

Enterprise SQL Server Manager™ User's Guide

Enterprise SQL Server Manager Release 11.0

Document ID: 32492-01-1100-02

Last Revised: May 31, 1996

Principal author: Systems Management Products Technical Publications

Contributing authors: Systems Management Products and Technical Support staff

Document ID: 32492-01-1100

This publication pertains to Enterprise SQL Server Manager Release 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 © 1989–1996 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, S-Designor, SQL Advantage, SQL Debug, SQL SMART, SQL Solutions, SQR, SYBASE, the Sybase logo, Transact-SQL, and VQL are registered trademarks of Sybase, Inc. ADA Workbench, AnswerBase, Application Manager, APT-Build, APT-Edit, APT-Execute, APT-Library, APT-Translator, APT Workbench, Backup Server, Bit-Wise, Client-Library, 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, ExElerator, 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, ObjectCONNECT, OmniCONNECT, OmniSQL Access Module, OmniSQL Gateway, OmniSQL Server, OmniSQL Toolkit, Open Client, Open Client CONNECT, Open

Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open Server CONNECT, Open Solutions, PC APT-Execute, PC DB-Net, PC Net Library, Powersoft Portfolio, Powersoft Professional, 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, Warehouse WORKS, Watcom SQL, web.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

Audience	xxix
How to Use This Book	xxix
Related Documents	xxx
Other Documents	xxxii
Other Sources of Information	xxxii
Conventions	xxxiii
Style Conventions	xxxiii
Shortcuts	xxxiii
Roles	xxxiv
Mouse Buttons	xxxiv
SQL Server 11.0 Features	xxxv
If You Need Help	xxxvi
Using Online Help	xxxvi
Contacting Technical Support	xxxvii

1. Introduction

What's in This Chapter	1-1
Managing Individual SQL Server Installations	1-1
Performing System and Database Administration Tasks	1-2
Making It Easy	1-3
Managing the Enterprise	1-4
The Enterprise SQL Server Manager User Interface	1-5
Graphical User Interface	1-5
Command Line Interface	1-6
Enterprise SQL Server Manager Policy Region Resources	1-7
Profile Management	1-8
Default Policy and Validation Policy	1-8
Event Monitoring Services	1-9
Notices	1-10

2. Setting Up the Tivoli Management Environment for ESSM

What's in This Chapter	2-1
Preparing the TME to Manage SQL Servers	2-1
Assigning Enterprise SQL Server Manager Roles to Administrators	2-2
Tivoli Management Region (TMR) and Resource Roles	2-3

Assigning SQL Server Logins for Administrators	2-6
SQL Server Logins for Multiple UNIX Logins.....	2-6
SQL Server Logins for Multiple Management Levels.....	2-6
SQL Server Roles	2-7
Assigning ESSM Administrator SQL Server Logins for the Enterprise	2-7
Assigning ESSM Administrator SQL Server Logins for Policy Regions	2-9
Assigning ESSM Administrator SQL Server Logins for Managed SQL Servers.....	2-10
Configuring Policy Regions for Managed SQL Server Resources.....	2-11
Creating a Managed SQL Server Resource in a Policy Region	2-13
Enterprise SQL Server Manager Management Hosts	2-13
Creating a Managed SQL Server Resource.....	2-16

3. Enterprise SQL Server Manager Windows and Dialog Boxes

What's in This Chapter	3-1
Starting Enterprise SQL Server Manager	3-1
Stopping Enterprise SQL Server Manager	3-1
Using the Voyager Window	3-2
Features of Voyager	3-2
Moving Through a Voyager Window	3-4
Opening Another Voyager Window	3-5
Customizing the Display.....	3-6
Selecting Display Formats	3-6
Filtering the Display	3-8
Using Drag-and-Drop Shortcuts	3-9
Using Drag-and-Drop Between Windows	3-10
Menus and Toolbars	3-11
Standard Menus	3-11
Context-Sensitive Menus.....	3-13
Pull-Down Menus	3-13
Shortcut Menus	3-14
Toolbar and Status Bar.....	3-16
Standard Toolbar	3-16
Status Bar.....	3-19
Types of Dialog Boxes	3-20
Object Dialog Boxes	3-20
Using Dialog Box Tabs	3-20
Selecting Multiple Objects	3-22
Navigating Between Objects	3-23

