trigger on another object, the owner of the first trigger must have privileges on the target object.

This section describes the following:

- Creating a trigger
- Displaying trigger properties
- Displaying dependencies
- Navigating to objects with dependencies on a trigger

Creating a Trigger

When you create a trigger, you specify the table affected and the command (*insert*, *update*, or *delete*) that activates the trigger. You also specify the action the trigger is to take.

Privileges	Only a table owner or database owner can create a trigger.
------------	--

To create a trigger:

1. Select the icon of the database in which to create the trigger.
2. From the Database menu, choose Create; then, choose Trigger from the cascading menu. The Create Trigger dialog box opens.

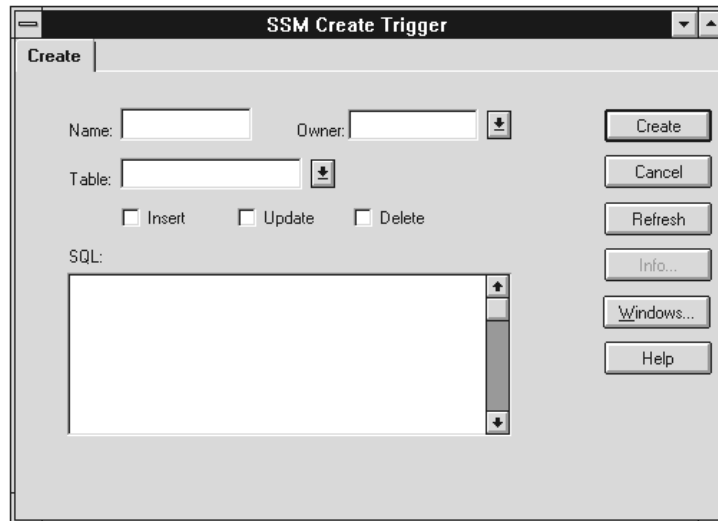


Figure 9-26: Create Trigger dialog box

3. In the Name box, enter a name for the trigger.
4. To make someone other than yourself the owner, choose an owner in the Owner list box.
5. From the Table list, select the table on which to put the trigger.
6. To indicate the action that causes the trigger to execute, select one of the following:
 - Insert
 - Update
 - Delete
7. Enter the text of the trigger in the SQL box. This text can consist of any valid SQL statements that would follow the `as` keyword in a `create trigger` statement.
8. Click Create.

Example

You want to create a trigger, owned by user “homer” and called *newhero*, that executes when the *homer.heroes* table receives an insert because a new hero is added. The trigger should print a reminder to make sure the hero’s story is added to the *titles* table.

In the Create Trigger dialog box, enter the values shown in the following figure:

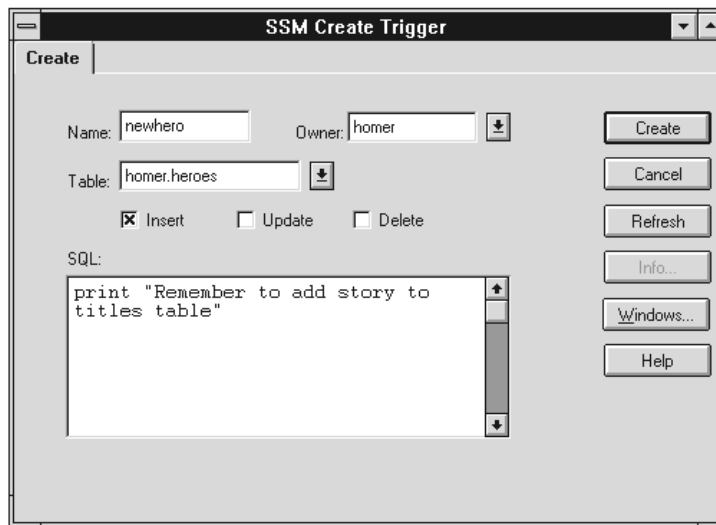


Figure 9-27: Creating the newhero trigger

SQL Server Manager constructs the appropriate SQL query from the data you provide and creates the trigger shown in the following figure.

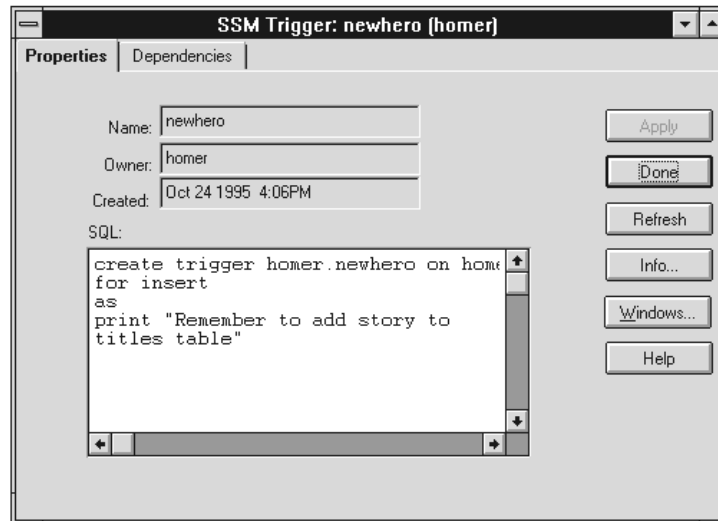


Figure 9-28: newhero trigger as created by SQL Server Manager

Shortcuts



Select the Trigger container icon in the appropriate database and select the Create toolbar button.



Click the right mouse button over the Trigger container icon in the appropriate database and choose Create from the shortcut menu.



Click the right mouse button over the icon of the database in which to create the trigger. From the shortcut menu, choose Create; then, choose Trigger from the cascading menu.

Displaying Trigger Properties

To display the Properties tab:

1. Select the icon of the trigger.
2. From the Trigger menu, choose Properties...

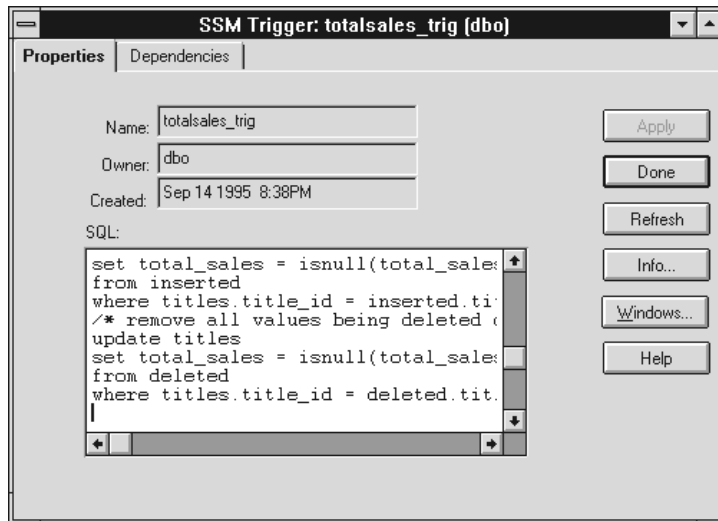




Figure 9-29: Trigger Properties tab


Shortcuts

- 

Double-click the trigger icon.

- 

Select the trigger icon, and select the Properties toolbar button.

- 

Click the right mouse button over the trigger icon; then, choose Properties... from the shortcut menu.

Displaying Trigger Dependencies

To display objects that the trigger references:

1. Display the Trigger Properties tab as described in “Displaying Trigger Properties” on page 9-52.
2. Click Dependencies. The display changes to the Dependencies tab.

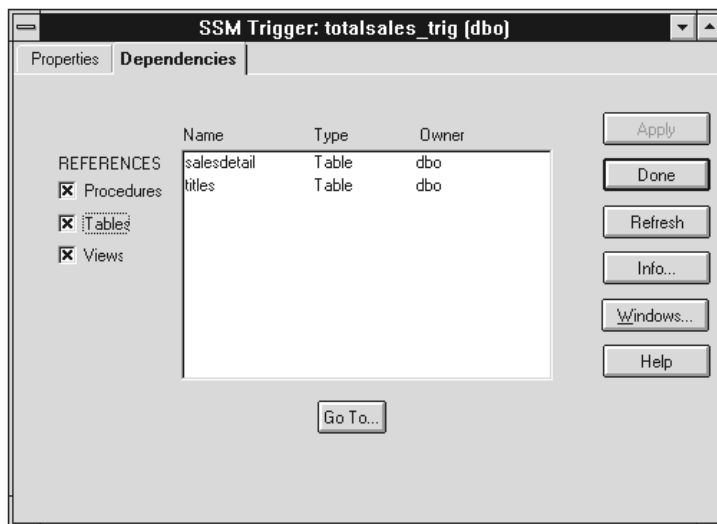


Figure 9-30: Trigger Dependencies tab

3. To display only specific types of objects that the trigger references, clear Procedures, Tables, or Views.

Navigating to Objects with Dependencies on a Trigger

To navigate to objects that the trigger references:

1. Display the Trigger Properties tab as described in “Displaying Trigger Properties” on page 9-52.
2. Click Dependencies. The display changes to the Dependencies tab
3. To display only specific types of objects that the trigger references, clear Procedures, Tables, or Views.
4. To display the Properties tab for an object, select the object in the list and click Go To...

Managing User Datatypes

The datatype of a table column specifies the kind of information (characters, numbers, or dates) the column holds and how the data is stored. For example, the integer (*int*) datatype stores whole numbers in the range of minus 2 billion to plus 2 billion. The tiny integer (*tinyint*) datatype stores whole numbers between 0 and 255.

A **user datatype** is a customized datatype based on system datatypes. You can give a name to a frequently used datatype definition to save time. Unlike system datatypes, user datatypes are case-sensitive. In addition, if a user datatype has *precision*, *scale*, or *identity*, you cannot overwrite these attributes when you use the datatype to define a table column. When you define a user datatype, you can bind it to rules and defaults.

This section describes the following:

- Creating a user datatype
- Displaying user datatype properties
- Editing a user datatype
- Displaying dependencies
- Navigating to objects with dependencies on a user datatype

Creating a User Datatype

To use a user datatype in more than one database, create it in the *model* database template so it is known to all new databases you create.

Privileges	A table owner can define user datatypes.
------------	--

To create a user datatype:

1. Select the icon of the database in which to create the user datatype.
2. From the Database menu, choose Create; then, choose User Datatype from the cascading menu. The Create User Datatype dialog box opens.

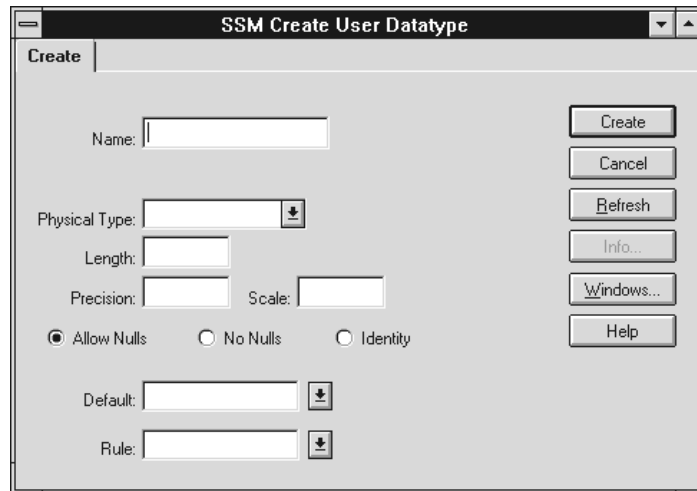
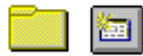


Figure 9-31: Create User Datatype dialog box

3. In the Name box, enter a name for the user datatype.
4. From the Physical Type list, select a system datatype on which to base the user datatype.
5. If the datatype is *char*, *varchar*, *binary*, *varbinary*, *nchar*, or *nvarchar*, enter a maximum length for the datatype in the Length box.
6. If the datatype is *decimal*, *numeric*, or *float*, enter the precision in the Precision box. For *decimal* or *numeric*, precision is the maximum number of decimal digits that can be stored in the column. The range is 1 to 38 with a default of 18. For *float*, precision is the number of significant binary digits in the mantissa.
7. If the datatype is *decimal* or *numeric*, enter the scale in the Scale edit box. Scale is the maximum number of digits to the right of the decimal point. The scale must be less than or equal to the precision. The range is 10 to 38 with a default of 0.

8. To specify that the datatype can accept null values, select **Allow Nulls**. To specify that the datatype does not accept null values, select **No Nulls**.
9. To specify that the datatype can use the system identity property to identify each row in the table, select **Identity**. Only one column can have the identity property.
10. To bind the user datatype to a rule, select a rule from the **Rule list**.
11. To bind the user datatype to a default, select a default from the **Default list**.
12. Click **Create**.

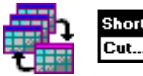
Shortcuts



Select the User Datatype container icon in the appropriate database and select the **Create** toolbar button.



Click the right mouse button over the User Datatype container icon in the appropriate database and choose **Create** from the shortcut menu.



Click the right mouse button over the icon of the database in which to create the user datatype. From the shortcut menu, choose **Create**; then, choose **User Datatype** from the cascading menu.

Displaying User Datatype Properties

To display the Properties tab:

1. Select the icon of the user datatype to display.
2. From the User Datatype menu, choose Properties...

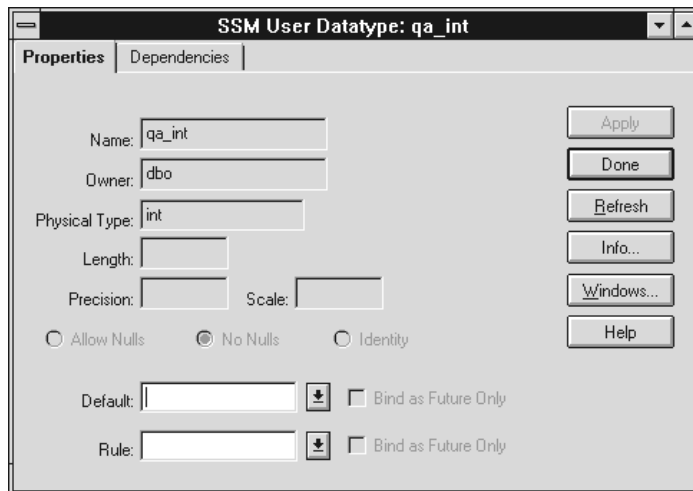


Figure 9-32: User Datatype Properties tab

Shortcuts

Double-click the icon of the user datatype.



Select the user datatype icon, and select the Properties toolbar button.



Click the right mouse button over the icon of the user datatype; then, choose Properties... from the shortcut menu.



Displaying User Datatype Dependencies

To display objects that the user datatype references and that reference the user datatype:

1. Display the User Datatype Properties tab as described in “Displaying User Datatype Properties” on page 9-57.
2. Click Dependencies. The display changes to the Dependencies tab.

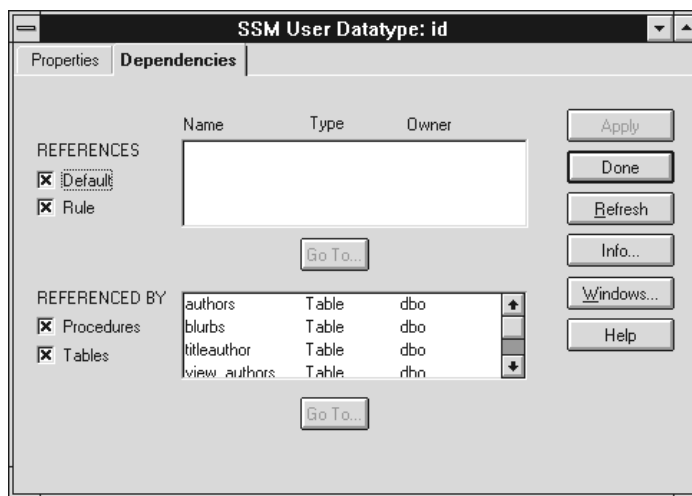


Figure 9-33: User Datatype Dependencies tab

3. To display only defaults that the user datatype references, clear the Rules box in the References list. To display only rules, clear the Defaults box.
4. To display only procedures referenced by the user datatype, clear the Tables box in the Referenced By list. To display only tables, clear the Procedures box.

Navigating to Objects with Dependencies on a User Datatype

To navigate to objects that the user datatype references:

1. Display the User Datatype Properties tab as described in “Displaying User Datatype Properties” on page 9-57.
2. Click Dependencies. The display changes to the Dependencies tab.

3. To display only defaults that the user datatype references, clear the Rules box in the References list. To display only rules, clear the Defaults box.
4. To display only procedures referenced by the user datatype, clear the Tables box in the Referenced By list. To display only tables, clear the Procedures box.
5. To display the Properties tab for an object, select the object in the list and click Go To...

Managing Views

A **view** is a subset of columns from one or more tables. For example, if you frequently want to see the expense column from Table A and the income column from Table B, you can create a named view to give you this information quickly. Once you create a view, you can set permissions for it.

This section describes the following:

- Creating a view
- Displaying view properties
- Displaying view dependencies
- Navigating to objects with dependencies on a view
- Updating user and group view permissions
- Navigating to users and groups

Creating a View

Privileges	Only a database owner or a user or group with create view permission can create a view.
------------	--

To create a view:

1. Select the icon of the database in which to create the view.
2. From the Database menu, choose Create; then, choose View from the cascading menu. The Create View dialog box opens.

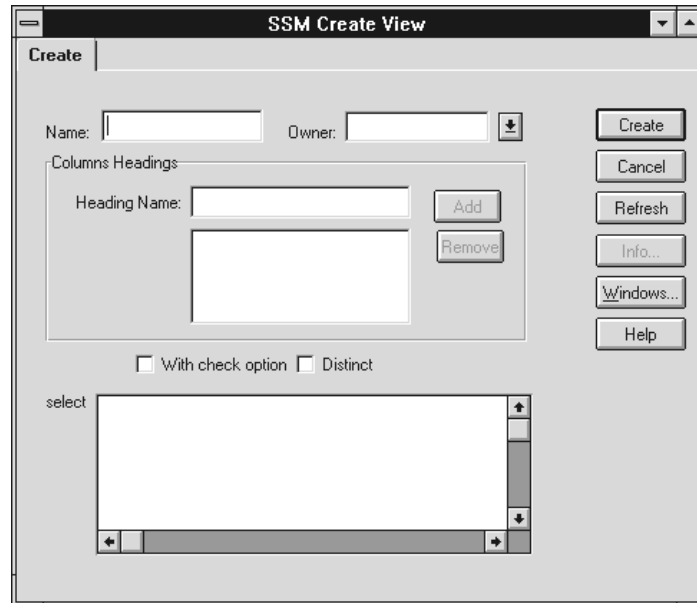


Figure 9-34: Create View dialog box

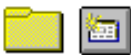


3. In the Name box, enter a name for the view.
4. From the Owner list, choose an owner for the view.
5. To create different column heading names from the columns in the tables used to generate the view, enter each new column heading name in the Heading Name box above the Columns Headings list and click Add. To remove a column name from the Columns Headings list before creating the view, select the name and click Remove.

When SQL Server Manager translates the information in the Create View dialog box to SQL statements, these optional column heading names form the column name specifications that appear in parentheses in the create view command.

6. To validate all data modification statements on the selected columns against the view criteria, select With Check Option.
7. To specify that the view cannot contain duplicate rows, check Distinct.

8. In the Select box, enter the SQL text that follows the select keyword in a create view command. Do not include the select keyword.
9. Click Create.

Shortcuts

	<p>Select the View container icon in the appropriate database, and select the Create toolbar button.</p>
	<p>Click the right mouse button over the View container icon in the appropriate database; then, choose Create from the shortcut menu.</p>
	<p>Click the right mouse button over the icon of the appropriate database. Choose Create from the shortcut menu; then, choose View from the cascading menu.</p>

Example

In the *pubs2* database, you want to create a view, owned by user “homer” and called *titlepub*, that contains book titles and publishers. This information is stored in the *titles* and *publishers* tables. In the new view, you want the column containing title names to be called “book_title” instead of “title,” as it is called in the *titles* table. You want the column containing publisher names to be called “publishers” instead of “pub_name,” as it is called in the *publishers* table.

In the Create View dialog box, enter the values shown in the following figure:

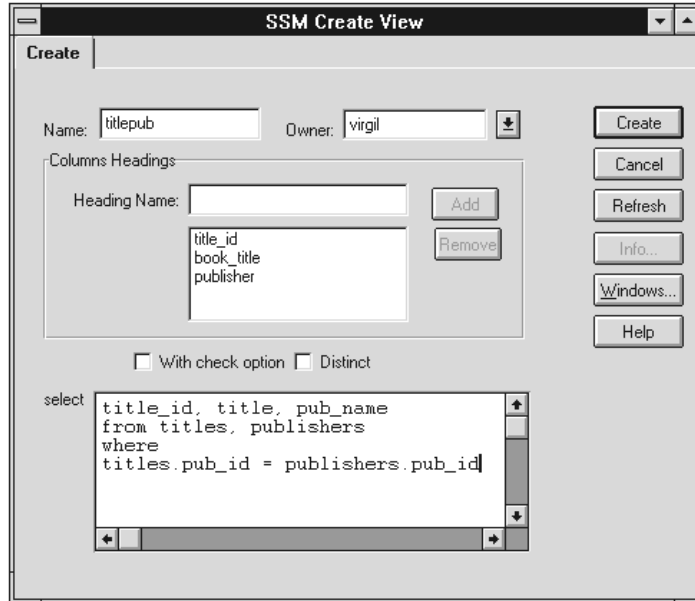


Figure 9-35: Creating the titlepub view

SQL Server Manager constructs the appropriate SQL query from the data you provide and creates the view shown in the following figure.

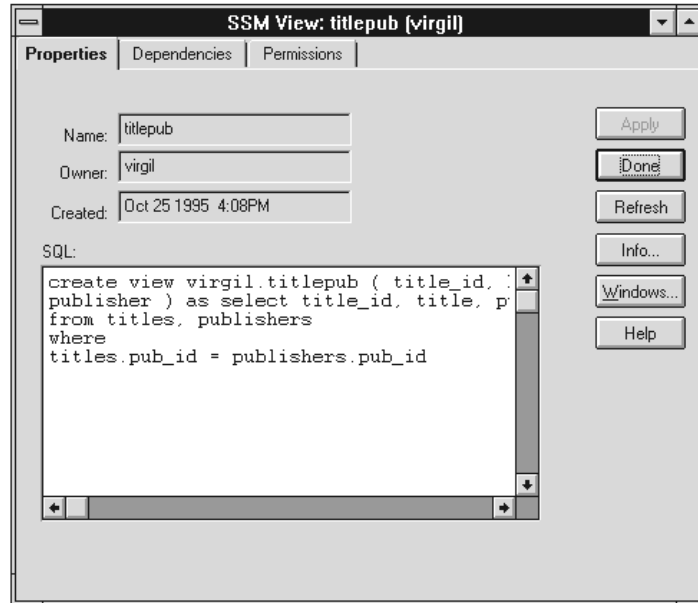


Figure 9-36: titlepub view as created by SQL Server Manager

Note that SQL Server Manager supplies the statements before the select statement from the data entered in the Create dialog box.

Displaying View Properties

To display the Properties tab:

1. Select the icon of the view to display.
2. From the View menu, choose Properties...

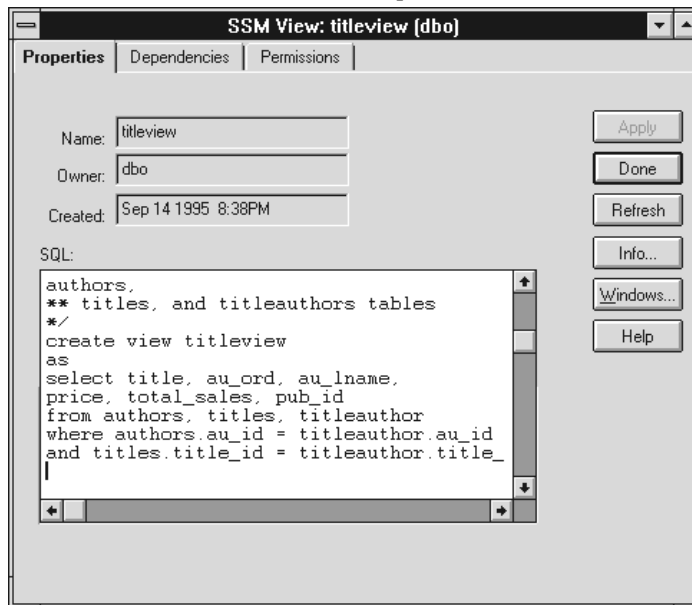


Figure 9-37: View Properties tab

Shortcuts

Double-click the view icon.