Closing an Object Dialog Box	3-23
Command Dialog Boxes	3-24
Closing a Command Dialog Box	3-24
Confirmation Dialog Boxes	3-25
Closing a Confirmation Dialog Box	3-25
Message Dialog Boxes	3-25
Closing a Message Dialog Box	3-26
Refreshing Window and Dialog Box Displays	3-27
Moving Between Open Windows and Dialog Boxes	3-28

4. Managing the SQL Server Operating Environment

What's in This Chapter	4-1
Managing SQL Servers on Non-ESSM Hosts	4-1
Starting SQL Server or Backup Server	4-2
Displaying the Status of SQL Server	4-3
Stopping SQL Server or Backup Server	4-4
Stopping SQL Server from a Policy Region Window	4-4
Stopping SQL Server from a SQL Server Window	4-5
Starting SQL Server Monitor from Enterprise SQL Server Manager	4-6
Starting the SQL Server Monitor Server	4-7
Customizing SQL Server Monitor Launch	4-8
The SQL Server Window	4-10
Displaying SQL Server Details	4-12
Examining SQL Server Properties	4-12
Examining SQL Server Objects	4-13
Configuring SQL Server	4-14
Parameters Requiring Restart	4-15
Configuring SQL Server Release 10.x	4-15
Configuring SQL Server Release 11.0	4-17
Setting Configuration Parameters	4-18
Displaying and Killing SQL Server Processes	4-20
Displaying Processes	4-20
Viewing Additional Process Details	4-21
Viewing SQL Server Release 11.0 Process Information	4-21
Killing a Process	4-22
Displaying SQL Server Locks and Statistics	4-22
Enabling Communication with Other Servers	4-23
Creating a Remote Server	4-24
Deleting a Remote Server	4-26
Managing Remote Server Logins	4-26

Setting Up Explicit Mappings	4-29
Displaying Remote Server Details	4-30
Generating and Executing SQL Server DDL Scripts.....	4-32
Generating SQL Server DDL.....	4-33
Executing SQL Server DDL.....	4-34
Examining and Editing DDL Script Files	4-35
Checking Consistency in SQL Server Release 11.0	4-38
Auditing SQL Server	4-40

5. Managing Cache

What's in This Chapter	5-1
Creating and Deleting a Named Data Cache.....	5-2
Creating a Cache.....	5-2
Deleting a Cache.....	5-4
Displaying and Modifying Cache Information	5-4
Displaying Cache Properties.....	5-5
Changing Cache Size or Type.....	5-6
Defining Buffer Pools.....	5-7
Creating a Buffer Pool.....	5-9
Deleting a Buffer Pool.....	5-10
Binding Objects to a Cache.....	5-10
Rules for Binding	5-11
Creating a Binding	5-12
Deleting a Binding	5-14

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.....	6-3
Deleting a Database Device.....	6-6
Displaying and Modifying Device Information	6-6
Examining Database Device Properties.....	6-7
Examining Allocations on a Database Device	6-8
Navigating to a Database	6-9
Examining Segment Mapping on a Database Device	6-10
Navigating to a Segment.....	6-11
Using Disk Mirroring.....	6-12
Adding Disk Mirroring.....	6-14

Storage Management Considerations and Tips	6-19
Managing for Recovery	6-19
Changing Default Device Designation	6-19
Using Disk Mirroring	6-20
Separating Data and Transaction Log	6-20
Managing for Performance	6-20

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	7-2
Deciding Where to Allocate Storage	7-2
Creating a Database	7-3
Deleting Databases	7-6
Deleting a Database	7-6
Deleting a Damaged Database	7-7
Displaying and Modifying Database Information	7-7
Examining Database Details	7-7
Modifying Database Storage Allocations	7-9
Allocating Space on the Same Device	7-9
Allocating Space on a New Device	7-10
Examining Database Storage Allocations	7-10
Transferring Database Ownership	7-12
Changing Log Buffer Size	7-12
Generating and Executing Database DDL Scripts	7-13
Location of DDL Script Files	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
List of Options	7-20
Issuing a Database Checkpoint	7-22
Using Database Segments	7-23
Creating and Deleting Segments	7-24
Creating a Segment	7-24
Deleting a Segment	7-26
Displaying and Modifying Segment Information	7-27
Examining Segment Details	7-28
Navigating to a Database Device	7-29

Extending a Segment	7-29
Examining Objects Stored on a Segment	7-30
Navigating to an Index or Table on a Segment	7-31
Using Segment Thresholds	7-31
Threshold Concepts: Last-Chance Threshold	7-32
Threshold Concepts: Hysteresis Value	7-32
Creating a Threshold	7-32
Deleting a Threshold	7-34
Modifying a Threshold	7-35
Preparing for Backup and Recovery	7-36
Developing a Backup and Recovery Plan	7-36
Identifying the Backup Server	7-37
Starting Backup Server	7-37
Stopping Backup Server	7-38
Managing Dump Devices	7-38
Creating a Dump Device	7-39
Deleting a Dump Device	7-40
Examining Dump Device Properties	7-41
Backing up a Database	7-42
Scheduling Database Backups	7-42
Guidelines for Backing up User Databases	7-42
Using the Tivoli Scheduler to Schedule Backups	7-43
Guidelines for Backing up the <i>master</i> Database	7-43
Using the Database Backup Dialog Box	7-44
Responding to Volume Change Prompts	7-47
Restoring a Database	7-49
Deciding How to Restore	7-49
Recovery Considerations	7-49
Using the Database Restore Dialog Box	7-50
Responding to Volume Change Prompts	7-53
To Get Information About Backup Server Messages	7-54
Checking Database Consistency	7-55
Consistency Check Options	7-57
Overall Database Option	7-57
Database Allocations Option	7-58
System Catalog Option	7-59

Setting a Database Online	7-60
-------------------------------------	------

8. Controlling Access

What's in This Chapter	8-1
About Access Privileges and Permissions	8-2
Access Based on Administrative Role	8-3
Access Based on Special User Status	8-4
Access Based on Group Membership	8-5
Access Based on Command Permissions	8-5
Access Based on Object Permissions	8-6
Access Example.	8-8
Creating and Deleting SQL Server Logins	8-9
Creating a SQL Server Login.	8-9
Deleting a SQL Server Login.	8-11
Displaying and Modifying Login Information	8-11
Modifying Login Information.	8-12
Locking a SQL Server Login.	8-13
Changing the Login Password.	8-14
Displaying Databases Owned by a Login.	8-15
Sharing Logins by Using Aliases	8-15
Adding an Alias to a Login.	8-16
Deleting an Alias from a Login	8-17
Navigating to Users.	8-18
Creating and Deleting Users	8-19
Deleting a User	8-21
Displaying and Modifying User Information.	8-22
Displaying User Properties	8-23
Modifying Group Membership and User Aliases	8-24
Navigating to Logins (Aliases).	8-24
Displaying Objects Owned by the User	8-25
Navigating to an Owned Object	8-25
Granting and Revoking Object Permissions for a User.	8-26
Navigating to Object Permissions Information	8-29
Granting and Revoking Command Permissions for a User.	8-29
Creating and Deleting Groups	8-31
Creating a Group	8-31
Deleting a Group	8-33
Removing Users from a Group	8-34
Deleting an Empty Group.	8-35
Displaying and Modifying Group Information	8-35
Displaying Group Properties	8-36

Navigating to Users	8-36
Modifying Group Membership	8-37
Granting and Revoking Object Permissions for a Group	8-37
Navigating to Object Permissions Information	8-40
Granting and Revoking Command Permissions for a Group	8-41
Summary of Access Privileges and Permissions.	8-43
Summary of Transferable Privileges and Permissions.	8-48