Select the view icon, and select the Properties toolbar button.



Click the right mouse button over the view icon; then, choose Properties... from the shortcut menu.



Displaying View Dependencies

To display objects that the view references and that reference the view:

1. Display the View Properties tab as described in “Displaying View Properties” on page 9-64.
2. Click Dependencies. The display changes to the Dependencies tab.

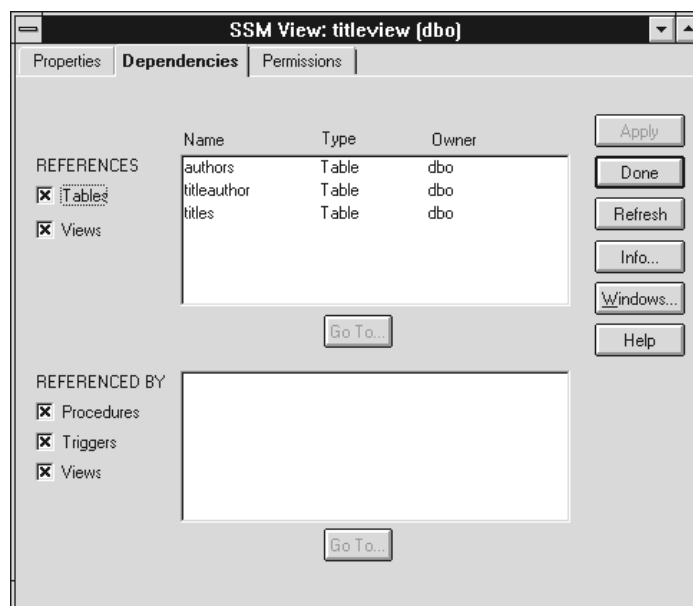


Figure 9-38: View Dependencies tab

3. To display only tables that the view references, clear Views in the References list. To display only views that the view references, clear Tables.
4. To display only objects of a specific type that reference the view, clear Procedures, Triggers, or Views in the Referenced By list.
5. Click Done.

Navigating to Objects with Dependencies on the View

To navigate to objects that the view references and that reference the view:

1. Display the View Properties tab as described in “Displaying View Properties” on page 9-64.
2. Click Dependencies. The display changes to the Dependencies tab.
3. To display only tables that the view references, clear Views in the References list. To display only views that the view references, clear Tables.
4. To display only objects of a specific type that reference the view, clear Procedures, Triggers, or Views in the Referenced By list.
5. To display the Properties tab for the object, select an object in the list and click Go To...

Updating User and Group View Permissions

You can grant and revoke permissions on a view. You can grant the permission to specific users or groups, or you can grant the permission using the **with grant** option so the recipient can also grant the permission to other users. You can revoke the permission from specific users or groups, or you can revoke the permission using the **with cascade** option to revoke it from the named user and all users who acquired the permission from the current user (directly or indirectly).

Privileges	A view owner can grant and revoke object permissions on a view.
------------	---

Displaying the Information

To manage permissions on a view, you can see which users or groups have permissions. You can also see all users and groups in the database and those that do not have any permission on the view.

To display view permissions:

1. Display the View Properties tab, as described in “Displaying View Properties” on page 9-64.
2. Select Permissions. The display changes to the Permissions tab.

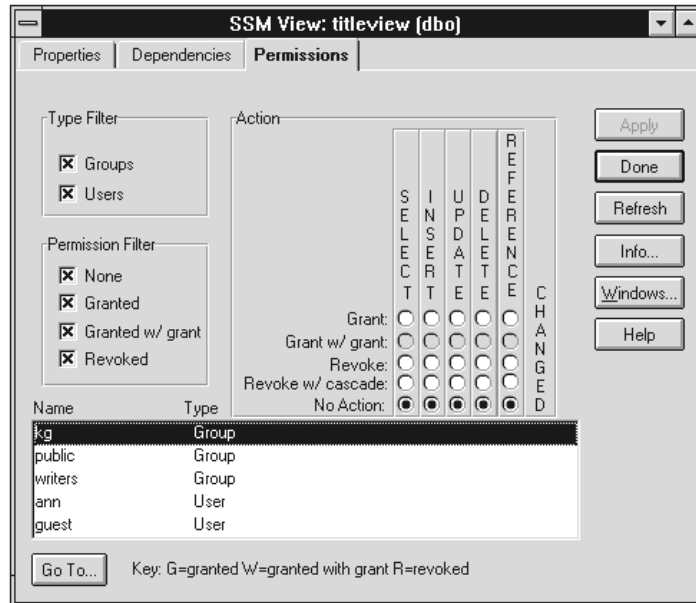


Figure 9-39: View Permissions tab

3. To display only users, clear Groups. To display only groups, clear Users.
4. Initially, the dialog box shows all users or groups and the current state of their permissions. To limit the display, clear the appropriate boxes:
 - None—no permission has been granted on the view to the user or group
 - Granted—at least one permission has been granted on the view to the user or group
 - Granted w/ grant—at least one permission has been granted on the view with grant option
 - Revoked—at least one permission on the view has been revoked from the user or group

Setting the Permissions

To set view permissions:

1. Select the user or group whose permissions you want to change.
2. To grant a permission on the view, select the Grant option button corresponding to the permission in the Action group box. To grant the permission with the grant option, select the Grant w/Grant option button corresponding to the permission. You can grant the following types of permissions:
 - Select—permission to select the view
 - Insert—permission to insert a row in the view
 - Update—permission to update a row in the view
 - Delete—permission to delete a row in the view
 - References—permission to create a referential constraint that references the view
3. To revoke a permission on the view, select the Revoke option button corresponding to the permission in the Action group box. To revoke the permission with cascade, select the Revoke w/Cascade option button corresponding to the permission.
4. When you make a permission change for a user or group, the “N” in the Changed column changes to a “Y,” indicating that you have altered the original settings. Repeat Steps 1 through 3 for each user or group for whom you want to update permissions.
5. Click Apply to update the permissions in the database. SQL Server Manager updates the list with one of the following codes in each column that corresponds to the type of permission you changed:
 - G—granted
 - W—granted with grant
 - R—revoked
 - Blank—no permission explicitly assigned

► Note

SQL Server Manager reports on permissions explicitly granted or revoked. It does not show permissions that users obtain by being members of a group or by having an administrative role, such as System Administrator, associated with their logins.

Shortcut

Drag and drop the view icon on a user icon to display the user Permissions tab.



Drag and drop the view icon on a group icon to display the group Permissions tab.

Navigating to Users and Groups

To navigate to users and groups from a view:

1. Display the View Properties tab, as described in “Displaying View Properties” on page 9-64.
2. Select Permissions. The display changes to the Permissions tab.
3. To display only users, clear Groups. To display only groups, clear Users.
4. Initially, the dialog box shows all users or groups and the current state of their permissions. To limit the display, clear the appropriate boxes:
 - None—no permission has been granted on the view to the user or group.
 - Granted—at least one permission has been granted on the view to the user or group.

- Grant w/Grant—at least one permission has been granted on the view with grant option. The grant option is available only for users, not for groups.
 - Revoked—at least one permission on the view has been revoked from the user or group.
5. Select a user or group in the Name list. Click Go To... to display the Properties tab for the user or group.

10

Managing User Tables

What's in This Chapter

Tables are the heart of a database. Consisting of columns and rows that contain specific kinds of data, a table is the mechanism for storing the data in a database:

- A **system table** stores information that allows the database to perform its services.
- A **user table** stores and provides access to user data.

System table definitions are not typically updated. This chapter describes how to use SQL Server Manager to create and update user table definitions.

The chapter describes the following tasks:

- Creating and updating a table
- Displaying table properties
- Displaying table dependencies
- Updating user and group table permissions
- Placing a table on a segment
- Displaying indexes and triggers
- Creating check constraints on a table
- Creating table referential constraints on a table
- Recompiling procedures and triggers
- Checking table consistency
- Updating statistics
- Partitioning a table

Creating a Table

When you create a table, you define the characteristics of its columns. You can also specify **constraints** on the data a column contains. This section discusses constraints and then describes how to create a table, including how to specify column definitions and how to specify constraints on a column.

Using Constraints

Constraints allow you to specify controls on the type of data that can be added to a table column. SQL Server Manager lets you specify the following types of constraints:

- The value that appears in the column if no value is entered on an insertion or update. This type of constraint is called a **default**.
- Criteria against which SQL Server checks data entered on an insertion or update. This type of constraint is called a **rule** or **check constraint**.
- Key values in another table that must match key values in the row being inserted or updated. This type of constraint is called a **referential constraint**.

SQL Server Manager provides multiple ways of implementing constraints.

Implementing Defaults

You can create a default in the following ways:

- As part of a table column definition. As you create or add columns to a table, you can specify default values that apply only to those columns by using the Table Column Advanced dialog box.
- As a database object. Once you create a default, you can bind it to multiple table columns and to user datatypes.

The advantage of the first method is that you specify the table column and its default value in the same step. The advantage of the second is that you can define the default once and apply it as needed to multiple table columns. Either method is available whenever you create a table column.

This section describes creating a default as part of a table column definition. For information on creating a default as a database object, see “Creating a Default:” on page 9-9.

Implementing Rules and Check Constraints

You can create a rule or check constraint in the following ways:

- As part of a table column definition. This type of constraint, called a column check constraint, applies only to the column on

which you define it. You create it as you create the column definition, using the Table Column Advanced dialog box.

- As part of a table definition. This type of constraint, called a table check constraint, can apply to all columns of a table. To create a check constraint on a table, use the Check Constraint tab of the Table dialog box.
- As a database object. This type of constraint is called a rule. Once you create a rule, you can bind it to multiple table columns and to user datatypes. To create a rule, use the Create Rule dialog box.

The advantage of the first two methods is that you specify the table or column and its constraint in the same step. The advantage of the third is that you define the rule once and apply it as needed to multiple table columns. All methods are available whenever you create a table column.

This section describes the method in which you create a column check constraint on a table column. For information on creating a table check constraint, see “Creating Check Constraints on a Table” on page 10-25. For information on creating a rule and binding it to a table column, see “Creating a Rule” on page 9-40 and “Binding a Rule” on page 9-44.

Implementing Referential Constraints

Referential constraints apply to a table as a whole. To create referential constraints, first create the table and then modify it by using the Referential Constraints tab of the Table dialog box. For details, see “Creating Referential Constraints on a Table” on page 10-30.

Constraints Summary

The following table summarizes the types of constraints you can place on table data and lists how to implement them by using SQL Server Manager.

Table 10-1: Types of table constraints

Constraint type	Column level?	Table level?	Dialog box where implemented
Default	Yes	No	<ul style="list-style-type: none"> • Table Column Advanced dialog box to create a default or to bind an existing default to the column • Create Default dialog box to create a default, Default Bindings tab to bind a default to a column
Check	Yes	Yes	<ul style="list-style-type: none"> • Table Column Advanced dialog box to create a check constraint that applies to a single column • Check Constraints tab of Table dialog to create a check constraint that applies to a whole table
Rule	Yes	No	<ul style="list-style-type: none"> • Table Column Advanced dialog box to bind an existing rule to the column • Create Rule dialog box to create a rule, Rule Bindings tab to bind a rule to a column
Referential	No	Yes	<ul style="list-style-type: none"> • Referential Constraints tab of Table dialog box

Instructions


Privileges	Only a database owner or a user or group with create table permission can create a table.
------------	---

To create a table:

1. Select the icon of the database in which to create the table.
2. From the Database menu, choose Create; then, choose Table from the cascading menu. The Create Table dialog box opens.



Figure 10-1: Create Table dialog box

3. In the Name box, enter a name for the table.
4. From the Owner list, choose an owner for the table.
5. From the Segment list, choose a segment on which to store the table.
6.  Optionally, to override the default assigned by SQL Server for the maximum number of index rows to store on a data page, enter a number in the Maximum Rows Per Page box. This option is available in SQL Server release 11.0 or later.


For a table with clustered indexes, values between 0 and 256 are valid. For a table with nonclustered indexes, the largest allowable Maximum Rows Per Page value depends on the size of the index key. To approximate the largest allowable value, subtract 32 from the page size and divide the resulting number by the index key size.

7. Specify column definitions, as described in “Specifying Column Definitions” on page 10-7.

8. Optionally, create column constraints, as described in “Specifying Column Constraints” on page 10-8.

► **Note**

When creating a table, you can create column constraints on only one column. To create column constraints on additional columns, **do not add those columns when initially creating the table**. Instead, add them after the table is created, as described in “Adding Columns to a Table” on page 10-13.

9.  If you are connected to a release 11.0 SQL Server, you can select a cache strategy for the table. This selection overrides the default strategy, determined by the SQL Server optimizer, for reading data pages from a table into the buffers in data cache. The following selections are available:

- Most Recently Used Replacement—This selection specifies that SQL Server uses the most recently used (MRU) strategy for determining wherein cache to place data pages when reading in new data.

If you clear the check box, SQL Server reads new pages into the MRU end of the chain of buffers in cache. Subsequent reads move the pages along the chain towards the least recently used (LRU) end until they are flushed out by new reads at the MRU end. If you select Most Recently Used Replacement, SQL Server reads new pages into the LRU end. They are used and immediately flushed when a new page enters the MRU end.

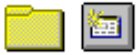


This strategy is advantageous when a page is needed only once for a query. It tends to keep such pages from flushing out other pages that can potentially be reused while still in cache.

- Large Buffer Prefetch—This selection applies if one or more large buffer pools is defined in the default cache or, if the table is bound to a named cache, in the named cache. A large buffer pool is one that has buffers larger than the 2K default, as specified in the Cache Properties dialog box. If you select Large Buffer Prefetch, the SQL Server optimizer can fetch data in I/Os of as many as eight 2K data pages at a time instead of the default of one page at a time.

This strategy is advantageous for data that is stored and accessed sequentially; for example, it can improve performance for queries that scan the table sequentially.

10. Click Create.

Shortcuts

	<p>Select the Table container icon in the appropriate database, and select the Create toolbar button.</p>
	<p>Click the right mouse button over the Table container icon in the appropriate database; then, choose Create from the shortcut menu.</p>
	<p>Click the right mouse button over the appropriate database icon. Choose Create from the shortcut menu; then, choose Table.</p>

Specifying Column Definitions

For each column in the table:

1. In the Name box, enter a column name.
2. From the Datatype list, select a datatype for the column.
3. Enter the specifications appropriate to the datatype:
 - If the datatype is *char*, *nchar*, *varchar*, *nvarchar*, *binary*, or *varbinary*, enter a length for the datatype in the Length box.
 - If datatype is *decimal*, *numeric*, or *float*, enter the precision in the Precision edit box. For *decimal* or *numeric*, precision is the maximum number of decimal digits that can be stored in the column. Specify a precision of 1 to 38. For *float*, precision is the number of significant binary digits in the mantissa.
 - If datatype is *decimal* or *numeric*, enter the scale in the Scale edit box. Scale is the maximum number of digits that can be stored to the right of the decimal point. The scale must be less than or equal to the precision. Specify a scale of 0 to 38.
4. Select an option in the Nulls group box:
 - To allow NULL values in the column, select the Allow option button.

- To disallow NULL values in the column, select the Disallow option button.
 - To allow the column to contain a system-generated, sequential value that identifies each row in the table, select the Identity option button. Only one column in a table can have this characteristic. When you specify an identity column, SQL Server Manager automatically sets its datatype to *numeric*, its precision to 18, and its scale to 0.
5. Click Add to add the column to the scrollable list. To delete an unapplied column, select the column definition in the scrollable list and click Remove. To change an unapplied column definition, select the definition in the scrollable list, make the changes, and click Change. Use Change and Remove **before** you create the table.

Specifying Column Constraints

After creating specifications for a column on which to add constraints and after adding the column to the list, but **before** creating the table, use the following steps to specify column constraints:

► **Note**

When creating a table, you can create column constraints on only one column. To create column constraints on additional columns, **do not add those columns when initially creating the table**. Instead, add them after the table is created, as described in “Adding Columns to a Table” on page 10-13.

1. Select the column in the column list.
2. Click Advanced... The Table Column Advanced dialog box opens, and the column specifications are transferred to the boxes at the top of the dialog box:

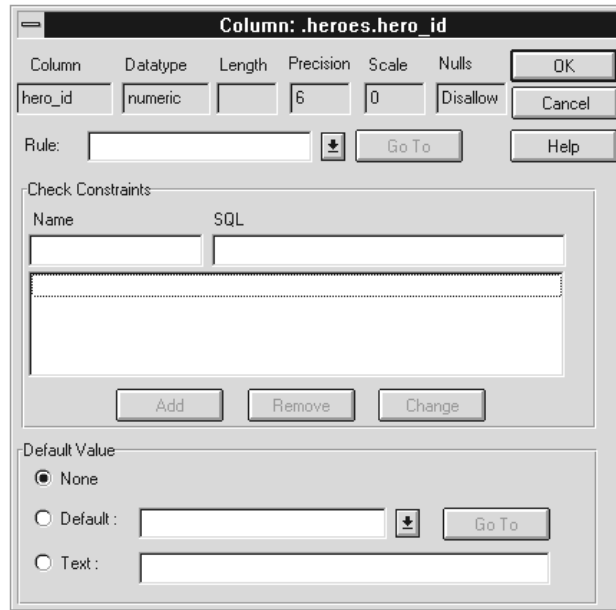


Figure 10-2: Table Column Advanced dialog box

3. To bind the column to a rule, select the rule from the Rule drop-down list. To display the Properties tab of the selected rule, click Go To...
4. To specify a column check constraint:
 - In the Name box, enter the constraint name. If you do not specify a name, SQL Server generates one.
 - In the SQL box, enter the text of the constraint. Check constraint text specifies a search condition that any value must pass before it is inserted into the table. For format information, see "Check Constraint Format."
 - To add the constraint to the Check Constraints list, click Add. To remove a constraint from the list, select the constraint in the list and click Remove. To change a constraint, select the constraint, make changes in the edit boxes, and click Change.
5. To specify a default constraint, select one of the following options:

- To prevent the column from having a default constraint, select None.
- To bind the column to an existing default, select the Default option button, and select the default from the corresponding drop-down list. To display the Properties tab of the selected default, click Go To...
- To specify the value of the default, select Text and enter the value in the accompanying box. The text of the default must be a valid constant expression that is compatible with the datatype of the column. The constant expression cannot include the name of any columns or other database objects, but can include built-in functions that do not reference database objects. Enclose strings in double quotes.

A table column can have only one default.

6. Click OK. In the Create Table dialog box, columns with constraints are identified by an "A" in the rightmost column of the list.

Check Constraint Format

The format of the information to enter in the SQL box consists of the check keyword followed by a search condition, enclosed in parentheses. The search condition specified by a check constraint can include any of the following:

- A list of constant expressions introduced with *in*
- A range of constant expressions introduced with *between*
- A set of conditions, introduced with *like*, that can contain wildcard characters

An expression can include arithmetic operators and Transact-SQL built-in functions. The search condition cannot contain subqueries, aggregate functions, or a host variable or parameter.

For example, the following constraint on the *pub_id* column specifies that the value of *pub_id* must be 1389, 0736, 0877, 1622, 1756, or any 4-digit number beginning with 99:

```
check (pub_id in "1389", "0736", "0877", "1622",  
"1756" or pub_id like "99[0-9][0-9]")
```

Deleting a Table

When you delete a table, SQL Server deletes any indexes and triggers associated with the table and unbinds any rules or defaults that are bound to its columns.

Privileges	A table owner can delete a table.
------------	-----------------------------------

To delete a table:

1. Select the icon of the table to delete.
2. From the Table menu, choose Delete.
3. In the confirmation dialog box, click Yes.

Shortcuts

Select the table icon, and select the Delete toolbar button.



Click the right mouse button over the table icon; then, choose Delete from the shortcut menu.



Displaying Table Properties

The Table Properties tab provides current information about a table. In addition, it is the starting place for any modifications you make to a table. These are the types of modifications you can make:

- Adding columns and column constraints
- Adding or deleting check or referential constraints
- Adding or modifying table permissions
- Placing new table growth on a different segment
- In a release 11.0 SQL Server, partitioning or unpartitioning the table

To display the Table Properties tab:

1. Select the icon of the table to display.
2. From the Table menu, choose Properties...

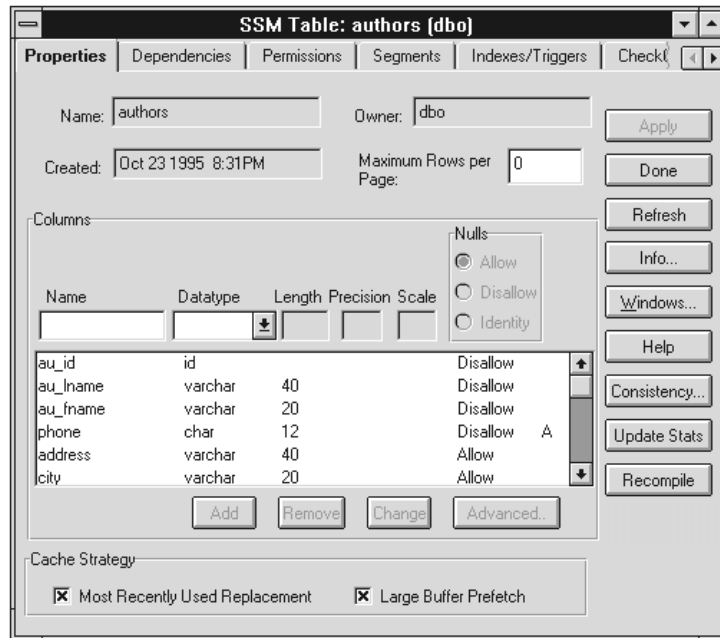


Figure 10-3: Table Properties tab

Shortcuts

Double-click the table icon.



Select the table icon, and select the Properties toolbar button.



Click the right mouse button over the table icon, and choose Properties... from the shortcut menu.



Adding Columns to a Table

To add columns to a table:

1. Display the Properties tab of the table as described in “Displaying Table Properties” on page 10-11.
2. For each column you add, follow the instructions in “Specifying Column Definitions” on page 10-13.
3. To create constraints on new columns, follow the instructions in “Specifying Column Constraints” on page 10-14.

► **Note**

You can add columns on which you do not create constraints in a batch; however, columns for which you create constraints must be specified and applied one at a time.

4. Click Apply.

Specifying Column Definitions

For each column you add:

1. In the Name box, enter a column name.
2. From the Datatype list, select a datatype for the column.
3. Enter the specifications appropriate to the datatype:
 - If the datatype is *char*, *nchar*, *varchar*, *nvarchar*, *binary*, or *varbinary*, enter a length for the datatype in the Length box.
 - If the datatype is *decimal*, *numeric*, or *float*, enter the precision in the Precision edit box. For *decimal* or *numeric*, precision is the maximum number of decimal digits that can be stored in the column. Specify a precision of 1 to 38. For *float*, precision is the number of significant binary digits in the mantissa.
 - If the datatype is *decimal* or *numeric*, enter the scale in the Scale edit box. Scale is the maximum number of digits that can be stored to the right of the decimal point. The scale must be less than or equal to the precision. Specify a scale of 0 to 38.
4. Select an option in the Nulls group box:
 - To allow NULL values in the column, select the Allow option button.

- To disallow NULL values in the column, select the Disallow option button.
 - To allow the column to contain a system-generated, sequential value that identifies each row in the table, select the Identity option button. Only one column in a table can have this characteristic. When you specify an identity column, SQL Server Manager automatically sets its datatype to *numeric*, its precision to 18, and its scale to 0.
5. Click Add to add the column to the scrollable list. To delete an unapplied column, select the column definition in the scrollable list and click Remove. To change an unapplied column definition, select the definition in the scrollable list, make the changes, and click Change. Use Change and Remove **before** clicking Apply.