9. Managing Database Objects

What's in This Chapter	9-1
Operations Common to All Objects	9-2
Creating an Object	9-2
Closing an Object's Dialog Box.	9-4
Displaying Details About an Object.	9-4
Displaying Object Properties	9-4
Displaying SQL Server and Database Objects	9-6
Displaying an Object Relative to Other Objects.	9-7
Copying an Object	9-8
Deleting an Object.	9-8
Comparing Objects.	9-9
Managing Defaults	9-10
Creating a Default.	9-11
Example.	9-12
Displaying Default Properties	9-13
Displaying Default Dependencies	9-14
Navigating to Objects with Dependencies on a Default.	9-14
Binding a Default	9-15
Binding a Default to a Table Column	9-16
Binding a Default to a User Datatype	9-17
Deleting a Binding	9-18
Managing Indexes	9-19
Creating an Index:	9-20
Displaying Index Properties:	9-25
Placing an Index on a Segment.	9-26
Navigating to a Segment.	9-27
Checking Index Consistency.	9-28
Updating Statistics	9-30
Managing Procedures	9-31
Creating a Procedure	9-31
Displaying Procedure Properties	9-34

Displaying Procedure Dependencies	9-35
Navigating to Objects with Dependencies on a Procedure	9-36
Updating User and Group Procedure Permissions	9-36
Navigating to Users and Groups	9-39
Managing Rules	9-40
Creating a Rule	9-41
Displaying Rule Properties	9-42
Displaying Rule Dependencies	9-43
Navigating to Objects with Dependencies on a Rule	9-44
Binding a Rule	9-44
Binding a Rule to a Table Column	9-45
Binding a Rule to a User Datatype	9-46
Deleting a Binding	9-47
Managing Triggers	9-48
Creating a Trigger	9-49
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	9-55
Displaying User Datatype Properties	9-57
Displaying User Datatype Dependencies	9-58
Navigating to Objects with Dependencies on a User Datatype	9-59
Managing Views	9-60
Creating a View	9-61
Example	9-63
Displaying View Properties	9-65
Displaying View Dependencies	9-66
Navigating to Objects with Dependencies on the View	9-67
Updating User and Group View Permissions	9-68
Displaying the Information	9-68
Setting the Permissions	9-70
Navigating to Users and Groups	9-71

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	10-3

Implementing Referential Constraints	10-3
Constraints Summary	10-4
Procedure for Creating a Table	10-5
Specifying Column Definitions	10-8
Specifying Column Constraints	10-9
Deleting a Table	10-12
Displaying Table Properties	10-13
Adding Columns to a Table	10-15
Specifying Column Definitions	10-16
Specifying Column Constraints	10-17
Check Constraint Format	10-19
Modifying a Column's Datatype	10-19
Setting Maximum Rows Per Page	10-19
Setting Cache Strategy	10-20
Displaying Table Dependencies	10-21
Updating User and Group Table Permissions	10-22
Displaying the Information	10-22
Setting the Permissions	10-24
Navigating to Users and Groups	10-25
Placing a Table on a Segment	10-26
Navigating to Segments	10-27
Displaying Indexes and Triggers	10-28
Navigating to Indexes and Triggers	10-29
Creating a New Index or Trigger	10-29
Creating Check Constraints on a Table	10-30
Check Constraint Format	10-31
Partitioning and Unpartitioning a Table	10-32
Rules for Partitioning	10-32
Partitioning	10-33
Unpartitioning	10-34
Creating Referential Constraints on a Table	10-35
This Table References Another	10-37
This Table Referenced by Another	10-38
Recompiling Procedures and Triggers	10-40
Checking Table Consistency	10-40
Consistency Check Options	10-42
Overall Table Consistency Option	10-42
Table Allocations Option	10-43
Reindex Option	10-44
Fix Text Option	10-44

Updating Statistics	10-45
Copying Table Data	10-45

11. Enterprise SQL Server Manager Profile Managers

What's in This Chapter	11-1
Profile Management Concepts	11-2
Enterprise SQL Server Manager Profile Managers	11-4
SQL Server Profile Managers	11-5
Database Profile Managers	11-6
Creating Profile Managers	11-7
Creating a SQL Server Profile Manager	11-8
Creating a Database Profile Manager	11-10
Moving Profile Managers	11-11
Deleting Profile Managers	11-12
Profile Manager Window	11-13
Opening a Profile Manager	11-13
Left Pane of a Profile Manager Window	11-15
The Right Pane of the Profile Manager Window	11-15
Subscribers Icon	11-16
Details View	11-16
Profile Manager Menus	11-17
Standard Profile Manager Menus	11-17
Context-Specific Menus	11-19
Shortcut Menus	11-19
Opening a SQL Server Window from a Profile Manager	11-21

12. Managing ESSM Profiles

What's in this Chapter	12-1
Managing Profiles	12-1
Creating a Profile	12-2
Deleting a Profile	12-4
Populating a Profile	12-5
Populating a Profile with All Objects in One Step	12-5
Populating a Profile with the Populate Profile Dialog Box	12-6
Unpopulating a Profile	12-7
Synchronizing a Profile	12-8
Validating a Profile	12-8
Creating and Deleting Objects in Profile Managers	12-10
Creating Objects: Getting Default Values	12-10

Managing Subscribers	12-11
Profile Managers as Subscribers	12-12
Adding Subscribers to a SQL Server Profile Manager	12-14
Release Compatibility	12-15
Adding Subscribers to a Database Profile Manager	12-16
Removing Subscribers from a SQL Server Profile Manager	12-18
Removing Subscribers from a Database Profile Manager	12-19
Distributing Profiles	12-20
Distributing All Profiles to All Subscribers	12-21
Distributing Profiles Selectively	12-22
Previewing a Distribution	12-24
Distributing Table Data	12-25
Enabling and Disabling Distribution of Table Data	12-26
How Distribution Failures are Handled	12-27

13. Managing Policy

What's in this Chapter	13-1
What is Policy?	13-1
Distributing Policy	13-2
Defining Default Policies	13-3
Attribute Values	13-5
Generating Default Values	13-6
Default Policy Behavior	13-6
Enabling and Disabling Default Policy	13-7
Defining Validation Policies	13-8
Enabling and Disabling Validation Policy	13-10
Validating a Profile	13-11
Using Policy Scripts	13-12
Return Values in Policy Scripts	13-12
Creating a Policy Script	13-12
Using Policy Script Arguments	13-14
Sample Validation Policy Script	13-14

14. Using Event Monitoring Services

What's in this Chapter	14-1
Events You Can Monitor	14-1
Before You Use Event Monitoring Services	14-4
Specifying Roles	14-4
Using the Notices Group Response Action	14-4
Using Environment Variables in Response Scripts	14-4

Distributing Sentry Profiles	14-5
Monitoring SQL Servers on Non-Tivoli Platforms	14-5
How to Monitor for and Manage SQL Server Events	14-6
SQL Server Monitoring Sources	14-7
Generic Event Monitors	14-7
Number of Connection Attempts	14-8
Number of Connections Used	14-8
Number of Device Reads	14-8
Number of Device Writes	14-8
Number of Faulty Databases	14-9
Number of Locks Used	14-9
Number of Network Packets Received	14-9
Number of Network Packets Sent	14-9
Percent of Connections Used	14-10
Percent of CPU Busy	14-10
Percent of Locks Used	14-10
Percent of Segment Space Available	14-11
SQL Server Status	14-12
Sybase Login Status	14-13
User-Specified Event	14-14
Process-Specific Monitors	14-15
Number of Bad Status Processes	14-15
Number of Blocked Processes	14-15
Number of Infected Processes	14-15
Number of Stopped Processes	14-15
Error Log File Monitors	14-16
Search Using SQL Server Error Code(s)	14-17
Search Using Minimum Severity Level	14-17
Search Using Regular Expression	14-17
Search for Error Types	14-18
Setting the Error Log File Attribute	14-19
Using Sentry Profiles to Monitor for SQL Server Events	14-20
Adding Monitoring Sources to a Sentry Profile	14-20
Adding Monitors to a Profile	14-23
Subscribing to a Sentry Profile	14-26
Unsubscribing from a Sentry Profile	14-26
Using the waddmon Command	14-26
Troubleshooting Event Monitoring Services	14-30
Using the Event Monitoring Services Log File	14-30
Using the Notices Bulletin Board	14-31