Specifying Column Constraints

After creating specifications for a column on which to add constraints and after adding the column to the list, but **before** clicking Apply, use the following steps to specify column constraints:

1. Select the column in the column list.
2. Click Advanced... The Table Column Advanced dialog box opens, and the column specifications are transferred to the boxes at the top of the dialog box:

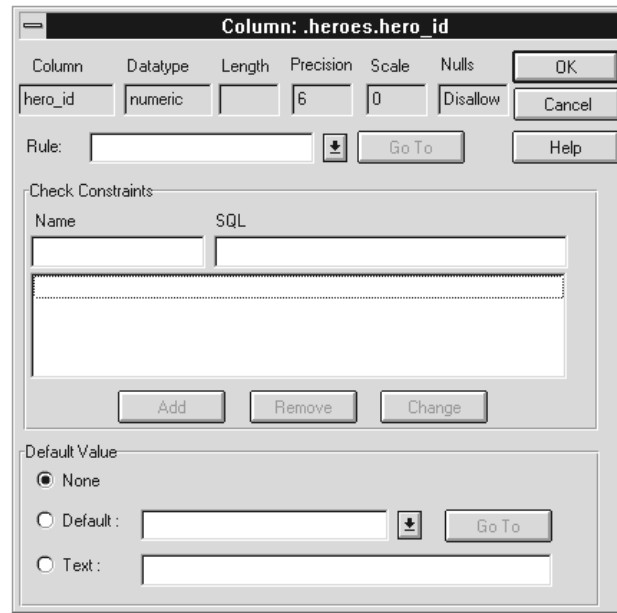


Figure 10-4: Table Column Advanced dialog box

3. To bind the column to a rule, select the rule from the Rule drop-down list. To display the Properties tab of the selected rule, click Go To...
4. To specify a column check constraint:
 - In the Name box, enter the constraint name. If you do not specify a name, SQL Server generates one.
 - In the SQL box, enter the text of the constraint. Check constraint text specifies a search condition that any value must pass before it is inserted into the table. For format information, see “Check Constraint Format.”
 - To add the constraint to the Check Constraints list, click Add. To remove a constraint from the list, select the constraint in the list and click Remove. To change a constraint, select the constraint, make changes in the edit boxes, and click Change.
5. To specify a default constraint, select one of the following options:
 - To prevent the column from having a default constraint, select None.

- To bind the column to an existing default, select the Default option button, and select the default from the corresponding drop-down list. To display the Properties tab of the selected default, click Go To...
- To specify the value of the default, select Text and enter the value in the accompanying box. The text of the default must be a valid constant expression that is compatible with the datatype of the column. The constant expression cannot include the name of any columns or other database objects, but can include built-in functions that do not reference database objects. Enclose strings in double quotes.

A table column can have only one default.

6. Click OK. In the Table Properties tab, columns with constraints are identified by an "A" in the rightmost column of the list.

Check Constraint Format

The format of the information to enter in the SQL box consists of the check keyword followed by a search condition, enclosed in parentheses. The search condition specified by a check constraint can include any of the following:

- A list of constant expressions introduced with *in*
- A range of constant expressions introduced with *between*
- A set of conditions, introduced with *like*, that can contain wildcard characters

An expression can include arithmetic operators and Transact-SQL built-in functions. The search condition cannot contain subqueries, aggregate functions, or a host variable or parameter.

For example, the following constraint on the *pub_id* column specifies that the value of *pub_id* must be 1389, 0736, 0877, 1622, 1756, or any 4-digit number beginning with 99:

```
check (pub_id in "1389", "0736", "0877", "1622",
"1756" or pub_id like "99[0-9][0-9]")
```



Setting Maximum Rows Per Page

To override the default assigned by SQL Server for the maximum number of index rows to store on a data page, enter a number in the Maximum Rows Per Page box. This option is available in SQL Server release 11.0 or later.

For a table with clustered indexes, values between 0 and 256 are valid. For a table with nonclustered indexes, the largest allowable Maximum Rows Per Page value depends on the size of the index key. To approximate the largest allowable value, subtract 32 from the page size and divide the resulting number by the index key size.

Displaying Table Dependencies

You can display defaults, rules, user datatypes, and tables that a table references, and you can display procedures, views, triggers, and tables that reference the table.

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Click Dependencies. The display changes to the Dependencies tab.

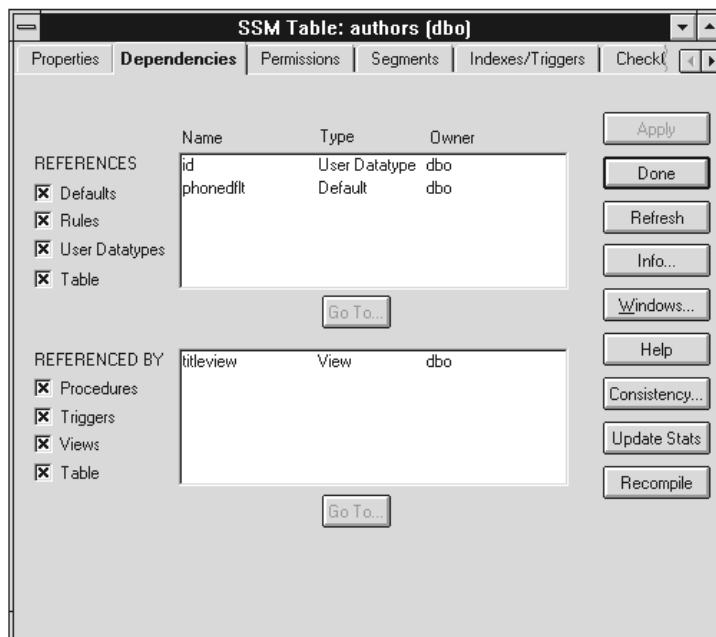


Figure 10-5: Table Dependencies tab

3. To display only specific object types that the table references, clear Defaults, Rules, User Datatypes, or Tables in the References list.
4. To display only specific object types referenced by the table, clear Procedures, Views, Triggers, or Tables in the Referenced By list.

Navigating to Objects with Dependencies on a Table

To navigate to objects that reference a table or that the table references:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Click Dependencies.
3. To display only specific object types that the table references, clear Defaults, Rules, User Datatypes, or Tables in the References list.
4. To display only specific object types referenced by the table, clear Procedures, Views, Triggers, or Tables in the Referenced By list.
5. To display the Properties tab for an object, select the object in the list and click Go To...

Updating User and Group Table Permissions

You can grant and revoke permissions on a table. You can grant the permission to specific users or groups, or you can grant the permission using the **with grant** option so the recipient can also grant the permission to other users. You can revoke the permission from specific users or groups, or you can revoke the permission using the **with cascade** option to revoke it from the named user and all users who acquired the permission from the current user (directly or indirectly).

Privileges	A table owner can grant and revoke object permissions on a table.
------------	---

Displaying the Information

To manage permissions on a table, you can see which users or groups have permissions. You can also see all users and groups in the database and those that do not have any permission on the table.

To display table permissions:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Click Permissions. The display changes to the Permissions tab.

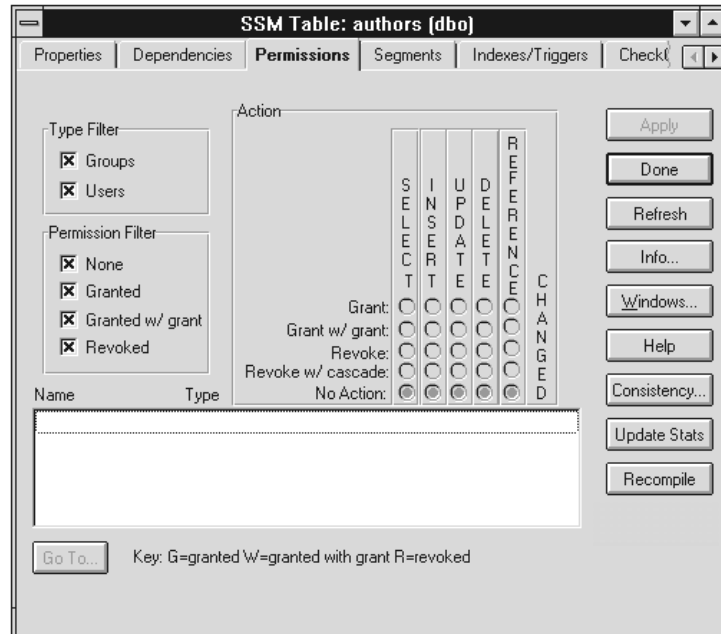


Figure 10-6: Table Permissions tab

3. To display only users, clear Groups in the Type Filter group box. To display only groups, clear Users.
4. Initially, the dialog box shows all users or groups and the current state of their permissions. To limit the display, clear the appropriate boxes in the Permission Filter group box:
 - None—no permission has been granted on the table to the user or group.
 - Granted—at least one permission has been granted on the table to the user or group.
 - Grant w/Grant—at least one permission has been granted on the table with grant option. The grant option is available only to users, not to groups.

- Revoked—at least one permission on the table has been revoked from the user or group.

Setting the Permissions

To set table permissions:

1. Select the user or group whose permissions you want to change.
2. To grant a permission on the table, select the Grant option button corresponding to the permission in the Action group box. To grant the permission with the grant option, select the Grant w/Grant option button corresponding to the permission. You can grant the following types of permissions:

- Select—permission to select the table
- Insert—permission to insert a row in the table
- Update—permission to update a row in the table
- Delete—permission to delete a row in the table
- Reference—permission to create a referential constraint that references the table

3. To revoke a permission on the table, select the Revoke option button corresponding to the permission in the Action group box. To revoke the permission with cascade, select the Revoke w/Cascade option button corresponding to the permission.

When you make a permission change for a user or group, the Changed column displays a check mark (√), indicating that you have altered the original settings.

4. Repeat Steps 1 through 3 for each user or group for whom you want to update permissions.
5. Click Apply to update the permissions in the database. SQL Server Manager updates the list with one of the following codes in each column that corresponds to the type of permission you changed:

- G—granted
- W—granted with grant
- R—revoked
- Blank—no permission explicitly assigned

► **Note**

SQL Server Manager reports on permissions explicitly granted or revoked. It does not show permissions that users obtain by being members in a group or by having an administrative role, such as System Administrator, associated with their logins.

Shortcuts



Drag and drop the table icon on a user icon to display the User Permissions tab.



Drag and drop the table icon on a group icon to display the group Permissions tab.

Navigating to Users and Groups

To navigate to users and groups from a table:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Click Permissions. The display changes to the Permissions tab.
3. To display only users, clear Groups. To display only groups, clear Users.
4. Initially, the dialog box shows all users or groups and the current state of their permissions. To limit the display, clear the appropriate boxes:
 - None—no permission has been granted on the table to the user or group.
 - Granted—at least one permission has been granted on the table to the user or group.
 - Grant w/Grant—at least one permission has been granted on the table with grant option. The grant option is available only to users, not to groups.
 - Revoked—at least one permission on the table has been revoked from the user or group.

5. Select a user or group in the Name list. To display the Properties tab for the user or group, click Go To...

Placing a Table on a Segment

Using a segment to put a table on a specific database device can increase SQL Server performance and can give increased control over placement, size, and space usage of database objects. For example:

- If you put a table on one device, and its nonclustered indexes on a device on another disk controller, the time required to read or write to the disk can be reduced.
- If you put tables and indexes only on specific segments, those objects cannot grow beyond the space available on the devices represented by the segments.
- If you put tables and indexes only on specific segments, other objects cannot contend for space with them.
- If a large, heavily-used table is split across devices on two separate disk controllers, read and write time may be improved.

To place a table on a segment:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Click Segments. The display changes to the Segments tab.

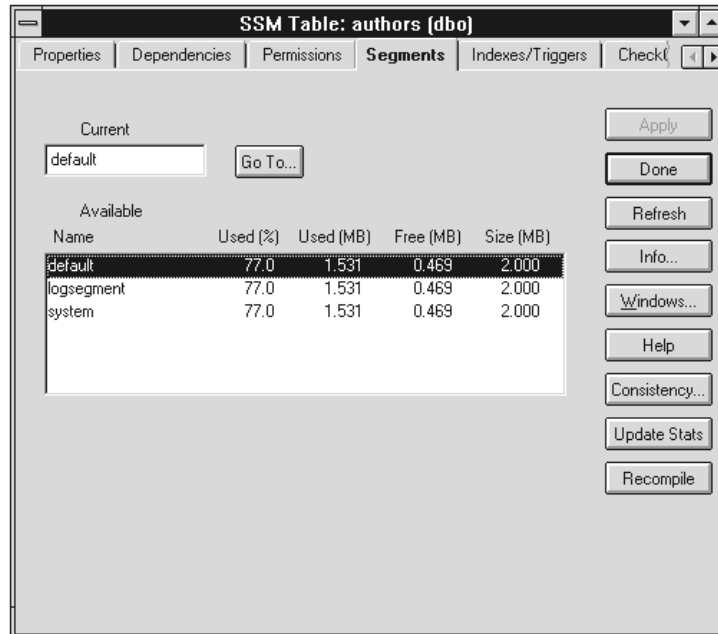
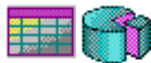


Figure 10-7: Table Segments tab

3. From the Available list box, select the segment where you want future growth of the table to occur. The name you select appears in the Current box.
4. Click Apply.

Shortcut

Drag and drop the table icon onto the segment icon.



Navigating to Segments

To navigate to a segment from a table:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Click Segments. The display changes to the Segments tab.
3. To display the Properties tab for a segment in the scrolling list, select the segment; then, click Go To...

Displaying Indexes and Triggers

To display triggers and indexes associated with a table:

1. Display the Table Properties tab as described in “Displaying Table Properties” on page 10-11.
2. Select Indexes/Triggers. The display changes to the Indexes/Triggers tab.

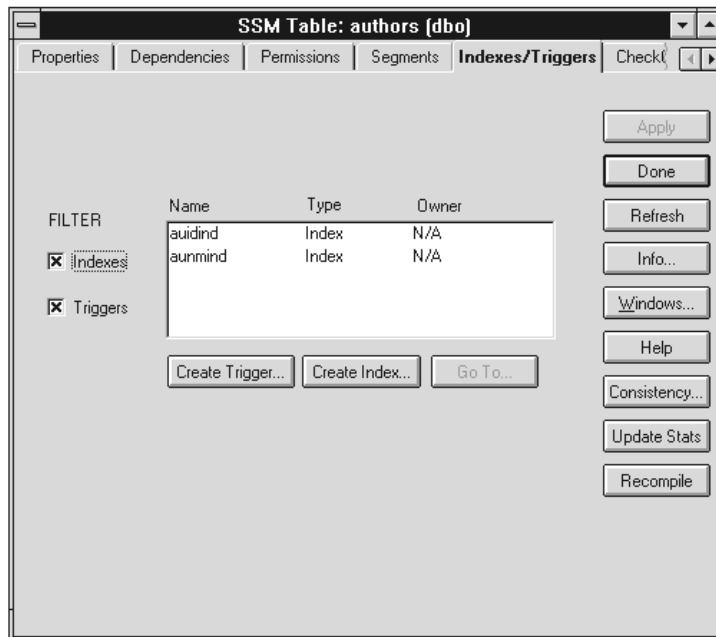


Figure 10-8: Table Indexes/Triggers tab

3. To display only indexes associated with the table, clear Triggers. To display only triggers associated with the table, clear Indexes.

Navigating to Indexes and Triggers

To navigate to indexes and triggers associated with the table:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Select Indexes/Triggers. The display changes to the Indexes/Triggers tab.
3. To display only indexes associated with the table, clear Triggers. To display only triggers associated with the table, clear Indexes.
4. To display the Properties tab for an index or trigger, select the object; then, click Go To...

Creating a New Index or Trigger

From the Indexes/Triggers tab, you can open the Create dialog box to create a new index or trigger without leaving the Table dialog box. To create an index or trigger:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Select Indexes/Triggers. The display changes to the Indexes/Triggers tab.
3. To open the Create Index dialog box, click Create Index... To open the Create Trigger dialog box, click Create Trigger...
4. Create the index or trigger as described in “Creating an Index” on page 9-18 or “Creating a Trigger” on page 9-48.

Creating Check Constraints on a Table

Check constraints allow you to specify criteria that data must meet before it can be added to a table. You can create check constraints on a single table column or on an entire table. A constraint on a table column applies only to that column. A constraint defined on a table can apply to any column in the table.

This section describes creating check constraints on a table. For general information on constraints and for instructions on creating

check constraints on a table column, see “Creating a Table” on page 10-1.

To create a table check constraint:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Click Check Constraints. The display changes to the Check Constraints tab:

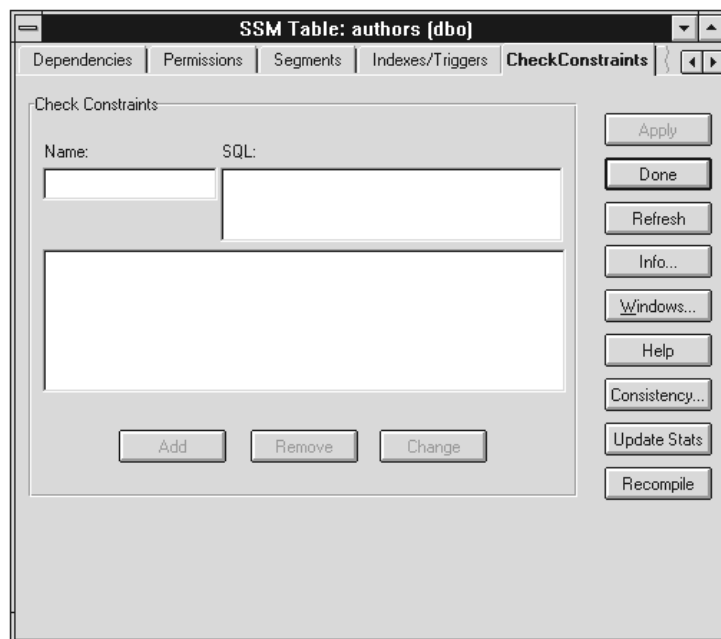


Figure 10-9: Table Check Constraints tab

3. In the Name box, enter the constraint name. If you do not specify a name, SQL Server generates one.
4. In the SQL box, enter the text of the constraint. Check constraint text specifies a search condition that any value must pass before it is inserted into the table. For format information, see “Check Constraint Format.”
5. To add the constraint to the Check Constraints list, click Add. To remove a constraint from the list, select the constraint in the list and click Remove. To change a constraint, select the constraint, make changes in the edit boxes, and click Change.
6. Click Apply.

Check Constraint Format

The format of the information to enter in the SQL box consists of the check keyword followed by a search condition, enclosed in parentheses. The search condition specified by a check constraint can include any of the following:

- A list of constant expressions introduced with *in*
- A range of constant expressions introduced with *between*
- A set of conditions, introduced with *like*, that can contain wildcard characters

An expression can include arithmetic operators and Transact-SQL built-in functions. The search condition cannot contain subqueries, aggregate functions, or a host variable or parameter.

For example, the following constraint on the *pub_id* column specifies that the value of *pub_id* must be 1389, 0736, 0877, 1622, 1756, or any 4-digit number beginning with 99:

```
check (pub_id in "1389", "0736", "0877", "1622",  
"1756" or pub_id like "99[0-9][0-9]")
```



Partitioning and Unpartitioning a Table

When connected to a Release 11.0 SQL Server, you can split a table into partitions across a segment. Partitioning allows the load of page allocation requests for the segment, which can span multiple database devices, to be distributed across the devices. Because multiple chains of data pages are available for inserts, concurrent inserts are possible, page contention is reduced, and performance is improved.

Partitioning is advantageous for a table you are populating by using the *bcp* utility or for a large, already populated table with busy insert activity. You can partition both empty tables and tables that contain data. Partitioning does not move data; existing data remains where it was originally stored in the first partition. For best performance, partition a table before inserting data.

Rules for Partitioning

The following rules govern table partitioning:

- You cannot partition a system table, a user table with a clustered index, or a table that is already partitioned.
- Once you have partitioned a table, you cannot create a clustered index on it or place it on a specified partition. To perform these operations, you must first unpartition the table.

Partitioning

To partition a table:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Click Partitions. The display changes to the Partition tab:

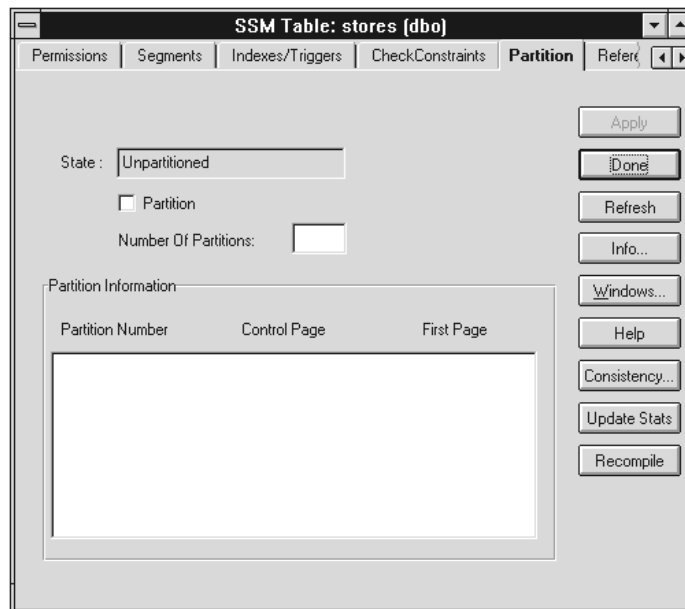


Figure 10-10: Table Partition tab

3. Select the Partition check box.
4. In the Number of Partitions box, enter the number of partitions into which to split the table.

5. Click **Apply**. For each partition you request, SQL Server allocates a control page, and for each partition other than the first, SQL Server allocates a page as the first in the partition chain. When you click **Apply**, the page numbers appear in the **Partition Information** group box.

Unpartitioning

You must unpartition a partitioned table before you can delete it, create a clustered index on it, or place its new growth on a specified segment. Unpartitioning a table does not move data. When you unpartition a table, SQL Server concatenates the chains of the table's data pages to produce a single page chain. To remove partitions from a table:

1. Display the **Table Properties** tab, as described in "Displaying Table Properties" on page 10-11.
2. Click **Partitions**. The display changes to the **Partitions** tab.

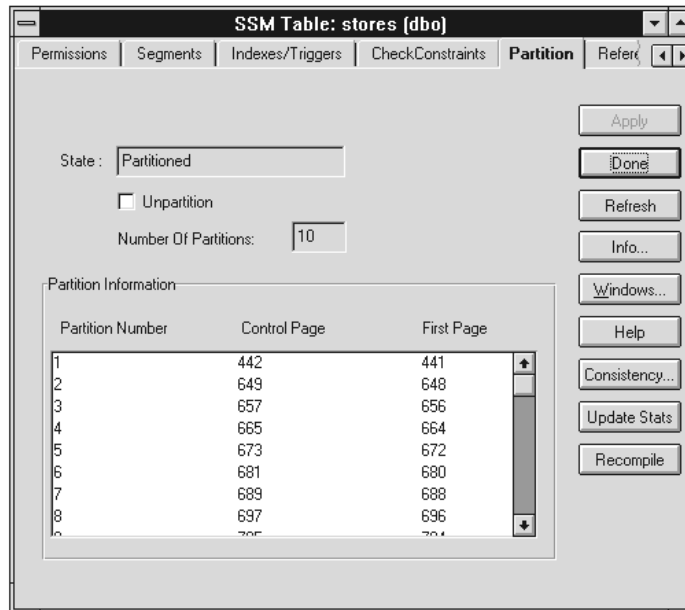


Figure 10-11: Table Partitions tab

3. Select the **Unpartition** check box.
4. Click **Apply**.

Creating Referential Constraints on a Table

Referential constraints allow you to ensure that data inserted in one table has matching values in another table. For example, in the *pubs2* database, a row inserted into the *salesdetail* table, recording the sale of books, must have a valid *title_id* in the *titles* table. Currently, *pubs2* enforces this referential integrity with a trigger. However, a referential constraint on the *salesdetail* table, referencing the *title_id* column in the *titles* table, could accomplish the same enforcement of referential integrity.

A referential constraint involves two tables:

- A **referencing** table on which data insertions are performed. This table contains one or more columns that serve as a foreign key.
- A **referenced** table against which insertions are checked. This table contains one or more columns that serve as a primary key.

When an insertion occurs on the referencing table, the value of the foreign key columns must match the value of the primary key columns on the referenced table. You can define a referential constraint on either the referenced or the referencing table.

Privileges	Only a database or table owner who has <code>references</code> permission on the referenced table can create a referential constraint.
------------	--

To create a referential constraint:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-11.
2. Click Referential Constraints. The display changes to the Referential Constraints tab.