A. SQL Server Configuration Parameters

What's in This Appendix	A-1
SQL Server 10.x Parameter List	A-1
SQL Server Release 11.0 Parameter List	A-4
Parameters by Functional Group	A-11
Backup/Recovery Parameters	A-11
Cache Manager Parameters	A-12
Disk I/O Parameters	A-12
General Information Parameter	A-12
Languages Parameters	A-12
Lock Manager Parameters	A-12
Memory Use Parameter	A-13
Network Communications Parameters	A-13
Operating System Resources Parameters	A-13
Physical Memory Parameters	A-13
Processors Parameters	A-13
SQL Server Administration Parameters	A-14
User Environment Parameters	A-15

B. Guide to Enterprise SQL Server Manager Icons

What's in this Appendix	B-1
List of Icons	B-1
Valid Drag-and-Drop Operations	B-4
SQL Server Release 11.0 Drag-and-Drop Operations	B-7

C. SQL Server Profile Operations

What's in this Appendix	C-1
Profile Distribution Order	C-1
Profiles in a SQL Server Profile Manager	C-2
SQLServerProfile	C-2
SQLDumpDeviceProfile	C-2
SQLDbDeviceProfile	C-2
SQLDatabaseProfile	C-3
SQLLoginProfile	C-3
SQLRemoteServerProfile	C-3
SQLCacheProfile	C-4
Profiles in a Database Profile Manager	C-4
SQLDatabaseProfile	C-4
SQLSegmentProfile	C-5

SQLGroupProfile	C-5
SQLUserProfile	C-5
SQLRuleProfile	C-6
SQLDefaultProfile	C-6
SQLDataTypeProfile	C-7
SQLTableProfile	C-7
SQLIndexProfile	C-8
SQLViewProfile	C-8
SQLProcedureProfile	C-8
SQLTriggerProfile	C-9

D. Enterprise SQL Server Manager Role Requirements

Role Requirements for Creating Objects	D-2
Role Requirements for Changing SQL Server and Database Object Properties	D-3
Role Requirements for Deleting SQL Server and Database Objects	D-5
Role Requirements for Displaying SQL Server and Database Object Properties ...	D-6
Role Requirements for Managing Enterprise SQL Server Manager Resources	D-7
Role Requirements for Miscellaneous Database Management Tasks	D-8
Role Requirements for Miscellaneous SQL Server Management Tasks	D-9

Glossary

Index

List of Figures

Figure 1-1:	Tivoli Management Environment	1-4
Figure 1-2:	The Enterprise SQL Server Manager graphical user interface (GUI).....	1-5
Figure 2-1:	Administrator's popup menu.....	2-4
Figure 2-2:	Set TMR Roles dialog box.....	2-4
Figure 2-3:	Set Login Names dialog box	2-7
Figure 2-4:	SQL Server Logins for Administrators: Enterprise Level dialog box	2-8
Figure 2-5:	SQL Server Logins for Administrators: Policy Level dialog box.....	2-9
Figure 2-6:	SQL Server Logins for Administrators: SQL Server Level dialog box	2-10
Figure 2-7:	Set Managed Resources dialog box.....	2-12
Figure 2-8:	Managed SQL Server resides on management host.....	2-14
Figure 2-9:	Managed SQL Server and remote management host.....	2-15
Figure 2-10:	Manage SQL Server dialog box	2-16
Figure 2-11:	Policy region with managed ESSM resources	2-17
Figure 3-1:	Enterprise SQL Server Manager Voyager window.....	3-2
Figure 3-2:	Object display formats	3-6
Figure 3-3:	Details view of databases, sorted by name and by size	3-7
Figure 3-4:	Filter by Name dialog box.....	3-8
Figure 3-5:	Drag-and-drop symbol	3-10
Figure 3-6:	Standard menu bar	3-11
Figure 3-7:	Pull-down menu for a table	3-13
Figure 3-8:	Shortcut menu for a table	3-15
Figure 3-9:	Toolbar and status bar.....	3-16
Figure 3-10:	Enterprise SQL Server Manager toolbar	3-17
Figure 3-11:	ToolTip for the Refresh tool bar button	3-18
Figure 3-12:	Status bar.....	3-19
Figure 3-13:	Properties tab of a multiple-tab Table dialog box	3-21
Figure 3-14:	Dependencies tab of the multiple-tab Table dialog box.....	3-22
Figure 3-15:	Navigating to a related object.....	3-23
Figure 3-16:	Example of a command dialog box.....	3-24
Figure 3-17:	Example of a confirmation dialog box.....	3-25
Figure 3-18:	Example of a simple message dialog box.....	3-25
Figure 3-19:	Output dialog box.....	3-26
Figure 3-20:	Methods of refreshing a window or dialog box.....	3-27
Figure 3-21:	Methods of moving between open windows and dialog boxes	3-28
Figure 4-1:	Start SQL Server dialog box	4-2
Figure 4-2:	SQL Server Status dialog box.....	4-3
Figure 4-3:	Stop SQL Server dialog box	4-4
Figure 4-4:	SQL Server window	4-10