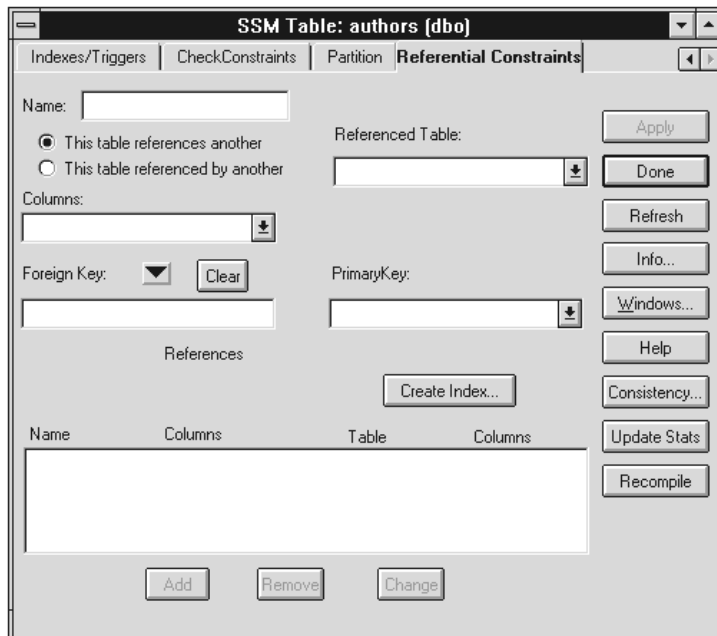


Figure 10-12: Table Referential Constraints tab

3. In the Name box, enter the referential constraint name. If you do not specify a name, SQL Server generates one.
4. If the current table is the referencing table, select This Table References Another, and follow the instructions in “This Table References Another” on page 10-32. If the current table is the referenced table, select This Table Referenced by Another, and follow the instructions in “This Table Referenced by Another” on page 10-33.

This Table References Another

When you select This Table References Another, the Referential Constraints tab has the controls shown in the following figure:

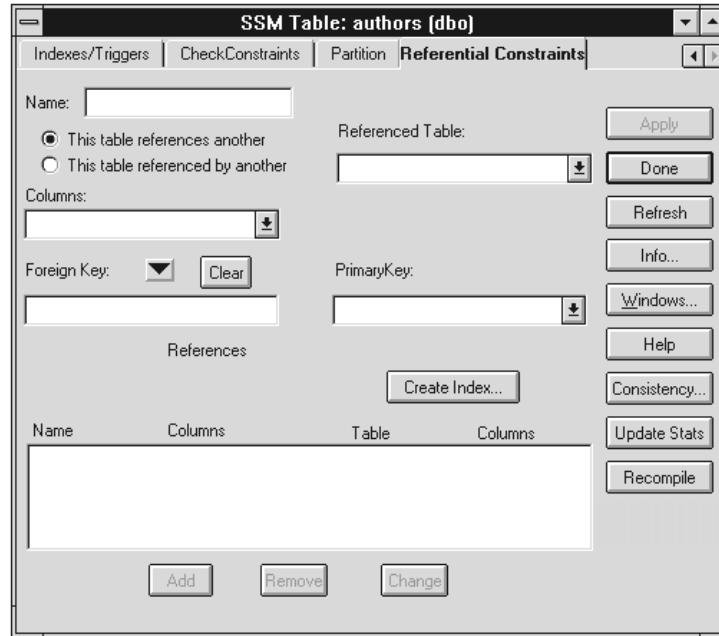


Figure 10-13:Referential Constraints tab for referencing table

To specify a constraint:

1. From the Referenced Table list, select the table against which inserts on the current table will be validated.
2. From the Columns list, select the columns in the current table that make up the foreign key:
 1. Select a column.
 2. Click the down arrow to add it to the Foreign Key box.
 3. Repeat until the Foreign Key box contains all necessary columns.

To clear the Foreign Key box, click Clear.

3. From the Primary Key list, which contains the primary keys defined for the referenced table, select the primary key that corresponds to the columns that make up the Foreign Key box.

Note that the datatypes of foreign key and primary key columns must match exactly.

To be included in the Primary Key list, a column must have a unique index defined on it. To create an index on one or more columns in the referenced table, click the Create Index... button.

4. To add the constraint definition to the list at the bottom of the dialog box, click Add. To remove a constraint from the list, click Remove. To change a constraint, select it in the list, update the specifications, and click Change.
5. Click Apply.

This Table Referenced by Another

When you select This Table Referenced by Another, the Referential Constraints tab has the controls shown in the following figure:

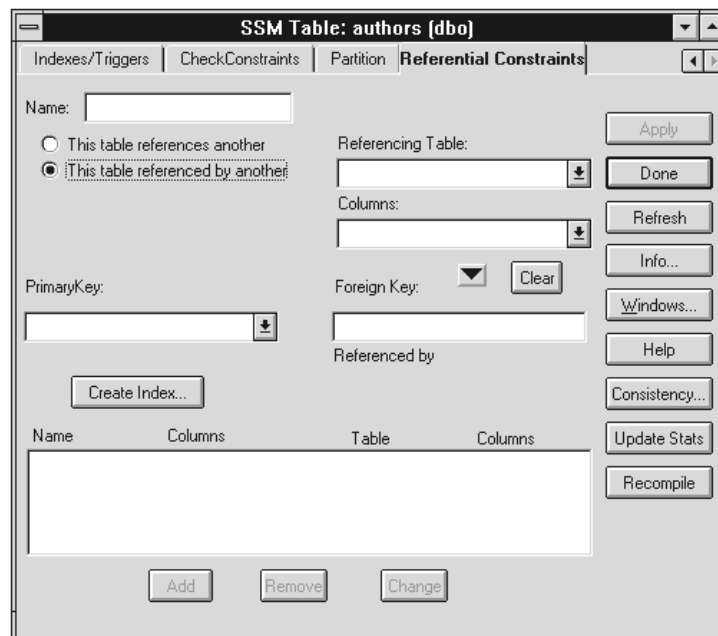


Figure 10-14: Referential Constraints tab for referenced table

To specify a constraint:

1. From the Referencing Table list, select the table for which SQL Server will validate inserts against the current table.
2. From the Columns list, select the columns in the referencing table that make up the foreign key:
 1. Select a column.
 2. Click the down arrow to add it to the Foreign Key box.
 3. Repeat until the Foreign Key box contains all necessary columns.To clear the Foreign Key box, click Clear.
3. From the Primary Key list, which contains the primary keys defined for the current table, select the primary key that corresponds to the columns that make up the Foreign Key box. Note that the datatypes of foreign key and primary key columns must exactly match.

To be included in the Primary Key list, a column must have a unique index defined on it. To create an index on one or more columns in the current table, click the Create Index... button.
4. To add the constraint definition to the list at the bottom of the dialog box, click Add. To remove a constraint from the list, click Remove. To change a constraint, select it in the list, update the specifications, and click Change.
5. Click Apply.

Recompiling Procedures and Triggers

After you define a table, you can recompile each procedure and trigger that uses the table. The recompilation is done the next time the procedures or triggers run. As you add indexes or change the database in ways that affect its statistics, compiled procedures and triggers may lose efficiency. Recompiling procedures and triggers can optimize efficiency. To recompile:

1. Select the icon of the table.
2. From the Table menu, choose Recompile. A message dialog box appears when the operation that sets up the recompilation is complete.
3. Click OK.

Shortcut

From any Table dialog tab, select the Recompile button.

**Checking Table Consistency**

When you add or delete rows or tables, SQL Server allocates or deallocates data pages and extents (blocks of data pages) to accommodate the change. The Database Consistency Checker (dbcc) is a set of utility commands for checking the record of this activity and the logical and physical consistency of a database. The dbcc command has several options to check table consistency.

This section gives general instructions for executing the dbcc command for table consistency and then describes each command option.

Instructions

To check table consistency:

1. Select the icon of the table to check.
2. From the Table menu, choose Consistency... The Table Consistency Check dialog box opens.

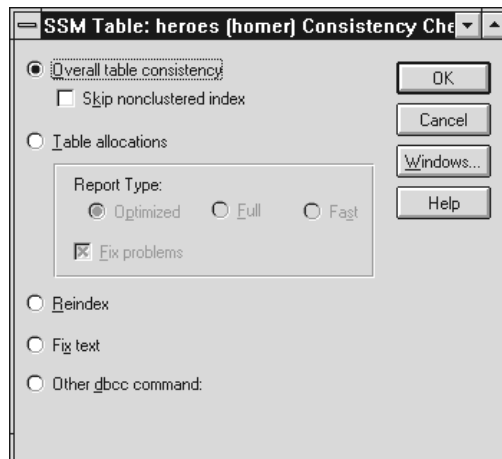


Figure 10-15: Table Consistency dialog box

3. Select the option button of the type of consistency check to perform and enter any other specifications that apply to that type. For consistency check details, see “Consistency Check Options” on page 10-36.
4. Click OK. SQL Server executes the `dbcc` command. When it completes, SQL Server Manager displays the results in the Output dialog box. You can copy and paste the output into a word processor or text editor.

The command option descriptions that follow include sample report output.

Shortcuts

From any Table dialog tab, select the Consistency... button.



Click the right mouse button over the icon of the table; then, choose Consistency... from the shortcut menu.



Consistency Check Options

The following consistency check options are available:

- Overall Table Consistency
- Table Allocations
- Reindex
- Fix Text
- Other dbcc Command

Overall Table Consistency Option

This option corresponds to the `dbcc checktable` command. This option checks that index and data pages are correctly linked, indexes are properly sorted, all pointers are consistent, and the data rows on each page have entries in the first page of an allocation map. The entries match their respective locations on the page.

To skip checking nonclustered indexes on user tables, select the Skip Nonclustered Index box. If you leave the box unselected, the `dbcc` command checks all indexes on all tables.

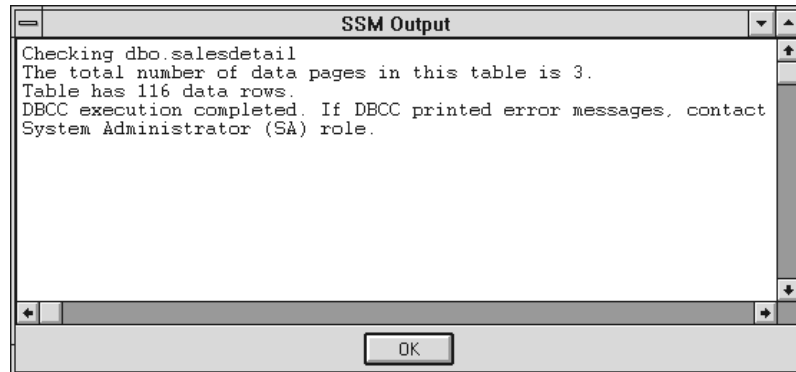


Figure 10-16: Table consistency check output—overall consistency option

Table Allocations Option

This option corresponds to the `dbcc tablealloc` command. This option checks that index and data pages are correctly allocated, that no page is allocated that is not used, and that no page is used that is not allocated. The following report types are available:

- To report allocation pages listed in the Object Allocation Map pages for the table, select `Optimized`.
- To report all types of allocation errors, select `Full`.
- To report pages that are referenced but not allocated in the extent, select `Fast`.

By default, `dbcc tablealloc` fixes allocation errors as it executes. SQL Server Manager automatically places the database in single-user mode while executing `dbcc tablealloc` and then returns the database to multiple-user mode when processing is complete. To generate a report without fixing allocation errors, clear the Fix Problems box.

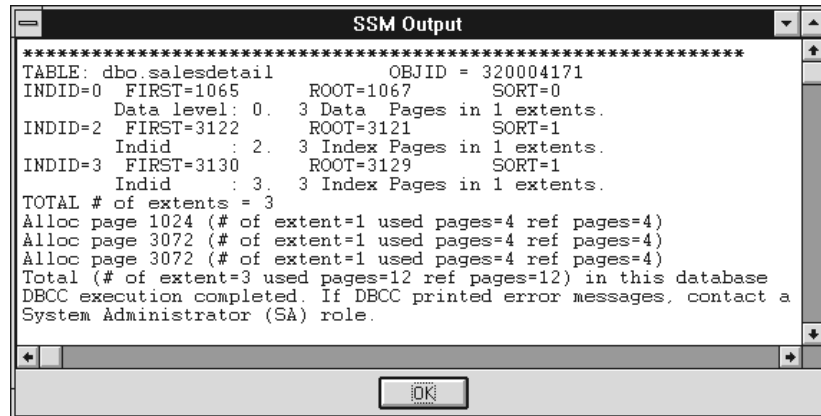


Figure 10-17: Table consistency check output—allocations option

Reindex Option

This option corresponds to the `dbcc reindex` command. This option checks the integrity of indexes on user tables. It drops and rebuilds indexes it suspects are corrupt.

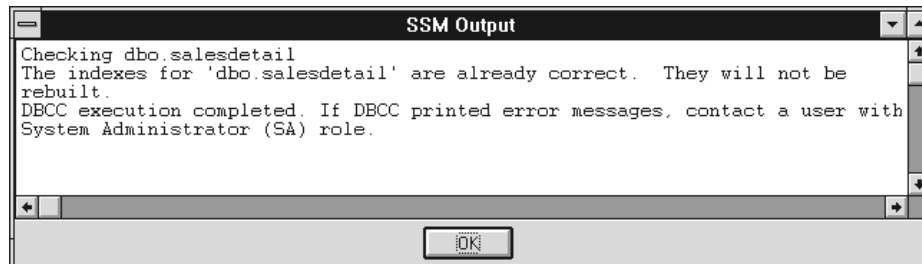


Figure 10-18: Table consistency check output—reindex option

Fix Text Option

This option corresponds to the `dbcc fix_text` command. This option upgrades *text* values after a SQL Server character set has been changed to a multibyte character set.



Figure 10-19: Table consistency check output—fix text option

Other dbcc Command Option

This option allows you to enter the text of other `dbcc` command options. When you select Other `dbcc` Command, an edit box becomes visible below the option button. Enter the text of the entire `dbcc` command option to execute, including the `dbcc` keyword.

Updating Statistics

When you update statistics, SQL Server updates information about the distribution of key values in indexes associated with a table. Update statistics if a large amount of data in an indexed column has been added, changed, or removed. Update statistics after adding new rows to a table whose rows have been deleted with `truncate table`.

To update statistics:

1. Select the icon of the table to update.
2. From the Table menu, choose Update Statistics...

Shortcuts

From any Table dialog tab, select the Update Stats button.



Click the right mouse button over the table icon; then, choose Update Statistics from the shortcut menu.



► **Note**

After running **update statistics**, recompile the table for which the index is defined. This ensures that the procedures and triggers that use the index will use the new key distribution.

11

Using System Management Tools

What's in This Chapter

SQL Server Manager provides access to specialized system management tools. These tools, and the tasks they help to perform, are:

- **sqledit**—manages *sql.ini* file entries
- **wisql**—provides an interface for creating, editing, and executing SQL queries

This chapter describes how to use each tool.

Online help is available for **sqledit** and **wisql**. To access help for these tools, choose the following commands from the SQL Server Manager Help menu, or access help from within the tool:

Table 11-1: Commands for online help on system management utilities

Utility	SQL Server Manager Help menu command	To access help from within the tool
sqledit	Help on Interfaces...	Choose Help on SQLEEDIT from the Help menu
WISQL	Help on ISQL...	N/A

Editing the *sql.ini* File with **sqledit**

A **sql.ini** file must be available on each machine from which clients connect to SQL Servers. Each *sql.ini* file entry tells a client or host machine how to connect to a specific SQL Server. The *sql.ini* file performs the same function in a Windows or NT environment as the **interfaces file** in a UNIX or VMS environment.

When you connect to a SQL Server in SQL Server Manager, the application looks for this file in the *ini* subdirectory of the root directory of the Sybase installation, referenced by the SYBASE environment variable.

When you install SQL Server Manager, the installation program creates a *sql.ini* file containing an entry for the SQL Server you designate as your primary server. To specify additional SQL Servers or to change connection information for an existing entry, you must edit your *sql.ini* file.

To help you create and edit *sql.ini* files, SQL Server Manager gives you access to the *sqledit* utility. This section describes how to perform the following tasks with *sqledit*:

- Starting and exiting *sqledit*
- Creating a new *sql.ini* file
- Updating an existing *sql.ini* file
- Updating platform and Net-Library driver information
- Testing whether a SQL Server is running

Additional information is available in the *sqledit* online help. To view online help, choose Help on SQLEDIT from the SQL Server Manager or *sqledit* Help menu.

Starting and Stopping *sqledit*

To start *sqledit*, choose Interfaces... from the Tools menu. The SQLEDIT window opens, displaying the information in the *sql.ini* file located in the *ini* subdirectory of the Sybase root directory.

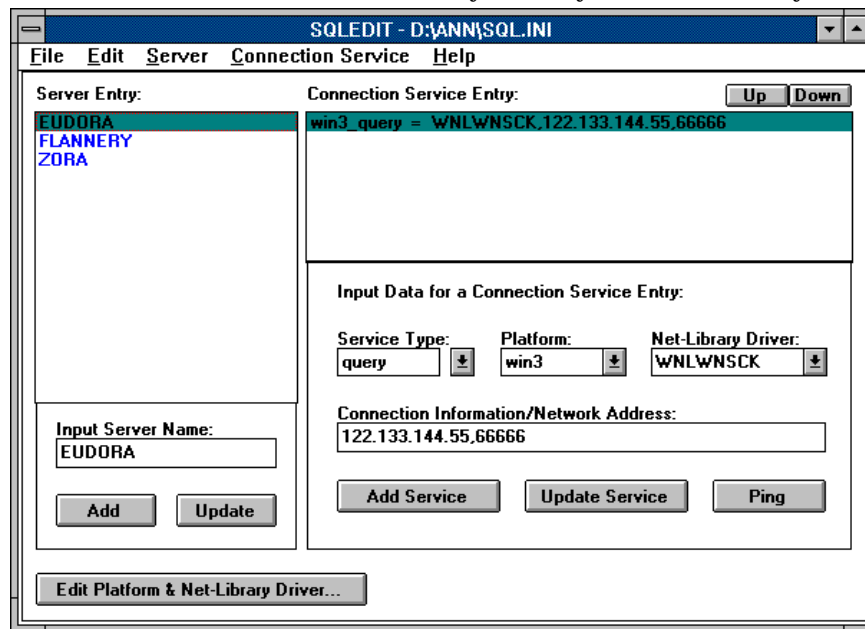


Figure 11-1: SQLEDIT window

Shortcut

Double-click the *sqledit* icon on the desktop.



To stop *sqledit*, choose Exit from the File menu in the SQLEEDIT window. If you have made changes to the current *sql.ini* file, *sqledit* prompts you to save them before exiting.

Creating a New *sql.ini* File

To create a new *sql.ini* file:

1. To clear the currently displayed *sql.ini* file from the window, choose New from the File menu.
2. Create entries consisting of a SQL Server name and the appropriate connection service information. For information on creating file entries, see “Creating a *sql.ini* File Entry.”
3. To save the file, choose Save or Save As... from the File menu. Specify drive, directory, and file name, and choose OK.

Creating a *sql.ini* File Entry

To connect to a SQL Server not already specified in your *sql.ini* file, you must create a new *sql.ini* file entry. This entry consists of two parts:

- Name of the SQL Server
- Connection service information

► **Note**

For convenience in creating repetitive entries, use the Cut, Copy, and Paste commands in the Edit menu. To undo an action, choose Undo from the Edit menu.

Adding a New SQL Server Name

1. From the Server menu, choose New Server Entry.
2. Enter the name of the SQL Server to add in the Input Server Name box.
3. From the Server menu, choose Add Server Entry.

Shortcut



Clear the entry in the Input Server Name box by selecting and deleting it. After entering the name of the new SQL Server, click Add.

Adding New Connection Service Information

Connection service information consists of the service type, platform, Net-Library driver name, and address information required by the Net-Library driver you use.

To specify new connection service information:

1. From the Connection Service menu, choose New Service.
2. Select or enter the following information in the Input Data area of the window:
 - From the Service Type list, select Query.
 - From the Platform list, select Win3 (for Microsoft Windows) or NT (for Windows NT).
 - From the Net-Library Driver list, select the appropriate Net-Library driver.

► ***Note***

If the platform and driver combination for your installation are not reflected in the drop-down lists, you may need to update the lists. For information, see “Editing Platform and Net-Library™ Driver Information” on page 11-9.

- In the Connection Information/Network Address box, enter the connection information required by the Net-Library driver selected in the Net-Library Driver list. The format of this information depends on the driver; for example, the connection information for the Windows Sockets driver, WNLWNSCK, is in the format *network_address, port_number*.

For information on the format of Net-Library connection information, see “Connection Format Examples” on page 11-7.

3. From the Connection Service menu, choose Add Service.

Shortcut



After entering all required connection information, click Add Service.

Updating an Existing sql.ini File

To update an existing *sql.ini* file:

1. From the File menu, choose Open... Specify drive, directory, and file name of the file to edit, and choose OK.
2. Modify or delete SQL Server names and connection service information, as appropriate. For information on modifying file entries, see “Modifying a sql.ini File Entry.”
3. To save the file, choose Save from the File menu. To save the file under a new name, choose Save As... from the File menu. Specify drive, directory, and file name, and choose OK.

Modifying a sql.ini File Entry

With sqledit, you can modify or delete SQL Server names and connection service information.

► **Note**

For convenience in creating repetitive entries, use the Cut, Copy, and Paste commands in the Edit menu. To undo an action, choose Undo from the Edit menu.

Modifying a SQL Server Name

To modify a SQL Server name:

1. In the Server Entry list, select the SQL Server name to update.
2. Make the appropriate changes in the Input Server Name box.

3. From the Server menu, choose Update Server Entry.

Shortcut

After updating the SQL Server name, click Add.



Deleting a SQL Server Name

To delete a SQL Server name:

1. In the Server Entry list, select the SQL Server name to delete.
2. From the Server menu, choose Delete Server Entry.

Updating Existing Connection Service Information

To update existing connection service information:

1. In the Server Entry list, select the SQL Server for which to update a connection service.
2. Select or enter the following information in the Input Data area of the window:
 - Select Query from the Service Type list.
 - Select Win3 (for Microsoft Windows) or NT (for Windows NT) from the Platform list.
 - Select the appropriate Net-Library driver from the Net-Library Driver list.

► Note

If the platform and driver combination for your installation are not reflected in the drop-down lists, you may need to edit the lists. For information, see “Editing Platform and Net-Library™ Driver Information” on page 11-9.

- In the Connection Information/Network Address box, enter the connection information required by the Net-Library driver selected in the Net-Library Driver list. The format of this information depends on the driver; for example, the connection information for the Windows Sockets driver, WNLWNSCK, is in the format *network_address, port_number*.

For information on the format of Net-Library connection information, see “Connection Format Examples” on page 11-7.

3. From the Connection Service menu, choose Update Service.

Shortcut



After entering all required connection information, click Update Service.

Deleting Connection Service Information

To delete connection service information:

1. In the Server Entry list, select the SQL Server for which to delete the connection service.
2. From the Connection Service menu, choose Delete Service.

Connection Format Examples

The format of the address information entered in the Connection Information/Network Address box depends on the Net-Library driver you use. This section provides examples for the most common Net-Library drivers supported by SQL Server Manager. For more information, see the *Open Client Net-Library Installation and Reference Manual*.

Windows Sockets Driver

1. Windows example:

```
[SYBASE]
WIN3_QUERY=WNLWNSCK,122.223.334.0,8888
```

If a client using this *sql.ini* file entry requests a connection to the SYBASE server, it connects to the server listening at port 8888 running on a TCP/IP network on the machine whose network IP address is 122.223.334.0.

2. Windows NT and Windows 95 example:

```
[SYBASE]
QUERY=NLWNSCK,VIRGIL,8888
```

If a client using this *sql.ini* file entry requests a connection to the SYBASE server, it connects to the server listening at port 8888

running on a TCP/IP network on a host machine named VIRGIL. Note that for NT, you must supply the host name, not its network IP address.

SPX/IPX Driver

1. Windows example:

```
[CONNECT50]
WIN3_QUERY=WNLNOVSP,GUMMO
```

If a client requests a connection to the CONNECT50 server, it connects to the server on a NetWare network that has an advertised server name of GUMMO. If this connection were to a SQL Server that did not use bindery services, you would need to supply the following information to build a connection string:

- Network number of the server host
- Node number
- Port number or SPX socket number

The SPX/IPX example would look like this:

```
[CONNECT50]
WIN3_QUERY=WNLNOVSP,2DA405E7,1,83bd
```

where 2DA405E7 is the network number, 1 is the node number, and 83bd is the SPX socket number.

2. Windows NT and Windows 95 example:

```
[CONNECT50]
QUERY=NLNWLINK,GUMMO
```

If a client requests a connection to the CONNECT50 server, it connects to the server on a NetWare network that has an advertised server name of GUMMO. If this connection were to a SQL Server that did not use bindery services, you would need to supply the following information to build a connection string:

- Network number of the server host
- Node number
- Port number or SPX socket number

The SPX/IPX example would look like this:

```
[CONNECT50]
QUERY=NLNWLINK,2DA405E7,1,83bd
```

where 2DA405E7 is the network number, 1 is the node number, and 83bd is the SPX socket number.

Named Pipes Driver

1. Windows example:

```
[SQLSERVER]  
WIN3_QUERY=WNLNMP, \\GROUCHO\PIPE\SQL10\QUERY
```

If a client requests a connection to the SQLSERVER server, it connects to the server on a Named Pipes network that uses the pipe named `\\groucho\pipe\sql10\query`. This example illustrates how the SQL Server host name, GROUCHO, is prepended to the default pipe name `\pipe\sql10\query` with a double-backslash prefix.

2. Windows NT example:

```
[SQLSERVER]  
QUERY=NLMSNMP, \\GROUCHO\PIPE\SQL10\QUERY
```

If a client requests a connection to the SQLSERVER server, it connects to the server on a Named Pipes network that uses the pipe named `\\groucho\pipe\sql10\query`. This example illustrates how the SQL Server host name, GROUCHO, is prepended to the default pipe name `\pipe\sql10\query` with a double-backslash prefix.

Editing Platform and Net-Library™ Driver Information

The Platform and Net-Library Driver lists in the Connection Service part of the SQLEDT window contain the platforms and Net-Library drivers recognized by the Open Client™ library. When Open Client changes to recognize new platforms or drivers or to discontinue old ones, you can edit the lists in sqledit accordingly, by adding or deleting platforms or Net-Library drivers. New platforms or drivers become available for selection in the Connection Service Entry part of the SQLEDT window; deleted ones are removed from the platform or driver list.

To edit platform and Net-Library information, click the Edit Platform & Net-Library Driver... button in the SQLEDT window. The following dialog box opens:

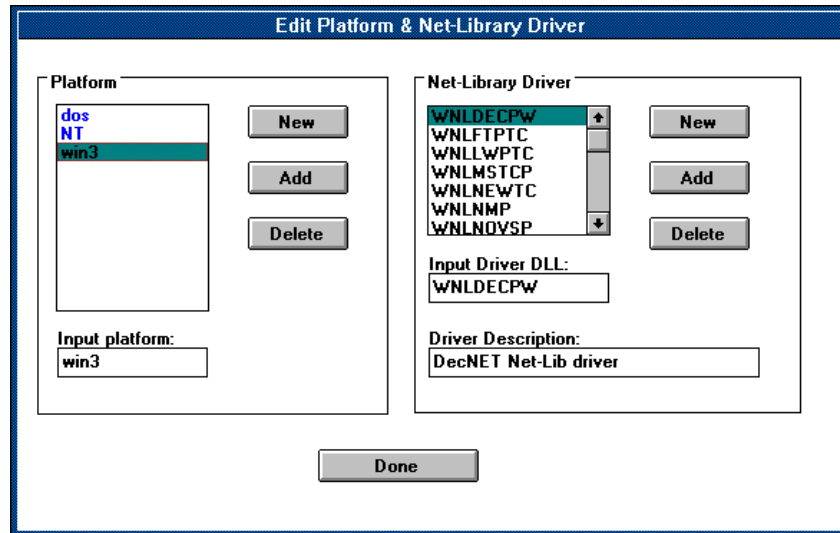


Figure 11-2: Edit Platform & Net-Library Driver dialog box

To return to the SQLEDIT window, click Done.

Editing the Platform List

To add a new platform name:

1. Click New.
2. Enter the new platform name in the Input Platform box.
3. Click Add.

To delete a platform name:

1. Select the name to delete.
2. Click Delete. A confirmation dialog box opens.

◆ **WARNING!**

When you delete a platform name, you delete all Net-Library Driver names associated with it. Also, you delete any Connection Service entries associated with those drivers.

3. Click Yes to confirm the deletion, or click No to cancel it.

Editing the Net-Library Driver List

To add a new driver name:

1. Click New.
2. Enter the new driver name in the Input Driver DLL box.
3. Enter a description for the driver in the Driver Description box.
4. Click Add.

To delete a driver name.

1. Select the name to delete.
2. Click Delete. A confirmation dialog box opens.

◆ **WARNING!**

When you delete a Net-Library driver name, you also delete any Connection Service entries associated with the driver.

3. Click Yes to confirm the deletion, or click No to cancel it.

Testing Whether a SQL Server is Running

To check whether a SQL Server in your *sql.ini* file is up and running:

1. In the Server Entry list, select the SQL Server to test.
2. Click Ping. sqledit issues a ping command to the SQL Server and reports the results in an Information dialog box. The following example shows the results of a test of the SOL_ENG11 SQL Server:

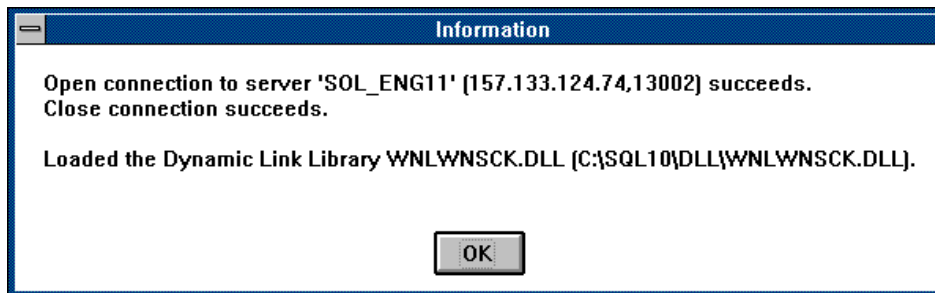


Figure 11-3: Results of pinging the SOL_ENG11 SQL Server

Querying SQL Server with wisql

SQL Server Manager gives you the option of communicating directly with SQL Server by using the `wisql` utility.

`wisql` is a utility that allows you to create and execute SQL queries against a SQL Server database. This section describes how to:

- Start and stop `wisql`
- Connect to SQL Server
- Manage query files
- Cut and paste query text
- Execute queries

Starting and Stopping `wisql`

To start the application, choose `ISQL...` from the Tools menu. The `WISQL32` window opens.

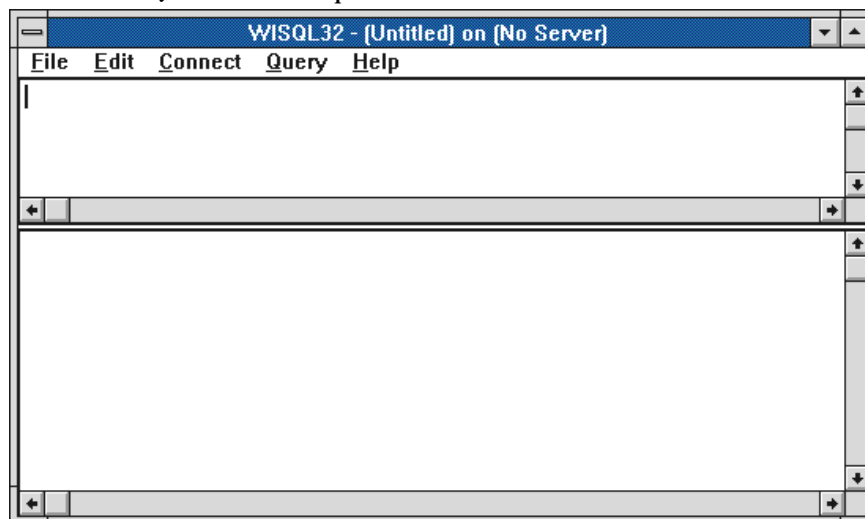


Figure 11-4: WISQL window

This window consists of two sections:

- The top section is the workspace, the area where you enter Transact-SQL commands.

- The bottom section is the results section, the area where SQL Server returns the results of queries.

Both sections scroll horizontally and vertically. You can enter as much as 62K of statements in the top section and display as much as 64K of results in the bottom section.

To stop wisql, choose Exit from the File menu.

◆ **WARNING!**

wisql does not prompt you to save changes in your workspace. Unless you save your query file by using the appropriate Save command on the File menu before choosing the Exit command, you lose any changes you made since your last save.

Setting wisql Run Options

The File menu also includes commands for setting wisql run options and for exiting wisql.

To set run options:

1. Choose Options... from the File menu. The WISQL Options dialog box opens.

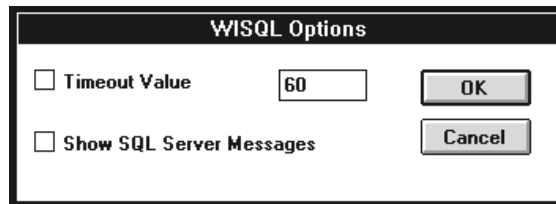


Figure 11-5: WISQL Options dialog box

2. Select the options you want by selecting the corresponding check box:
 - Timeout Value—sets the length of time to wait for a response from SQL Server before cancelling a query. The default value is 60 seconds; to reset the value, enter the new value in the edit box.
 - Show SQL Server Messages—specifies whether to display messages from SQL Server.
3. Click OK.

Managing SQL Server Connections

Use the commands on the Connection menu to connect to, disconnect from, and switch between connections with SQL Server.

Connecting

When you start wisql, you are connected to the SQL Server to which you were connected in SQL Server Manager.

To establish a connection to a different SQL Server:

1. From the Connect menu, choose Open Connection... The Open Connection dialog box opens. The fields are filled with current SQL Server information.

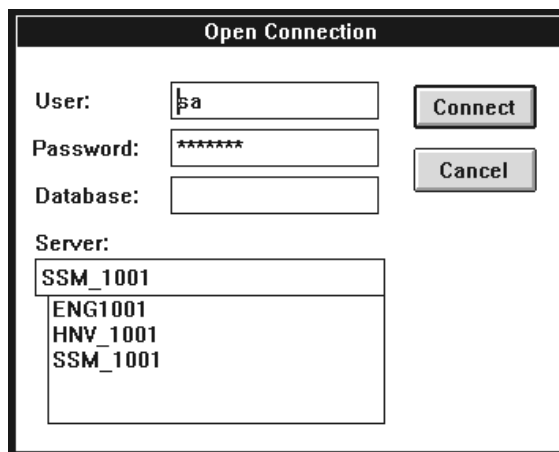


Figure 11-6: Open Connection dialog box

2. Change any of the following information, if necessary:
 - In User, enter the user name which you want to connect to SQL Server.
 - In Password, enter the password for that user.
 - In Database, enter the name of the database to use. The default is the user's default database.
 - Select the name of the SQL Server from the list.
3. Click Connect. You may receive messages about the character set, database context, and language settings, along with a request for confirmation.

4. If the confirmation dialog box appears, click OK. You can now enter SQL statements in the workspace of the WISQL window.

Switching Between Connections

You can be connected to more than one SQL Server or database at a time. This feature is useful for running the same query against multiple databases.

To switch between connections, select the connection that you want to activate from the list of connections at the bottom of the Connect menu. The current active connection has a check mark.

Disconnecting

To disconnect from the active connection:

1. From the Connect menu, choose Disconnect. A confirmation dialog box opens.
2. Click Yes.

Creating and Modifying Queries

wisql enables you to create, modify, and store query text.

Creating a New Query

To create a new query:

1. From the File menu, choose New. The workspace clears of any existing data.
2. Enter the text of the query in the query workspace.

► **Note**

For convenience in editing query text, use the Cut, Copy, Paste, and Delete commands in the Edit menu. To undo an action, choose Undo from the Edit menu.

3. From the File menu, choose Save Input As...
4. In the Save As dialog box, specify the drive, directory, and file name under which to save the query.
5. Click OK.

Modifying an Existing Query

To modify an existing query:

1. From the File menu, choose Open...
2. In the Open dialog, specify the drive, directory, and file name to retrieve, and click OK.
3. Modify the text of the query as necessary in the query workspace.

► **Note**

For convenience in editing query text, use the Cut, Copy, Paste, and Delete commands in the Edit menu. To undo an action, choose Undo from the Edit menu.

4. From the File menu, choose Save Input. To save the query under a different name, choose Save Input As... Specify the drive, directory, and file name, and click OK.

Executing Queries

To execute all or part of the text in the wisql workspace, use the commands in the Query menu:

- To execute all statements in the workspace, choose Execute All.
- To execute all statements from the current cursor location to the end of the workspace, choose Execute to End.
- To execute the statements from the current cursor location to the next go statement, choose Execute Next.
- To execute the currently selected workspace text, choose Execute Selection.

Saving Query Results

To save the results of a query:

1. From the File menu, choose Save Output...
2. In the Save As dialog box, specify the drive, directory, and file name, and click OK. The output is saved as a text file with an extension of *.out*.

A

SQL Server Configuration Parameters

What's in This Appendix

This appendix contains information about SQL Server configuration parameters. Information is in the following forms:

- An alphabetical list of SQL Server 10.x parameters, along with a brief description of each.
- An alphabetical list of SQL Server 11.0 parameters, along with a brief description of each.
- A list of SQL Server 11.0 parameters grouped according to the area of SQL Server behavior they affect.

For more detailed information about each configuration parameter, see the *System Administration Guide*.

SQL Server 10.x Parameter List

The following is a list of SQL Server 10.x configuration parameters:

- **additional netmem** specifies additional network memory.
- **allow updates** is a code indicating whether or not users with appropriate permissions can update system tables. Values are:
 - 0 (default)—users cannot update system tables.
 - 1—users can update system tables directly.In general, allowing updates is not recommended.
- **audit queue size** specifies the number of audit records that the audit queue can hold.
- **cpu flush** determines the number of machine clock ticks to accumulate before adding usage statistics to the *syslogins* table, for chargeback accounting.
- **database size** determines the default number of megabytes allocated with a **create database** statement.
- **default language** is the number of the default language used to display system messages.
- **default network packet** specifies the default network packet size.
- **devices** specifies the number of virtual device numbers that SQL Server can use.

- **engine adjust interval** is not currently used.
- **extent i/o buffers** specifies the number of extents (each consisting of eight data pages) to allocate for use by the `create index` statement.
- **fill factor** is a percentage that determines how full SQL Server makes each page when it is creating a new index on existing data (unless a value is specified in the `create index` statement).
- **i/o flush** specifies the number of read or write I/Os to accumulate before flushing the data to the `syslogins` table, for chargeback accounting.
- **identity burning set factor** is a blocking factor that determines how many potential `IDENTITY` column values become available when SQL Server allocates a new `IDENTITY` column block in memory.
- **language in cache** determines the maximum number of languages that can be held simultaneously in the language cache.
- **locks** specifies the maximum number of locks available.
- **max online engines** is a number that controls the maximum number of engines in a symmetric multiprocessor environment.
- **maximum network packet** specifies the maximum network packet size.
- **memory** specifies the amount of memory (in 2K units) available to SQL Server.
- **min online engines** is not currently used.
- **nested triggers** is a code indicating whether or not nested triggers (that is, triggers that set off other triggers) are allowed. Values are:
 - 0—nested triggers are not allowed; an error is generated if a trigger tries to set another off.
 - 1 (default)—nested triggers are allowed.
- **open databases** determines the maximum number of databases that can be open at one time.
- **open objects** determines the maximum number of database objects that can be open at one time.
- **password expiration interval** specifies the number of days that passwords remain in effect after they are changed. The default is 0 (passwords do not expire).
- **pre-read packets** specifies the number of packets that a site handler pre-reads in connections with remote servers. This is set to default values when Remote Access is set to 1.

- **procedure cache** specifies the percentage of memory allocated to the procedure cache after SQL Server's memory needs are met. The procedure cache is the area of memory where the most recently used procedures are stored.
- **recovery flags** is a code that determines what information SQL Server displays on the console during recovery. Values are:
 - 0 (default)—only the database name and a brief message are displayed.
 - 1—information about each individual transaction is displayed.
- **recovery interval** determines the maximum number of minutes per database that SQL Server uses to complete the recovery procedures in case of a system failure.
- **remote access** is a code indicating whether or not users from remote servers can log into this server. Values are:
 - 0—users from remote servers cannot log in.
 - 1 (default)—users from remote servers can log in.
- **remote connections** determines the maximum number of active connections between this SQL Server and other servers. This is set to default values when Remote Access is set to 1.
- **remote logins** determines the maximum number of active user connections between the local SQL Server and remote servers. This is set to default values when Remote Access is set to 1.
- **remote sites** determines the number of remote sites that can simultaneously access the local SQL Server. This is set to default values when Remote Access is set to 1.
- **stack size** specifies the size of the SQL Server stack.
- **tape retention** specifies the number of days to retain each tape after it has been used for a database or transaction log dump. If you try to use the tape before that number of days has passed, SQL Server issues a warning message.
- **time slice** specifies the number of milliseconds that a user process is allowed to run by the scheduler. There is seldom any reason to change this parameter.
- **upgrade version** specifies the current SQL Server upgrade version. This variable is automatically changed by the upgrade program provided with new releases.
- **user connections** specifies the maximum number of user connections allowed simultaneously by SQL Server. If this

parameter is incorrectly configured, some SQL processes may fail.



SQL Server 11.0 Parameter List

The following is a list of SQL Server 11.0 configuration parameters:

- **additional network memory** allocates additional memory for clients which request packet sizes that are larger than the default packet size for the server.
- **allow nested triggers** is a toggle that controls the use of nested triggers. When the value is set to 1, data modifications made by triggers can fire other triggers.
- **address lock spinlock ratio** specifies the number of rows in the address locks hash table protected by one spinlock (rows per spinlock).
- **allow remote access** determines whether users from remote servers can access this SQL Server. The default is 1, to allow SQL Server to communicate with Backup Server.
- **allow sql server async i/o** is a toggle that enables SQL Server to run with asynchronous disk I/O.
- **allow updates to system tables** allows system tables to be updated directly. The default is 0 (off).
- **audit queue size** determines the number of audit records that the audit queue can hold. The default is 100.
- **configuration file** specifies the location of the configuration file you want to use.
- **cpu accounting flush interval** specifies how many machine clock ticks to accumulate before adding cpu usage data to *syslogins* for use in chargeback accounting statistics.
- **cpu grace time** specifies the maximum amount of time (in milliseconds) a user process can run without yielding the CPU before SQL Server infects it.
- **deadlock checking period** specifies the minimum amount of time (in milliseconds) a process must wait for a lock before SQL Server initiates a deadlock check.
- **deadlock retries** specifies the number of times a transaction will retry to acquire a lock after it has become a deadlock victim.
- **default character set id** is the number of the default character set used by the server.

- **default database size** sets the default number of megabytes allocated to each new user database. The default run value is 2 (megabytes).
- **default fill factor percent** determines how full SQL Server makes each page when it is creating a new index on existing data (unless the user specifies some other value in the `create index` statement). The default run value is 0.
- **default language id** is the number of the language that is used to display system messages unless a user has chosen another language from those available on the server.
- **default network packet size** sets the default size of network packets for all users on SQL Server.
- **default sortorder id** is the number of the sort order that is the current default on this SQL Server. **Do not change this parameter.** See the *System Administration Guide* for more information about changing the sort order.
- **deadlock checking period** specifies the minimum amount of time (in milliseconds) before SQL Server initiates a deadlock check for a process waiting on a lock to be released.
- **disk i/o structures** specifies the initial number of disk I/O control blocks SQL Server allocates on start-up.
- **engine adjust interval** is not currently used.
- **event buffers per engine** specifies the number of events per SQL Server engine that can be simultaneously monitored. Events are used in conjunction with Monitor Server and a client tool for observing SQL Server performance.
- **executable code size** reports the size of the SQL Server executable.
- **freelock transfer block size** specifies the number of locks moved between the engine freelock cache and the global freelock list.
- **housekeeper free write percent** determines the maximum percentage by which database writes can increase as a result of free writes initiated by the housekeeper process during the server's idle cycles. Values can range from 0 through 100.
- Setting this parameter to 0 disables the housekeeper process. Setting it to 100 allows the housekeeper process to work continuously during the server's idle cycles. The default value, 10, allows the housekeeper process to continue moving buffers into the buffer wash region during the server's idle cycles as long as database writes do not increase by more than 10%.

- **io accounting flush interval** specifies how many disk I/Os to accumulate before flushing the data to *syslogins* for use in chargeback accounting.
- **io polling process count** specifies the number of tasks the scheduler will run before checking for disk and/or network I/O completions.
- **identity burning set factor** determines the percentage of potential IDENTITY column values that is made available in each block. The default value, 5000, releases .05 percent of the potential IDENTITY column values for use at a time.
- **identity grab size** allows each SQL Server process to reserve a block of IDENTITY column values for inserts into tables that have an IDENTITY column.
- **lock shared memory** disallows swapping of SQL Server pages to disk, and allowing the operating system kernel to avoid the server's internal page locking code.
- **lock promotion HWM** sets the maximum number of page locks allowed before SQL Server escalates to a table lock. The default value is 200.
- **lock promotion LWM** sets the minimum number of page locks allowed before SQL Server escalates to a table lock. The default value is 200.
- **lock promotion PCT** sets the percentage of page locks allowed before SQL Server escalates to a table lock. The default value is 100.
- **max async i/o's per engine** specifies the maximum number of asynchronous disk I/O requests that can be outstanding for a single engine at one time.
- **max async i/o's per server** specifies the maximum number of asynchronous disk I/O requests that can be outstanding for SQL Server at one time.
- **max engine freelocks** specifies the maximum number of locks available in an engine freelock cache.
- **max online engines** controls the number of engines in a symmetric multiprocessor environment.
- **max network packet size** sets the maximum network packet size that a client program can request.
- **max number of network listeners** specifies the maximum number of network listeners that can be open at one time.

- **memory alignment boundary** determines on which boundary buffer caches are aligned.
- **min online engines** is not currently used.
- **number of alarms** specifies the number of alarms allocated by SQL Server. Alarms are used with the Transact-SQL `waitfor` command.
- **number of devices** controls the number of database devices SQL Server can use. It does not include devices used for database dumps.
- **number of extent i/o buffers** allocates the specified number of extents (8 data pages) for use by `create index`. Do not set this value to more than 100.
- **number of index trips** specifies the number of times an aged index page recycles itself onto the MRU chain.
- **number of languages in cache** is the maximum number of languages that can simultaneously be held in the language cache. The default is 3.
- **number of locks** sets the number of available locks. The default run value is 5000.
- **number of mailboxes** specifies the number of mailbox structures SQL Server allocates on start-up. Mailboxes are use for process-to-process communication and synchronization.
- **number of messages** specifies the number of message structures allocated by SQL Server at start-up time. Messages are used in conjunction with mailboxes for process-to-process communication and synchronization.
- **number of oam trips** specifies the number of times an aged Object Allocation Map (OAM) page recycles itself onto the MRU chain.
- **number of open databases** sets the maximum number of databases that can be open at one time on SQL Server. The default run value is 12.
- **number of open objects** sets the maximum number of database objects that can be open at one time on SQL Server. The default run value is 500.
- **number of preallocated extents** specifies the number of extent structures allocated in a single trip to the page manager.
- **number of remote connections** controls the limit on active connections initiated to and from this SQL Server. The default is 20.