Figure 4-5:	Server Properties dialog box	4-12
Figure 4-6:	Configuration Parameters dialog box for SQL Server release 10.x	4-15
Figure 4-7:	Configuration Parameters dialog box for SQL Server release 11.0.....	4-18
Figure 4-8:	Server Processes dialog box	4-20
Figure 4-9:	Create Remote Server dialog box	4-24
Figure 4-10:	Remote Server Login Mappings tab	4-28
Figure 4-11:	Remote Server Properties dialog box.....	4-30
Figure 4-12:	Server Generate DDL dialog box	4-33
Figure 4-13:	Server Run DDL dialog box	4-34
Figure 4-14:	Server Consistency Check dialog box.....	4-38
Figure 4-15:	dbcc engine output	4-39
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	5-8
Figure 5-4:	Cache Bindings tab	5-12
Figure 6-1:	Logical database devices on a physical disk device	6-3
Figure 6-2:	Create Database Device dialog box.....	6-4
Figure 6-3:	Properties tab of Database Device dialog box	6-7
Figure 6-4:	Databases tab of Database Device dialog box	6-8
Figure 6-5:	Segments tab of Database Device dialog box	6-10
Figure 6-6:	Database device on disk1 mirrored on disk2.....	6-12
Figure 6-7:	Database Device dialog box with disk mirroring enabled	6-14
Figure 6-8:	Database device with mirroring disabled	6-17
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 Options dialog box	7-18
Figure 7-8:	Create Segment dialog box.....	7-24
Figure 7-9:	Properties tab of Segment dialog box	7-28
Figure 7-10:	Contains tab of the Segment dialog box.....	7-30
Figure 7-11:	Thresholds tab of Segment dialog	7-33
Figure 7-12:	Create Dump Device dialog box	7-39
Figure 7-13:	Properties tab of the Dump Device dialog box	7-41
Figure 7-14:	Database Backup dialog box	7-44
Figure 7-15:	Volume Change dialog box	7-47
Figure 7-16:	Database Restore dialog box	7-50
Figure 7-17:	Volume Change dialog box	7-53
Figure 7-18:	Database Consistency Check dialog box.....	7-56