- **number of remote logins** controls the number of active user connections from this SQL Server to remote servers. The default is 20.
- **number of remote sites** controls the number of simultaneous remote sites that can access this SQL Server. The default is 10.
- **number of sort buffers** specifies the number of buffers used to hold pages read from input tables.
- **number of user connections** sets the maximum number of user connections that can be connected to SQL Server at the same time. The maximum value for your system is stored in the global variable @@max_connections, and varies according to platform and operating system.
- **page lock spinlock ratio** specifies the ratio of **spinlocks** protecting the internal page locks hash table.
- **page utilization** controls when SQL Server performs an OAM scan to find unused pages. The default run value is 95.
- **partition groups** specifies how many partition groups to allocate for the server. Partition groups are internal structures that SQL Server uses to control access to individual partitions of a table. SQL Server allocates partition groups to a table when you partition the table or when you access it for the first time after restarting the server.
- A partition group is composed of 16 partition caches, each of which stores information about a single partition. All caches in a partition group are used to store information about the same partitioned table. The default value, 64, allows for a maximum of 64 open partitioned tables and 1024 (64 times 16) open partitions.
- **partition spinlock ratio** specifies the number of partition caches that each spinlock protects. A partition spinlock prevents a process from accessing a partition cache currently used by another process.
- The default value of 32 (1 spinlock for every 32 partition caches) is correct for most servers. Increasing or decreasing it may have little impact on performance. The suggested number of available spinlocks is 10 percent of the total number of partitions in use at any one time.
- **perform disk i/o on engine 0** is used on multiprocessor machines to tie disk I/O to SQL Server engine 0.
- **permission cache entries** determines the number of cache protectors per task.

- **print deadlock information** enables printing of deadlock information to the error log.
- **print recovery information** sets a toggle that determines what information SQL Server displays on the console during recovery. The default run value is 0, which means that SQL Server displays only the database name and a message saying that recovery is in progress.
- **procedure cache size** specifies the amount of memory allocated to the procedure cache after SQL Server's memory needs are met. The default run value is 20.
- **recovery interval in minutes** sets the maximum number of minutes per database that SQL Server should use to complete its recovery procedures in case of a system failure. The default is 5 (minutes per database).
- **remote server pre-read packets** controls the number of packets that a site handler will pre-read in connections with remote servers. The default is 3.
- **runnable process search count** specifies the number of times an engine will loop looking for a runnable task before relinquishing the CPU.
- **shared memory starting address** determines the virtual address at which SQL Server starts its shared memory region.
- **size of auto identity column** sets the precision of IDENTITY columns automatically created with the `sp_dboption` "auto identity" option.
- **sort page count** specifies the maximum amount of memory a sort operation can use.
- **sql server clock tick length** specifies the duration of the server's clock tick, in microseconds.
- **stack guard size** specifies the size of the stack guard area.
- **stack size** sets the size of SQL Server's execution stack.
- **systemwide password expiration** is the number of days that passwords remain in effect after they are changed. The default is 0 (passwords do not expire).
- **table lock spinlock ratio** specifies the number of spinlocks protecting the table locks hash table.
- **tape retention in days** sets the number of days that you expect to retain each tape after it has been used for a database or transaction log dump. The default run value is 0.

- **tcp no delay** disables TCP packet batching.
- **time slice** sets the number of milliseconds that SQL Server's scheduler allows a user process to run. The default run value is 100 milliseconds.
- **total data cache size** represents the amount of memory that is currently available for use as a data cache. It is a calculated value that is not directly user-configurable.
- **total memory** sets the size of memory, in 2K units, that SQL Server allocates from the operating system.
- **upgrade version** is changed by the upgrade program provided with new releases.
- **user log cache size** specifies the size (in bytes) for each user's log cache.
- **user log cache spinlock ratio** specifies the number of user log caches per user log cache spinlock.



Parameters by Functional Group

With release 11.0 of SQL Server, configuration parameters are grouped by functional area. While each parameter has a primary group to which it belongs, many are also present in secondary groups. For example, the **number of remote connections** parameter belongs primarily to the Network Communications group but also to the SQL Server Administration group and the Memory Use group. The parameter lists in this section include parameters in all groups to which they belong.

The groups are:

- Backup/Recovery
- Cache Manager
- Disk I/O
- General Information
- Languages
- Lock Manager
- Memory Use
- Network Communications
- Operating System Resources

- Physical Memory
- Processors
- SQL Server Administration
- User Environment

Backup/Recovery Parameters

- print recovery information
- recovery interval in minutes
- tape retention in days

Cache Manager Parameters

- memory alignment boundary
- number of index trips
- number of oam trips
- procedure cache size
- total data cache size

Disk I/O Parameters

- allow sql server async i/o
- disk i/o structures
- number of devices
- page utilization percent

General Information Parameter

- configuration file

Languages Parameters

- default character set id
- default language id
- default sortorder id

- number of languages in cache

Lock Manager Parameters

- address lock spinlock ratio
- deadlock checking period
- freelock transfer block size
- max engine freelocks
- number of locks
- page lock spinlock ratio
- table lock spinlock ratio

Memory Use Parameter

- executable code size

Network Communications Parameters

- allow remote access
- default network packet size
- max network packet size
- max number of network listeners
- number of remote connections
- number of remote logins
- number of remote sites
- remote server pre-read packets
- tcp no delay

Operating System Resources Parameters

- max async i/o's per engine
- max async i/o's per server
- shared memory starting address

Physical Memory Parameters

- additional network memory
- lock shared memory
- total memory

Processors Parameters

- max online engines
- min online engines

SQL Server Administration Parameters

- allow nested triggers
- allow updates to system tables
- audit queue size
- cpu accounting flush interval
- cpu grace time
- default database size
- default fill factor percent
- housekeeper free write percent
- identity burning set factor
- identity grab size
- i/o accounting flush interval
- i/o polling process count
- lock promotion hwm
- number of alarms
- number of extent i/o buffers
- number of mailboxes
- number of messages
- number of open databases
- number of open objects
- number of pre-allocated extents

- number of sort buffers
- print deadlock information
- runnable process search count
- partition groups
- partition spinlock ratio
- size of auto identity
- sort page count
- sql server clock tick length
- time slice
- upgrade version

User Environment Parameters

- number of user connections
- permission cache entries
- stack guard size
- stack size
- systemwide password expiration
- user log cache size
- user log cache spinlock ratio

B

Guide to SQL Server Manager Icons

What's in This Appendix

This appendix is a guide to the icons used in SQL Server Manager. The appendix contains the following parts:

- An alphabetical list of icons, showing a picture and label for each type of icon
- A list of operations you can perform by dragging one icon onto another

List of Icons

The following table illustrates and identifies each type of icon used in SQL Server Manager.

Table B-1: SQL Server Manager icons






Icon	Object
	Container
	Database
	Database device
	Default
	Dump device

Table B-1: SQL Server Manager icons (continued)




















Icon	Object
	Group
	Index
	Login
	Named Cache
	Procedure
	Remote server
	Rule
	Segment
	sqledit utility
	SQL Server (connected)

Table B-1: SQL Server Manager icons (continued)

Icon	Object
	SQL Server (unconnected)
	SQL Server Manager application
	Table (system)
	Table (user)
	Trigger
	User
	User datatype
	View
	wisql utility

Valid Drag-and-Drop Operations

The following table summarizes the operations you can perform by dragging one icon on top of another. In the Opened Dialog Box

column, **tab** refers to a specific screen in a multiple-screen object dialog box. For example, the Login dialog box's User tab is the Login dialog box screen for managing the relationships between logins and users.

Table B-2: Icon drag-and-drop operations













Operation	Icon 1	Icon 2	Opened dialog box
Add an alias between a login and a user	 Login	 User	Login dialog box's Users tab
Add a user to a database	 Login	 Database	Create User
Add a user to a group and remove the user from the old group if appropriate	 User	 Group	Group dialog box Properties tab
Back up a database onto a dump device	 Database	 Dump Device	Database Backup dialog box
Bind a column in a table to a default	 Table	 Default	Default dialog box Bindings tab
Bind a column in a table to a rule	 Table	 Rule	Rule dialog box Bindings tab

Table B-2: Icon drag-and-drop operations (continued)

























Operation	Icon 1	Icon 2	Opened dialog box
Bind a user datatype to a default	 User Datatype	 Default	Default dialog box Bindings tab
Bind a user datatype to a rule	 User Datatype	 Rule	Rule dialog box Bindings tab
Extend a database onto a database device	 Database	 Database Device	Database Properties dialog box
Extend a segment onto a device	 Segment	 Database Device	Segment dialog box Properties tab
Modify a group's permissions on a procedure	 Group	 Procedure	Group dialog box Object Permissions tab
Modify a group's permissions on a table	 Group	 Table	Group dialog box Object Permissions tab
Modify a group's permissions on a view	 Group	 View	Group dialog box Object Permissions tab

Table B-2: Icon drag-and-drop operations (continued)

Operation	Icon 1	Icon 2	Opened dialog box
Modify a user's permissions on a procedure	 User	 Procedure	User dialog box Object Permissions tab
Modify a user's permissions on a table	 User	 Table	User dialog box Object Permissions tab
Modify a user's permissions on a view	 User	 View	User dialog box Object Permissions tab
Place new index growth onto a segment	 Index	 Segment	Index dialog box Segments tab
Place new table growth onto a segment	 Table	 Segment	Table dialog box Segments tab

SQL Server 11.0 Drag-and-Drop Operations

When you connect to a release 11.0 SQL Server, the following additional drag-and-drop operations are valid:

Table B-3: SQL Server 11.0 icon-drag and-drop operations







Operation	Icon 1	Icon 2	Opened dialog box
Bind a database to a cache	 Cache	 Database	Cache dialog box Bindings tab

Table B-3: SQL Server 11.0 icon-drag and-drop operations (continued)

Operation	Icon 1	Icon 2	Opened dialog box
Bind an index to a cache	 Cache	 Index	Cache dialog box Bindings tab
Bind a table to a cache	 Cache	 Table	Cache dialog box Bindings tab

Glossary

account locking

A SQL Server facility that prevents a user from logging in to the SQL Server, but permits the user's account to own databases and objects. The user can be denied access without disrupting the permissions the user may have granted to other users or the availability of database objects owned by the user. Also known as login locking.

alias

A defined user login that can be used as an alternate name. Using an alias, multiple users can have the same privileges in a database.

allocation unit

An allocation unit is a logical unit of SQL Server storage equal to 256 2KB data pages (1/2 megabyte).

authorization

See **role**.

backup

A copy of a database or transaction log, used to restore data after a media failure.

Backup Server

Backup Server performs local or remote backups (dumps) and restores (loads) on selected databases and transaction logs on behalf of SQL Server. A Backup Server must be running on the same system as each SQL Server.

binding

An association between a default or a rule and a table column or a user datatype. When a rule or a default is bound to a table column or a specified user datatype, it affects the data that can be entered in the column. Binding can also refer to an association between a named data cache and a database, table, or index. When a database, table, or index is bound to a named data cache, reads from the database, table, or index go into the named cache.

cache

A portion of SQL Server memory used to store the most recently used database information. Data cache holds data and transaction log information, and procedure cache holds query execution plans for stored procedures. Users can partition the default data cache into named caches.

cascading menu

A submenu that appears to the right of a selected pull-down menu item. An arrow (->) next to an item on a pull-down menu indicates the existence of a cascading menu.

check constraint

A method of restricting the data inserted into a table column. When an insert on the column occurs, the value of the data being inserted must pass the criteria specified in the check constraint before SQL Server updates the table.

checkpoint

The point at which all data pages that have changed are guaranteed to have been written to the database device.

clustered index

An index in which the physical order and the logical (indexed) order are the same. The leaf level of a clustered index represents the data pages themselves. You can have only one clustered index per table.

command dialog box

A dialog box that opens when you execute a command that requires information on how it should execute, such as a dialog box for creating a new object or binding a rule.

command permissions

Permissions to create a specific object, such as `create table` or `create procedure`. Command permissions are granted and revoked by a Database Owner.

constraints

Mechanisms for restricting the data that can be inserted in a table. See **check constraint**, **default**, **referential constraint**.

container icon

An icon that represents all objects of a specific type that are contained in a database or SQL Server. For example, the users container icon represents all users in a database.

context-sensitive menu

A menu whose appearance in the menu bar depends on the selection of an icon representing a SQL Server or database object. For example, the Login menu appears in the menu bar only when a login container icon or login icon is selected.

data definition

The process of setting up databases and creating database objects such as tables, indexes, rules, defaults, procedures, triggers, and views.

data definition language (DDL)

Transact-SQL scripts that capture the state of a database. These scripts can recreate the database in the same location or on a different server.

database

A set of related tables of data and other database objects that are organized and presented to serve a specific purpose.

database device

A device dedicated to the storage of the objects that make up databases. It can be any piece of a disk or a file in the file system that is used to store databases and database objects.

database object

A database object is one of the components of a database: user, group, login, table, view, index, procedure, trigger, column, default, or rule.

database owner

The owner of a database. The System Administrator creates a database and is the original database owner. The System Administrator can then designate another user as the owner of a database. A database owner controls all the database objects in that database and can grant object and command permissions to other users. The user name for the database owner within his or her own database is *dbo*.

datatype

Specifies what kind of information each column holds and how the data is stored. Datatypes include *char*, *int*, *money*, and so on. Users can construct their own datatypes based on the SQL Server system datatypes.

dbcc commands

Instructions to the Database Consistency Checker (dbcc), which checks the logical and physical consistency of a database.

dbo

In a user's own database, SQL Server recognizes the user as "dbo." A database owner (dbo) logs into SQL Server using his or her assigned login name and password.

DDL

See **data definition language**.

default

1. The user-specified value that the server inserts for a column when no value is provided. 2. The option chosen by the system when no other option is specified.

default database

The database that users get by default when they log in to a SQL Server.

dependency

A relationship between objects that occurs when one object refers to another, such as a stored procedure that refers to a table. If you delete the table without changing the stored procedure, errors occur.

disk allocation pieces

Disk allocation pieces are the groups of allocation units from which SQL Server constructs a new database file. The minimum size for a disk allocation piece is one allocation unit (256 2Kb pages).

disk initialization

The process of preparing a database device or file for SQL Server use. Once the device is initialized, it can be used for storing databases and database objects.

disk mirror

A duplicate SQL Server database device. All writes to the device being mirrored are copied to a second physical device. If one device fails, the other contains an up-to-date copy of all transactions.

drag and drop

A quick way to perform operations involving two SQL Server or database objects. You can drag the icon for one object and drop it on the icon for another object to perform a variety of operations, including backing up the database, creating a user, binding rules, modifying permissions, and so on.

dynamic configuration parameter

A SQL Server configuration parameter that is updated immediately when you reset it; a restart of SQL Server is not needed.

dump

A backup copy of a database or a transaction log, or the process of creating a backup.

error message

A message that SQL Server issues, usually to the user's terminal, when it detects an error condition.

error state number

The number attached to a SQL Server error message that allows unique identification of the line of SQL Server code at which the error was raised.

extent

When a table or index requires space, SQL Server allocates a block of eight 2K pages. This block is called an extent. Each 256-page allocation unit contains 32 extents.

guest

If the user name "guest" exists in the *sysusers* table of a database, any user with a valid SQL Server login can use that database, with limited privileges.

hysteresis

A value used to control the spacing of thresholds on a segment and to prevent the stored procedure associated with a threshold from being triggered too frequently.

icon

A graphical representation of an object or an action.

identifier

A string of characters used to identify a database object, such as a table name or column name.

identity column

A table column containing a system-generated value that uniquely identifies each row in the table. A table can have only one identity column.

index

An index is created on one or more columns of a table to speed up data retrieval by pointing to the place where the column's data is stored. See also **unique index**, **clustered index**, and **nonclustered index**.

initialization

See **disk initialization**.

interfaces file

The interfaces file is an operating system file that must be available on each machine from which connections to SQL Servers are made. By default, this file is located in the directory that is specified in the SYBASE environment variable.

Each entry in the interfaces file tells the host machine's front-end software how to connect to a SQL Server. An interfaces file entry contains the name of a SQL Server and a list of services provided by the server.

isql

See **wisql**.

keyword

A word or phrase that is reserved for exclusive use by Transact-SQL. Also known as **reserved word**.

last-chance threshold

A threshold created by SQL Server on a segment reserved for transaction log activity. SQL Server automatically adjusts its placement on the segment to ensure that enough space remains to dump the transaction log.

load

1. A copy of a database, used to recover from a media failure. 2. The process of copying a database backup to recover a database.

locking

The process of restricting access to resources in a multi-user environment to maintain security and prevent concurrent access problems. SQL Server automatically applies locks to tables or pages. The System Security Officer can lock a user's login to restrict access without the time and complications involved with deleting a user or a login.

login

The name a user uses to log in to SQL Server. A login is valid if SQL Server has an entry for that user in the system table *syslogins*.

master database

Controls the user databases and the operation of SQL Server as a whole. Known as *master*, it keeps track of such things as user accounts, ongoing processes, and system error messages.

message dialog box

A dialog box that communicates information from SQL Server Manager or SQL Server. Some message dialog boxes present multiple messages that you can read by using a scroll bar on the dialog box.

message number

The number that uniquely identifies an error message.

mirror

See **disk mirror**.

model database

A template for new user databases. Each time a database is created, SQL Server makes a copy of *model* and extends it to the size requested, if necessary.

nonclustered index

An index that stores key values and pointers to data. The leaf level points to data pages rather than containing the data itself.

nondynamic configuration parameter

A SQL Server configuration parameter that requires you to restart SQL Server after resetting the value.

null

Having no explicitly assigned value. NULL is not equivalent to zero or to blank. A value of NULL is not considered to be greater than, less than, or equivalent to any other value, including another value of NULL.

object

See **database object**.

object icon

See **icon**.

object owner

User who owns an object either by having created the object or by being given ownership. For example, a System Administrator can designate a user as a database owner; a database owner can designate a user as a table owner or give a user permission to create a table.

object permissions

Permissions to access and modify tables, views, or procedures, such as **select**, **insert**, **execute**, and so on. Object permissions are granted and revoked by an object owner.

object dialog box

A Dialog box that displays information about a database object. The information appears on multiple screens, called **tabs**, within the dialog box. You can use the dialog box to view or modify the object. The dialog box also provides a direct navigation path to related objects.

object-specific menu

A menu whose commands are specific to objects contained or described in the window. The object-specific menu appears in the menu bar at the top of the window when the object is selected.

operating system

A group of programs that translates your commands to the computer, helping you perform such tasks as creating files, running programs, and printing documents.

Operator

A SQL Server user in charge of performing server-wide database operations such as backing up and restoring databases. The System Security Officer can assign the role of Operator to a user.

See also **role**.

permission

The authority to perform actions on certain database objects, such as **execute** a procedure or **select** from a table, or to run certain commands, such as **create table**.

See also **privilege**, **command permissions** and **object permissions**.

precision

The number of significant digits in a *numeric* or *decimal* datatype that can be stored in a column. For *float* datatypes, precision is the number of significant binary digits in the mantissa.

privilege

The authority to access database objects and use database commands that is implied with a user's role.

See also **permission**.

procedure

A collection of SQL statements and optional control-of-flow statements stored under a name. SQL Server-supplied procedures are called system procedures.

public

All registered users of the database are members of the group, "public." Users at this level of authority can create a temporary table and have access to objects whose owners have granted permissions to "public." Users at this level of authority can also dump and load transactions.

pulldown menu

A menu opened from a main menu selection. The pulldown menu appears below the main menu selection.

query

1. A request for the retrieval of data. 2. Any SQL statement that manipulates data.

recovery

The process of rebuilding one or more databases from database dumps or log dumps or both.

referential constraint

A type of constraint used to ensure that data being inserted in specified columns of one table matches the data in specified columns of another table.

remote procedure calls (RPCs)

A procedure executed on a different SQL Server from the server where the user is logged in.

role

A SQL Server user's authorization level: System Administrator, System Security Officer, or Operator. Role confers permission to use commands and access and modify database objects. A user may be assigned more than one role, and more than one user may have the same role.

rule

The domain of acceptable values for a table column or a user datatype.

sa

See **System Administrator**.

scale

The number of digits to the right of the decimal point in a *numeric* or *decimal* datatype.

schema

A collection of objects owned by a single user and created in one transaction. The schema can include tables, permissions, and all database objects.

segment

A named subset of database devices available to a particular database. It is a label that points to one or more database devices. Segments can be used to control the placement of tables and indexes on specific database devices.

server user ID

The ID number by which a user is known to SQL Server.

severity level number

The severity of an error condition: errors with severity levels of 19 and above are fatal errors.

shortcut menu

A menu opened by clicking the second mouse button over an icon. The shortcut menu contains the same commands as the object-specific menu.

sqledit

A utility for creating and editing *sql.ini* files and file entries.

sql.ini file

The interfaces file containing definitions for each SQL Server to which your workstation can connect. The file must be on each machine from which clients connect to SQL Servers. Each *sql.ini* file entry tells a client or host machine how to connect to a specific SQL Server. The file contains the name of the SQL Server, a list of services provided by the SQL Server, and the port to use for connecting to the SQL Server for each service.

SQL Server

The server in the Sybase Client/Server architecture, SQL Server manages multiple databases and multiple users, keeps track of the actual location of data on disks, maintains mapping of logical data description to physical data storage, and maintains data and procedure caches in memory.

SQL Server Manager

A forms-based interface for administration of SQL Servers.

status bar

A horizontal bar at the bottom of the Voyager window that displays information about the current action or object.

See also **toolbar**.

stored procedure

See **procedure**.

suid

See **server user ID**.

symmetrical drag-and-drop

An icon drag-and-drop feature in which either one of two icons can be dragged onto the other one to start an operation.

System Administrator

The user in charge of administrative tasks including managing disk storage, creating databases, creating user accounts, assigning permissions, and running diagnostic and repair functions. The System Administrator's login name is "sa."

See also **role**.

system databases

The three databases on a newly installed SQL Server: the *master* database, which controls user databases and the operation of the SQL Server; the temporary database (*tempdb*), used for temporary tables; and the *model* database, which is used as a template to create new user databases.

system function

A function that returns special information from the database, particularly from the system tables.

system procedures

Stored procedures that SQL Server supplies for use in system administration. These procedures are shortcuts for retrieving information from the system tables, as well as mechanisms for accomplishing database administration and other tasks that involve updating system tables.

System Security Officer

The user in charge of security-sensitive tasks, such as creating, dropping, and locking user accounts, and changing passwords of other users.

See also **role**.

system table

One of the data dictionary tables. The system tables keep track of information about the SQL Server as a whole and about each user database. The *master* database contains some system tables that are not in user databases.

tab

An interactive screen displayed within a dialog box as the result of pressing one of the buttons across the top of the dialog box or the Go To... button or dragging one icon onto another icon. Tabs display information about an object, allow you to modify the object, and allow you to navigate to other objects.

temporary database

The temporary database in SQL Server, *tempdb*, that provides a storage area for temporary tables and other temporary working storage needs.

threshold

A space usage value of a database or log segment. When free space in a segment falls below a threshold value, a system procedure executes.

toolbar

A horizontal or vertical bar in the window that contains buttons. The buttons give you alternate ways to execute menu commands or let you show and hide objects.

See also **status bar**.

ToolTip

A small window that appears when the mouse cursor rests over a command button for one second or more. The window contains brief help on the command button.

transaction

A mechanism for ensuring that a set of actions is treated as a single unit.

transaction log

A system table (*syslogs*) in which all changes to the database are recorded.

trigger

A special form of stored procedure attached to a column that goes into effect when a user gives a change command such as **insert**, **delete**, or **update** to a specified table or column. Triggers are often used to enforce referential integrity.

unique index

An index in which no two rows can have the same key value. You cannot create a unique index on a column that includes duplicate values or more than one null value.

user ID

The ID number by which a user is known in a specific database. Distinct from **server user ID**.

user table

A database table that stores user data.

view

An alternative way of looking at the data in one or more tables. Usually created as a subset of columns from one or more tables.

Voyager

SQL Server Manager's graphical, hierarchical representation of the object relationships in the SQL Servers specified in a user's *sql.ini* file. Voyager provides access to all objects and activities managed by SQL Server Manager.

wash area

A portion of each memory pool within a data cache. The wash area is used to ensure that queries that need clean pages in a data cache can find them. When the number of dirty pages (pages that have been changed in cache) fills the rest of the buffers and begins to enter the wash area, SQL Server writes the data in the wash area to disk. When this write completes, the wash area is marked clean and is available for queries needing clean pages.

wisql

A utility for connecting to SQL Server. The Windows or NT application, **wisql**, allows you to create, execute, and edit queries against a SQL Server database. You can also submit DDL scripts to SQL Server for execution by using the **wisql** utility.

Index

Symbols

@@*thresh_hysteresis* system
variable 7-30

A

Abort Transaction If Log Full database
option 7-20

Access 8-1

aliases 8-1

command permissions 8-1

dialog boxes 8-46

granting 8-46

object permissions 8-1

revoking 8-46

roles 8-1, 8-46

shared logins 8-1

tabs 8-46

Active transaction 4-15

Activities

drag-and-drop B-3

activities

drag and drop B-3

Add button 2-30, 7-4, 9-31, 10-8, 10-14

Add Identity Column in Non-Unique

Indexes database option 7-21

Adding

columns 2-30, 9-59

disk mirroring 6-11

parameters 9-30

additional netmem configuration

parameter A-1

additional network memory

configuration parameter A-4

address lock spinlock ratio configuration

parameter A-4

Administration tasks 1-1, 2-25

Advanced configuration parameter

display level 4-11

Aliases

access 8-1

navigating to logins 8-22

sharing logins 8-2, 8-14

Allocating

databases on devices 7-4

index pages 9-26

space for logs 7-2

space on new device 7-10

storage space 7-2, 7-9

table pages 10-37

Allocation

errors 10-37

maps 10-37

pages 10-37

allow nested triggers configuration

parameter A-4

Allow Nulls by Default database

option 7-21

allow remote access configuration

parameter A-4

Allow Select Into and Bulk Copy

database option 7-20

allow sql server async i/o configuration

parameter A-4

allow updates configuration

parameter A-1

allow updates to system tables configuration

parameter A-4

Apply button 9-2

Assigning

columns 9-20

indexes 9-20

permissions 2-31

roles 2-21, 8-9, 8-11

audit queue size configuration

parameter A-1, A-4

Auto Identity database option 7-21

B

Backing up

- configuration parameters A-11
- dump devices 7-34
- guidelines 7-38
- identifying Backup Server 7-34
- master database 7-6, 7-33
- on multiple volumes 7-42
- planning 7-33
- scheduling 7-38
- transaction logs 7-20
- using `dbcc` 7-50
- volume change prompts and 7-42
- Backup, Truncate, and Log database
 - backup option 7-40
- Backup/Recovery configuration
 - parameters A-11
- Backup and Log database backup
 - option 7-40
- Backup Server 7-34
- Bars
 - Contents 9-5
 - displaying 3-16
 - help 3-18
 - hiding 2-9, 3-16
- Batch files
 - DDL 2-37
 - defining objects 2-36
- bcp utility 7-20
- Binding
 - defaults 9-14, 9-15
 - deleting. *See* Unbinding.
 - future use 9-16, 9-46
 - objects to a named cache 5-9
 - rules 9-45
- Blocked time, process 4-14
- Blocking spid 4-14
- Buffer pools
 - creating 5-6
 - deleting 5-9
- Buffer pool wash area 5-8
- Buffer prefetch 9-22, 10-6
- Building indexes 9-21
- Buttons
 - Add 2-30, 7-4, 9-31, 10-8, 10-14
 - Apply 9-2

- Cancel 2-4
- Change 7-4, 9-31, 10-8, 10-14
- Connect 2-4, 4-4
- Consistency... 9-26
- Create 2-21, 2-31, 9-2
- Disconnect 4-5
- Done 2-18, 9-3
- Go To... 2-17
- Network... 4-24, 4-25, 7-14, 7-15
- Refresh 3-25
- Remove 7-4, 9-31, 10-8, 10-14
- tab 2-16, 3-20, 9-6
- toolbar 3-16

C

- Cache
 - See also Named caches
 - and large I/Os 9-22, 10-6
 - data 5-1
 - management of 5-1
 - procedure 5-1
 - replacement strategy, index 9-22
 - replacement strategy, table 10-6
- Cache Bindings tab 5-11
- Cache management configuration
 - parameters A-11
- Cache Properties tab 5-5
- Cancel button 2-4
- Case sensitivity
 - search strings 3-9
- Change button
 - allocating space on a device 7-4
 - defining columns 10-8, 10-14
 - defining parameters 9-31
- Change Password dialog box 2-6, 8-13
- Changing
 - database defaults 7-18
 - default devices 6-17
- Changing log buffer size 7-13
- Characters
 - masking 3-9
 - wildcard 3-9
- Check constraints 10-2, 10-16

- column 9-39, 10-2, 10-8
- creating 10-25
- format of 10-10, 10-27
- table 9-39
- ways of creating 10-2
- Checking
 - database consistency 7-50
 - index consistency 9-26
 - SQL Server network I/O 4-29
- Checkpoints 7-22
 - issuing automatically 7-22
 - transaction logs 7-22
- Class on system administration xxv
- Closing dialog boxes 2-18, 9-3
- Clustered indexes 9-17, 9-20
- Column check constraint 9-39, 10-2
- Column constraints 10-8, 10-14
- Columns
 - adding 2-30, 10-7, 10-13
 - assigning indexes 9-20
 - binding defaults 9-14
 - binding rules 9-45
 - default values 9-8
 - displaying 9-13
 - identity 9-56
 - removing 10-8, 10-14
 - unbinding 9-16, 9-47
 - views 9-59
- Command line options 2-2
- Commands
 - access 8-1
 - create 8-4
 - dialog boxes 3-22
 - executing triggers 9-47
 - granting permissions 8-28
 - permissions 8-1, 8-4, **8-42 to 8-45**
 - revoking permissions 8-28
 - SQL 9-4
 - Transact-SQL 4-23, 7-13
- Common procedures 9-1
- Communicating with SQL Servers 11-12
- Complete configuration parameter display level 4-11
- Configuration... command 4-10, 4-12
- configuration file configuration parameter A-4
- Configuration parameters
 - backup/recovery A-11
 - cache manager A-11
 - disk I/O A-11
 - functional groups of A-10 to A-14
 - general information A-11
 - languages A-11
 - lock manager A-12
 - processors A-13
 - resetting 4-10, 4-12
 - SQL Server 10.x A-1 to A-3
 - SQL Server 11.0 A-4 to A-10
 - SQL Server administration A-13
 - user environment A-14
- Configuration Parameters dialog box
 - SQL Server 10.x 4-10
 - SQL Server 11.0 4-12
- Configuring
 - SQL Server 11.0 4-11
 - SQL Servers 4-8
 - System Administrators and 4-9
 - System Security Officers and 4-9
- Confirmation dialog box 9-3
- Confirmation dialog boxes 2-38, 9-7
- Connect...
 - button 2-4, 4-4
 - menu command 2-4
- Connect dialog box 2-3, 2-4, 4-2
- Connecting
 - shortcuts 2-5, 4-4
 - SQL Server 4-1
 - SQL Server Manager 2-4
 - switching between connections 11-15
 - with wisql 11-14
- Connect toolbar button 3-17
- Consistency...
 - button 9-26
- Consistency... command 4-29
- Consistency checking
 - database 7-50
 - dbcc utility 7-50

- index 9-26
- Constraints
 - and foreign keys 10-32, 10-34
 - and primary keys 10-32
 - check 10-2, 10-10, 10-16, 10-25, 10-27
 - column 10-8, 10-14
 - creating 10-1
 - default 10-2
 - description of 10-2 to 10-4
 - primary keys in 10-34
 - referential 10-2, 10-3, 10-30
 - rules 10-2
 - summary of 10-3
- Contacting Technical Support xxx
- Container icons 3-2
- Contents bar 9-5
- Context-sensitive menus
 - description of 3-13
 - pop-up 3-14
 - pull-down 3-14
 - shortcut 3-14
- Copying device contents 6-10
- cpu accounting flush interval configuration
 - parameter A-4
- cpu flush configuration option A-1
- cpu grace time configuration
 - parameter A-4
- CPU time 4-14
- CPU usage
 - configuration parameters
 - affecting A-4
- Create
 - button 2-21, 2-31, 9-2
 - menu command 9-2
- Create dialog boxes
 - Cache 5-2
 - Database 7-3
 - Database Device 6-3
 - Default 9-9
 - Dump Device 7-35
 - Group 8-30
 - Index 9-19
 - Login 8-8
 - Procedure 9-30
 - Remote Server 4-17
 - Rule 9-40
 - Segment 7-24
 - Trigger 9-49
 - User 8-18
 - User Datatype 9-55
 - View 9-60
- Create toolbar button 3-17
- Creating
 - buffer pools 5-6
 - check constraints 9-31, 10-10, 10-16, 10-25
 - column constraints 10-8, 10-14
 - constraints 10-1
 - database devices 6-3
 - databases 7-1, 7-3
 - defaults 9-8
 - devices 6-2
 - directories (DDL) 2-36
 - dump devices 7-34
 - groups 8-29
 - guest users 7-4
 - indexes 9-18
 - logins 8-7
 - named cache binding 5-10
 - named cache buffer pools 5-6
 - named caches 5-2
 - objects 9-2
 - primary key index 9-20
 - procedures 9-29
 - referential constraints 10-30
 - remote servers 4-16
 - rules 9-31, 9-39, 9-40
 - schemas 9-4
 - segments 7-23
 - segment thresholds 7-30
 - shortcuts 9-2
 - table columns 10-7, 10-13
 - tables 2-28, 10-1, 10-4
 - toolbars and 1-3
 - triggers 9-48
 - user datatypes 9-55
 - users 2-25, 8-18
 - views 9-59

Creating check constraints 9-39

Cumulative CPU time 4-14

D

d_master device 6-3

Data

retrieving 9-17

sorting 9-21

Data and log cache 5-6

Database Allocations dialog box 7-11

Database Backup dialog box 7-39, 7-40

Database backup options 7-40

Database Consistency Check dialog box 7-51

Database Definition Language (DDL).
See DDL.

Database Device Databases tab 6-8

Database Device menu 6-6

Database Device Properties tab 6-6

Database Device Segments tab 6-9

Database Generate DDL dialog box 7-14

Database Options dialog box 7-19

Database Properties dialog box 7-8

Database Run DDL dialog box 7-15

Databases

allocating on devices 7-4

backing up 7-6

backup guidelines 7-38

binding to a named cache 5-9

changing log buffer size 7-13

checking consistency 7-50

checkpoints 7-22

creating 7-1, 7-3

default options 7-18

deleting damaged 7-7

deleting normal 7-6

devices 6-2, 6-7

displaying 8-14

displaying objects 9-5

displaying properties 7-7

estimating size 7-1

executing DDL 7-15

guest users 7-4

increasing storage 7-9

master 6-2

model 6-3

navigating to 6-8

permissions 8-3

pubs2 2-7

recovering 7-44

reducing size 7-10

segments 6-9, 7-23

setting online 7-55

storing 6-2, 6-7

sybssystemprocs 6-3

tempdb 6-3

transferring ownership 7-12

database size configuration
parameter A-1

Data cache 5-1

Data Definition Language (DDL). *See*
DDL.

Data Definition Language in Transaction
database option 7-21

Data logs 6-17

dbcc

index consistency checking 9-26

network I/O checking 4-29

options 7-52

table consistency checking 10-35

utility 7-50, 10-35

DDL

batch files 2-37

deleting directories 2-38

editing 4-26 to 4-28, 7-16 to 7-18

executing scripts 4-23, 7-15

files containing 4-26 to 4-28, 7-16 to
7-18

generating directories 2-35

generating scripts 4-23

schemas 2-35

Transact-SQL commands 7-13

ddl directory 4-24, 4-26, 7-16

deadlock checking period configuration
parameter A-4

Default Bindings tab 9-15

- default character set id configuration
 - parameter A-4
- default database size configuration
 - parameter A-5
- Default Dependencies tab 9-13
- default fill factor percent configuration
 - parameter A-5
- default language configuration
 - parameter A-1
- default language id configuration
 - parameter A-5
- default network configuration
 - parameter A-1
- default network packet size configuration
 - parameter A-5
- Default Properties tab 9-12
- Defaults
 - as constraints 10-2
 - binding 9-14
 - Bindings tab 9-15
 - changing device 6-17
 - creating 9-9
 - databases 8-9, 8-11
 - dependencies 9-13
 - Dependencies tab 9-13
 - displaying properties 9-12
 - languages 8-9, 8-11
 - multiple devices 6-16
 - navigating to dependencies 9-13
 - options 7-18
 - permissions 8-41 to 8-45
 - storage devices 6-3, 6-4
 - unbinding 9-16
 - ways of creating 9-8, 10-2
- default* segment 7-23
- default sortorder id configuration
 - parameter A-5
- Delete... menu command
 - database 7-6
 - deleting objects 9-7
 - shortcut menu 2-38
- Delete confirmation dialog box 9-7
- delete permission 8-25, 8-37, 9-68
- Delete toolbar button 3-17
- Deleting
 - bindings. *See* Unbinding
 - buffer pools 5-9
 - database devices 6-5
 - databases 7-6
 - directories (DDL) 2-38
 - dump devices 7-36
 - files 2-38
 - groups 8-31
 - logins 8-9
 - named cache binding 5-12
 - Named cache buffer pools 5-9
 - named caches 5-3
 - objects 2-38, 9-7
 - segments 7-25
 - segment thresholds 7-32
 - tables 10-11
 - users 8-19
- Dependencies
 - displaying default 9-13
 - displaying procedure 9-34
 - displaying rule 9-43
 - displaying table 10-17
 - displaying trigger 9-53
 - displaying user datatype 9-58
 - displaying view 9-65
 - tabs showing 2-16
- Details display format 3-6
- Details toolbar button 3-18
- Devices
 - allocating databases 7-4
 - changing default 6-17
 - copying contents 6-10
 - creating 6-2, 6-3
 - d_master* 6-3
 - databases 6-2, 6-7
 - default storage 6-3, 6-4
 - deleting 6-5
 - disks 6-2
 - displaying properties 6-6
 - dump 6-2, 7-34
 - extending segments 7-27
 - master 6-16
 - mirroring 6-4

- multiple default 6-16
- navigating to 7-27
- segments 6-9
- segments pointing to 7-23
- separating usage 6-17
- storage space on 6-7
- storing transaction logs 7-4
- devices.ddl* file 4-26
- devices configuration parameter A-1
- Dialog Boxes
 - Filter by Name 3-9
- Dialog boxes
 - access 8-46
 - closing 2-18, 9-3
 - commands 3-22
 - confirmation 2-38, 9-7
 - granting permissions 8-46
 - messages 3-24
 - navigating to 3-21
 - object 3-20
 - refreshing 3-25
 - tab buttons 2-16
 - tabs 2-16, 3-20
- Directories
 - creating DDL 2-36
 - ddl* 4-24
 - deleting DDL 2-38
- Direct updates to system tables A-4
- Disabling disk mirroring 6-13
- Disconnect button 4-5
- Disconnect dialog box 4-5
- Disconnecting
 - with *wisql* 11-15
- Disconnect toolbar button 3-17
- Disk I/O configuration
 - parameters A-11
- disk i/o structures configuration
 - parameter A-5
- Disk reads and writes 4-14
- Disks
 - adding mirroring 6-11
 - devices 6-2
 - disabling mirroring 6-13
 - mirroring 6-1, 6-10
 - re-enabling mirroring 6-15
- Display formats, selecting 3-6
- Displaying
 - bars 3-16
 - columns 9-13
 - database device properties 6-6
 - database properties 7-7
 - databases 8-14
 - dependencies 9-13
 - details about an object 9-3
 - dump device properties 7-37
 - groups 8-33
 - icons 2-9, 3-16
 - indexes 10-24
 - logins 8-10
 - named caches 5-5
 - objects 9-6
 - procedure dependencies 9-34
 - procedures 10-18
 - properties used in schema 9-4
 - rule dependencies 9-43
 - segment properties 7-25
 - segments 9-25
 - SQL Server and database objects 9-5
 - SQL Server configuration
 - parameters 4-11, 4-12
 - SQL Server processes 4-13
 - SQL Server properties 4-6
 - table dependencies 10-17
 - tables 9-35
 - toolbars 2-9
 - trigger dependencies 9-53
 - triggers 9-35, 10-18, 10-24
 - user datatype dependencies 9-58
 - user datatypes 9-13, 9-35
 - user processes 4-13
 - users 8-20
 - view dependencies 9-65
 - views 9-35, 10-18
- Display menu 2-9, 3-13
- Done button 2-18, 9-3
- Drag-and-drop 1-2
 - activities summarized B-3
 - icons 3-10

- tables 10-21
- tutorial 2-25
- users 9-69, 10-21
- views 9-69
- Dragging. *See* Drag-and-drop.
- Dragging and dropping icons 3-10 to 3-11
- Drop Connections with No Activity
 - remote server option 4-17
- Drop-down list toolbar control 3-17
- dump database command 7-20
- Dump Device Properties tab 7-37
- Dump devices 6-2
 - backing up 7-34
 - creating 7-34
 - deleting 7-36
 - displaying properties 7-37
- Dumping transaction logs 7-20
- Duplicating. *See* Mirroring and Copying.

E

- Editing
 - DDL scripts 4-23
 - sql.ini* files 11-1
 - SQL queries 11-16
- Edit Platform & Net-Library Driver
 - dialog box 11-10
- Encrypt Password remote server
 - option 4-18
- Enforcing referential integrity 9-47, 10-30
- engine adjust configuration
 - parameter A-2
- Engine number, process 4-14
- Estimating
 - database size 7-1
 - running *sp_estspace* 7-2
 - running *wisql* 7-2
 - space requirements 7-2
- Executable, SSM 2-2
- executable code size configuration
 - parameter A-5

- execute permission 9-36, 9-38
- Executing
 - DDL scripts 4-23, 7-15
 - procedures 9-36
 - SQL queries 11-16
 - triggers 9-47, 9-49
- Exiting SQL Server Manager 2-4
- Expression, rule 9-41
- Extending segments 7-27
- extent *io* configuration parameter A-2

F

- File menu 3-12
- Files
 - .ddl* 4-26 to 4-28, 7-16 to 7-18
 - batch 2-37
 - creating 2-36
 - defining objects 2-36
 - deleting 2-38
 - devices.ddl* 4-26
 - remote.ddl* 4-26
 - sql.ini* 11-1
 - user.ddl* 2-37
- Fill factor 9-21
- fill factor configuration option A-2
- fill factor configuration parameter A-2
- Filter By Name... menu command 3-9
- Filter by Name dialog box 3-9
- Filter by Name toolbar button 3-17
- Filtering and status bar 3-9, 3-18
- Finding
 - case sensitivity 3-9
 - search strings 3-9
- Foreign keys 10-32
 - Referential constraints
 - foreign keys in 10-34
- freelock transfer block size configuration
 - parameter A-5
- Future bindings 9-16, 9-46

G

General information configuration
parameters A-11

Generate DDL... menu command 2-36

Generating

database DDL 2-35

Server DDL 4-23

Go To... button 2-17

Granting

command permissions 8-28, 8-40

object permissions 9-38

procedure permissions 9-36

select permission 2-33

table permissions 10-18

view permissions 9-66

with grant 8-5

with grant 9-38

Group Command Permissions tab 8-40

Group Object Permissions tab 8-36

Group Properties tab 8-32, 8-34

Groups

creating 8-29

deleting 8-31

displaying 8-33

modifying 8-22

navigating to 9-38, 9-69, 10-21

object permissions 8-36

procedure permissions 9-36

Public 8-18

removing users 8-31

sharing permissions 8-21, 8-29

table permissions 10-18

view permissions 9-66

Guest users

creating 7-4

databases 7-4

Guidelines, backing up 7-38

H

Help

button 3-17

how to use xxix

menu 3-13

status bar 3-18

toolbar buttons 3-18

Hiding

bars 3-16

icons 2-9, 3-16

toolbars 2-9

Host, process 4-14

housekeeper free write percent configuration
parameter A-5

I

i/o accounting flush interval configuration
parameter A-6

i/o flush configuration parameter A-2

i/o polling process count configuration
parameter A-6

I/Os, large 5-6, 9-22, 10-6

Icons 1-2

container 3-2

displaying 2-9, 3-16

dragging and dropping B-3

filtering by name 3-9

hiding 2-9, 3-16

object 3-3

start-up 2-1

Icons, list of B-1 to B-3

identity burning set configuration
parameter A-2

identity burning set factor configuration
parameter A-6

IDENTITY column 7-21

Identity columns 9-56

identity grab size configuration
parameter A-6

Increasing

database storage space 7-9

Index Consistency Check dialog
box 9-26

Indexes

allocating pages 9-26

assigning columns 9-20

binding to a named cache 5-9

building 9-21

- cache replacement strategy 9-22
- checking consistency 9-26
- clustered 9-17, 9-20
- Create tab 9-18
- creating 9-18
- creating as primary key 9-20
- displaying 10-24
- estimating space requirements 7-2
- fill factor 9-21
- keys 9-20
- limiting page size 9-21
- maximum rows per page 9-19
- modifying 9-17
- navigating to 7-29, 10-25
- nonclustered 9-17
- nonunique 9-17
- primary key 10-32
- segments 7-28, 9-19
- Segments tab 9-25
- sorting data 9-21
- tables 9-19
- unique 9-17, 9-20
- values 9-20
- Index Properties tab 9-23
- Index Segments tab 9-25
- insert permission 8-25, 8-37, 9-68
- Integrity
 - checks 9-26
 - referential 9-47
- Interfaces file and remote servers 4-15
- ISQL... menu command 7-2
- isql options 2-2
- Issuing checkpoints 7-22

- K**
- Keys, index 9-20
- Killing user processes 4-15

- L**
- language in cache configuration
 - parameter A-2
- Languages configuration
 - parameters A-11
- Large buffer prefetch 9-22, 10-6
- Large I/Os 5-6, 9-22, 10-6
- Large Icon display format 3-6
- Large Icon toolbar button 3-17
- Last-chance threshold 7-29
- Limiting page size 9-21
- List display format 3-6
- List toolbar button 3-17
- Locking logins 8-12, 8-19
- Lock management configuration
 - parameters A-12
- lock promotion HWM configuration
 - parameter A-6
- lock promotion LWM configuration
 - parameter A-6
- lock promotion PCT configuration
 - parameter A-6
- locks configuration parameter A-2
- lock shared memory configuration
 - parameter A-6
- Log buffer size, changing 7-13
- Login Alias/Users tab 8-15
- Login Databases Owned tab 8-14
- Login Properties tab 8-11
- Logins
 - access 8-1
 - aliases 8-2
 - assigning aliases 8-14
 - creating 8-7
 - default database 8-9
 - deleting 8-9
 - displaying 8-10
 - language 8-9
 - locking 8-12, 8-19
 - modifying 8-11
 - remote server 4-19
 - roles 8-9, 8-11
 - sharing 8-14, 8-21
- Log only caches 5-6
- Logs
 - allocating space 7-2
 - data 6-17

- maintaining 6-1
- separating 6-17
- storing 6-2, 7-4
- transaction 6-2, 6-17
- logsegment* segment 7-23

M

- Maintaining logs 6-1
- Managing
 - objects 9-1
 - resources 6-1
- Mapping remote server logins 4-19
- Maps, allocation 10-37
- Masking characters 3-9
- master* database
 - backing up 7-6, 7-33
 - creating devices 6-2
- Master device 6-16
- max async i/o's per engine configuration
 - parameter A-6
- max async i/o's per server configuration
 - parameter A-6
- max engine freelocks configuration
 - parameter A-6
- Maximizing performance 6-17
- Maximum index rows per page 9-19
- maximum network configuration
 - parameter A-2
- Maximum table rows per page 10-5, 10-16
- max network packet size configuration
 - parameter A-6
- max number of network listeners configuration
 - parameter A-6
- max online engines configuration
 - parameter A-2
- max online engines configuration
 - parameter A-6
- Memory, process 4-14
- memory alignment boundary configuration
 - parameter A-7
- memory configuration parameter A-2
- Memory pools 5-6

- Memory use configuration
 - parameters A-12
- Menus
 - context-sensitive 3-13
 - pop-up 3-14
 - pull-down 3-14
 - shortcut 3-14
 - standard 3-12
- Menus and toolbars 3-12 to 3-19
- Messages
 - dialog boxes 3-24
 - SQL Server 11-13
- min online engines configuration
 - parameter A-2
- Mirroring
 - adding 6-11
 - benefits 6-10
 - costs 6-11
 - devices 6-4
 - disabling 6-13
 - disks 6-1, 6-10
 - recovery benefits 6-17
 - re-enabling 6-15
- model* database 6-3, 9-54
- Modes
 - gracefully 4-7
 - immediately 4-7
 - no wait 4-7
 - wait 4-7
- Modifying
 - default databases 8-11
 - default languages 8-11
 - groups 8-22
 - indexes 9-17
 - logins 8-11
 - procedures 9-29
 - rules 9-39
 - user datatypes 9-54
 - views 9-59
- Most Recently Used replacement
 - strategy 9-22, 10-6
- MRU replacement strategy 9-22, 10-6
- MS Windows
 - File Manager 2-37

Notepad 2-37
 Multivolume backup and restore 7-48

N

Named caches

- binding objects to 5-9
- changing size of 5-6
- changing type of 5-6
- creating 5-2
- creating a binding 5-10
- data and log 5-6
- defining buffer pools 5-6
- deleting 5-3
- deleting a binding 5-12
- displaying 5-5
- log only 5-6
- status 5-3, 5-9
- wash buffers 5-8

Named Pipes

- example *sql.ini* Server entry 11-8

Names, filtering 3-9

Navigating

- between open windows 3-26
- Voyager window lists 3-4

Navigating to

- aliases 8-22
- databases 6-8
- default dependencies 9-13
- devices 7-27
- dialog boxes 3-21
- indexes 7-29, 9-26, 10-25
- objects 1-2, 2-14, 8-38
- procedure dependencies 9-35
- procedure permissions 9-38
- rule dependencies 9-43
- segments 6-10, 9-26
- table dependencies 10-18
- tables 7-29
- trigger dependencies 9-53
- triggers 10-25
- user datatype dependencies 9-58
- users 8-17, 8-35
- users and groups 9-38, 9-69, 10-21

- view dependencies 9-66

nested triggers configuration

- parameter A-2

- Network... button 4-24, 4-25, 7-14, 7-15

Network communications configuration

- parameters A-12

- Network I/O tasks, checking 4-29

- Network packet size 4-14

- New window, opening 3-5

- New Window... command 3-5

- New Window toolbar button 3-17

- No Checkpoint on Recovery database

- option 7-20

- No Free Space Accounting database

- option 7-21

- Nonclustered indexes 9-17

- Nonunique indexes 9-17

- No wait mode 4-7

- Null values 9-56

- number of alarms configuration

- parameter A-7

- number of devices configuration

- parameter A-7

- number of extent i/o buffers configuration

- parameter A-7

- number of index trips configuration

- parameter A-7

- number of languages in cache configuration

- parameter A-7

- number of locks configuration

- parameter A-7

- number of mailboxes configuration

- parameter A-7

- number of messages configuration

- parameter A-7

- number of oam trips configuration

- parameter A-7

- Number of objects displayed 3-18

- number of open databases configuration

- parameter A-7

- number of open objects configuration

- parameter A-7

- number of preallocated extents configuration

- parameter A-7

- number of remote connections configuration
 - parameter A-7
- number of remote logins configuration
 - parameter A-8
- number of remote sites configuration
 - parameter A-8
- number of sort buffers configuration
 - parameter A-8
- number of user connections configuration
 - parameter A-8

Numbers

- default character set ID A-4

O

- Object display, sorting 3-7
- Object icons 3-3
- Objects
 - and Contents bar 9-5
 - batch files defining 2-36
 - creating 9-2
 - database, displaying 9-5
 - deleting 2-38, 9-7
 - dialog boxes 3-20
 - displaying details 9-3
 - displaying ownership 8-23
 - group permissions 8-36
 - hiding 1-3
 - icons 1-2
 - managing 9-1
 - navigating to 1-2, 2-14, 8-38
 - owner permissions 8-4
 - owners 9-48
 - permissions 8-5, 8-36
 - properties 9-4
 - recreating definitions 2-35
 - relative to other objects 9-6
 - sorting Details display 3-7
 - SQL Server, displaying 9-5
 - user permissions 8-24
 - visual representation 1-2
- Oldest active transaction 4-15
- Online help xxix
- Open Connection dialog box 11-14

- open databases configuration
 - parameter A-2
- Opening a new window 3-5
- open objects configuration parameter A-2
- Operating system resources
 - configuration parameters A-12
- Operator permissions 8-3
- Options
 - command line 2-2
 - dbcc 7-52
 - default 7-18
 - isql 2-2
 - required 2-3
 - starting SQL Server Manager 2-2
 - wisql 11-13
- Options... menu command 7-19
- Output dialog box 3-24
- Ownership
 - displaying 8-23
 - permissions and 8-3
 - transferring databases 7-12

P

- Packet size, network 4-14
- page lock spinlock ratio configuration
 - parameter A-8
- Pages
 - allocating 9-26, 10-37
 - limiting size 9-21
- page utilization configuration
 - parameter A-8
- Parameters
 - adding 9-30
 - removing 9-31
- partition groups configuration
 - parameter A-8
- Partitioning a table 10-27
- partition spinlock ratio configuration
 - parameter A-8
- Password, encrypting 4-18
- password expiration configuration
 - parameter A-2
- Percentage, fill factor 9-21

- Performance
 - benefits 6-11
 - costs 6-11
 - maximizing 6-17
 - segment benefits 7-23
- perform disk i/o on engine 0 configuration
 - parameter A-8
- permission cache entries configuration
 - parameter A-8
- Permissions
 - access 8-1, 8-46
 - assigning 2-31
 - command 8-4, **8-42 to 8-45**
 - creating referential constraints 8-5, 8-25, 8-37
 - database owners 8-3
 - default 8-41 to 8-45
 - deleting a row 8-5, 8-25, 8-37
 - deleting a view 9-68
 - dialogs 8-46
 - execute 9-38
 - executing a procedure 8-5
 - granting 2-33, 8-40, 9-38
 - groups 8-36
 - inserting a row 8-5, 8-25, 8-37, 9-68
 - modifying 9-36, 9-66, 10-18
 - object 8-5, 8-24, 8-36
 - object owners 8-4
 - Operators 8-3
 - procedure 9-36
 - Public group 8-42 to 8-45
 - referential constraint referencing a view 9-68
 - revoking 8-40, 9-38
 - roles 8-1
 - selecting a table or view 8-5, 8-25, 8-37
 - selecting a view 9-68
 - sharing 8-2, 8-14, 8-21, 8-29
 - shortcuts 9-69, 10-21
 - summary 8-41 to 8-45
 - System Administrators 8-2
 - System Security Officers 8-2
 - table 10-18
 - tabs 2-32, 8-46
 - transfers and 8-41
 - updating 9-66
 - updating a row 8-5, 8-25, 8-37, 9-68
 - view 9-66
 - with cascade 8-5
 - with cascade 9-38
 - with grant 8-5
 - with grant 9-38
- Physical memory configuration
 - parameters A-13
- Planning
 - backup 7-33
 - recovery 7-33, 7-44
- Pools, buffer 5-6
- Pop-up menu commands 3-15
- Pop-up menus 3-14
- pre-read packets configuration
 - parameter A-2
- Primary key 9-20
- Primary keys 10-32, 10-34
- print deadlock information configuration
 - parameter A-9
- print recovery information configuration
 - parameter A-9
- Privileges. *See* Permissions.
- Procedure cache 5-1
- procedure cache configuration
 - parameter A-3
- procedure cache size configuration
 - parameter A-9
- Procedure Dependencies tab 9-35
- Procedure Permissions tab 9-37
- Procedure Properties tab 9-33
- Procedures
 - adding parameters 9-30
 - Create dialog box 9-30
 - creating 9-29
 - Dependencies tab 9-34
 - displaying 9-35, 10-18
 - displaying dependencies 9-34
 - executing 9-36
 - modifying 9-29
 - navigating to dependencies 9-35

- permissions 9-36
- Permissions tab 9-37
- recompiling 10-34
- sp_estspace 7-2
- Process details 4-14
- Processes
 - displaying 4-13
 - killing 4-13
- Processors configuration
 - parameters A-13
- Program requesting, process 4-15
- Properties
 - displaying database 7-7
 - displaying default 9-12
 - displaying dump device 7-37
 - displaying segment 7-25
 - object 9-4
 - tab 9-4
- Properties toolbar button 3-17
- Public
 - group permissions 8-42 to 8-45
 - membership 8-18
- pubs2* sample database 2-7
- Pull-down menus 3-14

- R**
- Read Only Database database
 - option 7-20
- Recompiling procedures and triggers 10-34
- Recovery
 - clean 6-16
 - configuration parameters A-11
 - database 7-44
 - mirroring benefits 6-10, 6-17
 - planning 7-33, 7-44
 - separating logs 6-17
 - strategies 6-1
 - transaction log 7-44
- recovery flags configuration
 - parameter A-3
- recovery interval configuration option A-3
- recovery interval in minutes configuration
 - parameter A-9
- Recreating object definitions 2-35
- Reducing database size 7-10
- Re-enabling disk mirroring 6-15
- references 8-5
- references permission 8-25, 8-37, 9-68
- Referential constraints 10-2, 10-3
 - description of 10-30
 - foreign keys in 10-32
 - primary keys in 10-32, 10-34
- Referential integrity, enforcing 9-47, 10-30
- Refresh button 3-25
- Refreshing windows and dialog boxes 3-25
- Refresh toolbar button 3-17
- remote.ddl* file 4-26
- remote access configuration
 - parameter A-3
- remote connections configuration
 - parameter A-3
- Remote login mapping 4-19
- remote logins configuration
 - parameter A-3
- Remote procedure calls (RPCs) 4-15
- remote server pre-read packets configuration
 - parameter A-9
- Remote Server Properties dialog box 4-22
- Remote servers
 - communicating with 4-15 to 4-23
 - creating 4-16
 - deleting 4-18
 - displaying 4-22
 - interfaces file and 4-15
 - login management 4-19
 - options 4-17
 - setting up for 4-15
- remote sites configuration parameter A-3
- Remove button 7-4, 9-31, 10-8, 10-14
- Removing
 - columns 10-8, 10-14
 - device allocation 7-4

- parameters 9-31
 - users from groups 8-31
 - Replacement strategy, MRU 9-22
 - Representation, visual 1-2
 - Requesting program, process 4-15
 - Resetting configuration
 - parameters 4-10, 4-12
 - Restoring a database
 - from multiple volumes 7-48
 - volume change prompts and 7-48
 - Retrieving data 9-17
 - Revoking
 - access 8-46
 - command permissions 8-28, 8-40
 - object permissions 9-38
 - procedure permissions 9-36
 - table permissions 10-18
 - view permissions 9-66
 - with cascade 8-5
 - with cascade 9-38
 - Roles
 - access 8-1, 8-46
 - assigning 2-21, 8-9, 8-11
 - configuring SQL Servers 4-9
 - resetting Configuration options 4-9
 - System Administrator 4-24, 4-25
 - System Administrators 4-9, 6-1
 - System Security Officers 4-9
 - Rule Bindings tab 9-45
 - Rule Dependencies tab 9-43
 - Rule Properties tab 9-42
 - Rules
 - as constraints 10-2
 - binding 9-45
 - Bindings tab 9-45
 - Create dialog box 9-40
 - creating 9-40
 - defining values 9-39
 - Dependencies tab 9-43
 - displaying 10-18
 - displaying dependencies 9-43
 - expression in definition of 9-41
 - modifying 9-39
 - navigating to dependencies 9-43
 - unbinding 9-47
 - ways of creating 9-31, 9-39, 10-2
 - runnable process search count configuration
 - parameter A-9
 - Running sp_estspace 7-2
- ## S
- Scheduling back ups 7-38
 - Schemas
 - creating objects 9-4
 - DDL 2-35
 - Scripts, DDL
 - editing 4-26 to 4-28, 7-16 to 7-18
 - files containing 4-26 to 4-28, 7-16 to 7-18
 - generating 4-23
 - Searching. *See* Finding.
 - Search strings
 - case sensitivity 3-9
 - masking characters 3-9
 - wildcard characters 3-9
 - Segment Contains tab 7-28
 - Segment Properties tab 7-26
 - Segments
 - creating 7-23
 - databases 6-9
 - definition 7-23
 - deleting 7-25
 - devices 6-9
 - displaying 9-25
 - displaying properties 7-25
 - extending 7-27
 - indexes on 7-28, 9-19
 - navigating to 6-10
 - navigating to devices 7-27
 - navigating to objects 7-29
 - performance benefits 7-23
 - storing databases 7-23
 - tables on 7-28
 - thresholds on 7-29 to 7-33
 - Segment Thresholds tab 7-31
 - select
 - permission, granting 2-33

- statement 9-61
- Selecting display formats 3-6
- select permission 8-25, 8-37, 9-68
- Separating
 - disk usage 6-17
 - logs 6-17
 - performance benefits 6-17
- Server Consistency Check dialog box 4-29
- Server Generate DDL dialog box 4-24
- Server menu 3-13
- Server Processes dialog box 4-14
- Server Properties dialog box 4-6
- Server Run DDL dialog box 4-25
- Servers
 - communicating with 11-12
 - configuring 4-8
 - connecting 4-1
 - displaying properties 4-6
 - killing processes 4-15
 - messages 11-13
 - timeout values 11-13
- Server user ID 8-12
- Set Database Online command 7-55
- Setting a database online 7-55
- shared memory starting address configuration parameter A-9
- Sharing
 - logins 8-2, 8-14, 8-21
 - permissions 8-2, 8-14, 8-21, 8-29
- Shortcut menu commands 3-15
- Shortcut menus 3-14
- Shortcuts xxvi
 - connecting SQL Server 2-5, 4-4
 - drag-and-drop 3-10
 - navigating to segments 6-10
 - updating permissions 9-69, 10-21
- Single-User Mode database option 7-20
- Size
 - estimating database 7-1
 - increasing database 7-9
 - reducing database 7-10
- size of auto identity column configuration parameter A-9
- Sorting data 9-21
- Sorting object details 3-7
- sort page count configuration parameter A-9
- sp_estspace procedure 7-2
 - estimating space requirements 7-2
- Space
 - allocating on segments (tables) 10-22
- spid, blocking 4-14
- sql.ini files
 - editing 11-2
- SQLEdit window 11-2
- SQL Server 11.0 features
 - Add Identity Column in Non-Unique Indexes database option 7-21
 - changing database log buffer size 7-13
 - checking network I/O 4-29
 - configuration 4-11
 - how described xxvi, xxviii
 - maximum index rows per page 9-19
 - maximum table rows per page 10-5, 10-16
 - named cache management 5-1 to 5-12
 - process information 4-15
 - table partitioning 10-27
 - table unpartitioning 10-29
- SQL Server administration
 - configuration parameters A-13
- sql server clock tick length configuration parameter A-9
- SQL Server Manager
 - about 1-1
 - connecting 2-4
 - customizing display of 3-6
 - exiting 2-4
 - features of 1-1
 - multiple logins and 3-3
 - windows and dialog boxes of 3-1
- SQL Server Reference Manual* xxv, 7-21
- SQL Server Utility Programs* manual xxiv, 4-8
- SQL statements 9-4, 9-31, 9-49
- ssm executable 2-2

- stack guard size configuration
 - parameter A-9
 - stack size configuration parameter A-3, A-9
 - Standard configuration parameter
 - display level 4-11
 - Standard menus 3-12
 - Standard toolbar 3-16
 - Start time, transaction 4-15
 - Start-up icon 2-1
 - Statements
 - SQL 9-4, 9-49
 - Statistics, updating 10-39
 - Status bar 3-18
 - command help 3-18
 - filtering and 3-9, 3-18
 - object count 3-18
 - Stopping `wisql` 11-13
 - Storing
 - allocating space 7-2
 - considerations 6-16, 7-2
 - costs 6-11
 - databases 6-2
 - default devices 6-3
 - increasing space 7-9
 - logs 6-2
 - segment benefits 7-23
 - strategies 6-16
 - transaction logs 7-4
 - Strategies
 - recovery 6-1
 - storing 6-16
 - `suid`. *See* Server user ID.
 - Summary of constraints 10-3
 - Sybase Troubleshooting Guide* xxiv
 - `sybssystemprocs` database 6-3
 - Symmetrical drag-and-drop 3-10
 - `sysdevices` table 6-2
 - System Administration Guide* xxv, 7-33, A-1
 - System Administration Guide Supplement* 4-16
 - System administration training xxv
 - System Administrators
 - changing database defaults 7-18
 - configuring SQL Servers 4-9
 - generating DDL scripts 4-24, 4-25
 - issuing checkpoints 7-22
 - managing resources 6-1
 - permissions 8-2
 - transferring database ownership 7-12
 - System Security Officers 8-7
 - configuring SQL Servers 4-9
 - permissions 8-2
 - remote login mapping 4-19
 - `system` segment 7-23
 - System table 10-1
 - System tables, updates to A-4
 - systemwide password expiration
 - configuration parameter A-9
- T**
- Table, system 10-1
 - Table, user 10-1
 - Table check constraint 9-39
 - Table Check Constraints tab 10-26
 - Table Column Advanced dialog
 - box 10-9, 10-15
 - Table Consistency dialog box 10-35
 - Table Dependencies tab 3-21, 10-17
 - Table Indexes/Triggers tab 9-6, 10-24
 - table lock spinlock ratio configuration
 - parameter A-9
 - Table Partitions tab 10-28, 10-29
 - Table Permissions tab 10-19
 - Table Properties tab 3-20, 9-4, 10-12
 - Table Referential Constraints tab 10-31
 - Tables
 - adding columns 10-13
 - allocating pages 10-37
 - assigning permissions 2-31
 - binding to a named cache 5-9
 - cache replacement strategy 10-6
 - check constraints 10-10, 10-16, 10-25, 10-27
 - column constraints 10-14
 - column definitions 10-7

- Create dialog box 10-4
- creating 2-28, 10-1, 10-4
- deleting 10-11
- Dependencies tab 10-17
- displaying 9-35, 10-18
- displaying dependencies 10-17
- estimating space requirements 7-2
- indexes 9-19
- Indexes/Triggers tab 10-24
- maximum rows per page 10-5, 10-16
- navigating to 7-29
- navigating to dependencies 10-18, 10-24
- partitioning 10-27
- permissions 10-18
- Permissions tab 10-19
- referential constraints 10-30
- removing columns 10-8, 10-14
- segments 7-28
- Segments tab 10-23, 10-24
- shortcuts 10-21
- sysdevices* 6-2
- unpartitioning 10-29
- updating statistics 10-39
- Table Segments tab 10-23
- Tabs
 - access 8-46
 - binding objects 9-15
 - buttons 2-16, 3-20, 9-6
 - dialog boxes 2-16, 3-20
 - displaying dependencies 9-13
 - displaying properties 2-16, 9-4
 - granting permissions 2-32, 8-46
- tape retention configuration
 - parameter A-3
- tape retention in days configuration
 - parameter A-9
- Tasks, administration 1-1, 2-25
- tcp no delay* configuration parameter A-10
- Technical Support, contacting xxx
- tempdb* database 6-3
- Thresholds, segment
 - creating 7-30
 - deleting 7-32
 - hysteresis value 7-30
 - last chance 7-29
 - modifying 7-32
 - purpose of 7-29
- Time blocked, process 4-14
- Timeout values, SQL Server 11-13
- time slice configuration parameter A-3, A-10
- Toolbar, standard 3-16
- Toolbar buttons 3-16
- Toolbars
 - creating objects 1-3
 - displaying 2-9
 - hiding 2-9, 3-16
- Tools menu 3-13
- ToolTip 3-18
- total data cache size configuration
 - parameter A-10
- total memory configuration
 - parameter A-10
- Training xxv
- Transaction, active 4-15
- Transaction log backup options 7-40
- Transaction logs 6-2, 6-17
 - allocating space for 7-2
 - backing up 7-20
 - checkpoints 7-22
 - dumping 7-20
 - recovering 7-44
 - storing 7-4
- Transaction start time 4-15
- Transact-SQL commands 4-23
 - DDL 7-13
- Transferring database ownership 7-12
- Trigger Dependencies tab 9-53
- Trigger Properties tab 9-52
- Triggers
 - commands 9-47
 - Create dialog box 9-48
 - creating 9-48
 - Dependencies tab 9-53
 - displaying 9-35, 10-18, 10-24, 10-25
 - displaying dependencies 9-53
 - enforcing referential integrity 9-47

- executing 9-49
 - navigating to 10-25
 - navigating to dependencies 9-53
 - nested A-4
 - object owners 9-48
 - recompiling 10-34
 - Truncate and Log transaction log
 - backup option 7-40
 - Truncate and No Log transaction log
 - backup option 7-40
 - Truncate Transaction Log on Checkpoint
 - database option 7-20
 - Tutorial 2-7
 - assigning object permissions 2-25, 2-31
 - creating logins 2-20
 - creating tables 2-28
 - creating users 2-25
 - deleting databases 2-38
 - displaying toolbars and icons 2-9
 - exiting SQL Server Manager 2-39
 - generating DDL 2-25, 2-35
 - hiding toolbars and icons 2-9
 - navigating between objects 2-14
 - using drag-and-drop 2-25
 - using tabs 2-16
- U**
- Unbinding
 - columns 9-16, 9-47
 - datatypes 9-47
 - defaults 9-16
 - objects from a named cache 5-12
 - user datatypes 9-16
 - Unique
 - indexes 9-17, 9-20
 - values 9-20
 - Unpartitioning a table 10-29
 - update permission 8-25, 8-37, 9-68
 - Updating permissions
 - procedures 9-36
 - tables 10-18
 - views 9-66
 - Updating statistics 10-39
 - upgrade version configuration
 - parameter A-3, A-10
 - Up toolbar button 3-17
 - Usable by Database Owner Only
 - database option 7-20
 - user.ddl* file 2-37
 - User Command Permissions tab 8-28
 - user connections configuration
 - parameter A-3
 - User Datatype Dependencies tab 9-58
 - User Datatype Properties tab 9-57
 - User datatypes
 - binding defaults 9-15
 - binding rules 9-45
 - Create dialog box 9-55
 - creating 9-55
 - Dependencies tab 9-58
 - displaying 9-13, 9-35, 10-18
 - displaying dependencies 9-58
 - modifying 9-54
 - navigating to dependencies 9-58
 - unbinding 9-16, 9-47
 - User-defined datatypes. *See* User datatypes
 - User environment configuration
 - parameters A-14
 - user log cache size configuration
 - parameter A-10
 - user log cache spinlock ratio configuration
 - parameter A-10
 - User Object Permissions tab 8-24
 - User Objects Owned tab 8-23
 - User permissions
 - procedures 9-36
 - tables 10-18
 - views 9-66
 - User processes, displaying 4-13
 - User Properties tab 8-21
 - Users
 - assigning aliases 8-21
 - changing groups 8-22
 - creating 2-25, 8-18
 - deleting 8-19

- displaying 8-20
- displaying objects owned 8-23
- granting command permissions 8-28
- granting object permissions 8-24, 8-36
- guest 7-4
- navigating to 8-17, 8-35, 9-38, 9-69, 10-21
- removing from groups 8-31

User table 10-1

using online help xxix

Utilities

- bcp 7-20
- dbcc 7-50, 10-35
- wisql 7-2, 11-12
- writetext 7-20

V

Values

- default column 9-8
- index keys 9-20
- null 9-56
- rules defining 9-39
- timeout 11-13
- unique 9-20

View Dependencies tab 9-65

View Permissions tab 9-67

View Properties tab 9-64

Views

- adding columns 9-59
- Create dialog box 9-59
- creating 9-59
- Dependencies tab 9-65, 9-66
- displaying 9-35, 10-18
- displaying dependencies 9-65
- modifying 9-59
- navigating to dependencies 9-66, 9-69
- permissions 8-43, 9-66
- Permissions tab 9-67

Visual representation 1-2

Volume change prompts 7-42, 7-48

Voyager window 2-2, 3-2

- container icons 3-2
- Contents bar 9-5

- customizing the display of 3-6 to 3-9
- description of 3-1 to 3-11
- dragging and dropping 3-10 to 3-11
- features 3-2 to 3-3
- logins and 3-3
- menus and toolbars 3-12 to 3-19
- moving through 3-4 to 3-5
- object icons 3-3
- opening new windows 3-5 to 3-6
- refreshing 3-25

W

Wait mode 4-7, 4-8

Wash area 5-8

When ready mode 4-8

Wildcard characters 3-9

Window menu 3-13

Windows

- File Manager 2-37
- Notepad 2-37
- status bar 3-18

WISQL Options dialog box 11-13

wisql utility

- connecting SQL Servers 11-14
- disconnecting SQL Servers 11-15
- estimating space requirements
 - with 7-2
- options 11-13
- running 11-12
- stopping 11-13

WISQL window 11-12

With cascade 8-5

with cascade permission 9-38

With grant 8-5

with grant permission 9-38

writetext utility 7-20