Figure 7-19:	dbcc checkdb output	7-57
Figure 7-20:	dbcc checkalloc output.....	7-58
Figure 7-21:	dbcc checkcatalog output	7-59
Figure 7-22:	Confirmation message: setting a database online.....	7-60
Figure 8-1:	Obtaining access to a database object	8-8
Figure 8-2:	Create Login dialog box.....	8-9
Figure 8-3:	Login Properties tab	8-12
Figure 8-4:	Change Password dialog box.....	8-14
Figure 8-5:	Login Databases Owned tab	8-15
Figure 8-6:	Login Alias/Users tab.....	8-16
Figure 8-7:	Create User dialog box.....	8-19
Figure 8-8:	User Properties tab	8-23
Figure 8-9:	User Objects Owned tab	8-25
Figure 8-10:	User Object Permissions tab.....	8-26
Figure 8-11:	User Command Permissions tab	8-30
Figure 8-12:	Create Group dialog box	8-32
Figure 8-13:	Group Properties tab	8-34
Figure 8-14:	Group Object Permissions tab	8-38
Figure 8-15:	Group Command Permissions tab.....	8-41
Figure 9-1:	Create dialog box for an index.....	9-3
Figure 9-2:	Confirmation dialog box.....	9-4
Figure 9-3:	Properties tab for an index	9-5
Figure 9-4:	Voyager window showing object hierarchy	9-6
Figure 9-5:	Dependencies tab for a view	9-7
Figure 9-6:	Create Default dialog box.....	9-11
Figure 9-7:	Creating the default <i>statedflt</i>	9-12
Figure 9-8:	<i>statedflt</i> default as created by Enterprise SQL Server Manager	9-13
Figure 9-9:	Default Dependencies tab	9-14
Figure 9-10:	Default Bindings tab.....	9-16
Figure 9-11:	Create Index dialog box.....	9-21
Figure 9-12:	Index Properties tab	9-25
Figure 9-13:	Index Segments tab.....	9-26
Figure 9-14:	Index Consistency Check dialog box.....	9-28
Figure 9-15:	Output from Full option of index consistency check	9-29
Figure 9-16:	Create Procedure dialog box.....	9-32
Figure 9-17:	Procedure Properties tab	9-34
Figure 9-18:	Procedure Dependencies tab.....	9-35
Figure 9-19:	Procedure Permissions tab	9-37
Figure 9-20:	Create Rule dialog box.....	9-41
Figure 9-21:	Rule Properties tab	9-42
Figure 9-22:	Rule Dependencies tab.....	9-43

Figure 9-23:	Rule Bindings tab.....	9-45
Figure 9-24:	Create Trigger dialog box	9-49
Figure 9-25:	Creating the newhero trigger.....	9-50
Figure 9-26:	newhero trigger as created by Enterprise SQL Server Manager	9-51
Figure 9-27:	Trigger Properties tab.....	9-52
Figure 9-28:	Trigger Dependencies tab	9-53
Figure 9-29:	Create User Datatype dialog box	9-55
Figure 9-30:	User Datatype Properties tab	9-57
Figure 9-31:	User Datatype Dependencies tab	9-58
Figure 9-32:	Create View dialog box.....	9-61
Figure 9-33:	Creating the titlepub view.....	9-63
Figure 9-34:	titlepub view as created by Enterprise SQL Server Manager	9-64
Figure 9-35:	View Properties tab	9-65
Figure 9-36:	View Dependencies tab.....	9-66
Figure 9-37:	View Permissions tab	9-69
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	10-14
Figure 10-4:	Table Column Advanced dialog box	10-17
Figure 10-5:	Table Dependencies tab	10-21
Figure 10-6:	Table Permissions tab.....	10-22
Figure 10-7:	Table Segments tab	10-26
Figure 10-8:	Table Indexes/Triggers tab.....	10-28
Figure 10-9:	Table Check Constraints tab.....	10-30
Figure 10-10:	Table Partition tab.....	10-33
Figure 10-11:	Table Partition tab.....	10-34
Figure 10-12:	Table Referential Constraints tab.....	10-36
Figure 10-13:	Referential Constraints tab for referenced table.....	10-38
Figure 10-14:	Table Consistency dialog box.....	10-41
Figure 10-15:	Table consistency check output—overall consistency option	10-42
Figure 10-16:	Table consistency check output—allocations option.....	10-43
Figure 10-17:	Table consistency check output—reindex option	10-44
Figure 11-1:	Copies of a profile are distributed to subscribers	11-3
Figure 11-2:	Profile Manager icons.....	11-4
Figure 11-3:	Create SQL Server Profile Manager dialog box.....	11-8
Figure 11-4:	Create Database Profile Manager dialog box	11-10
Figure 11-5:	Delete profile manager confirmation dialog box	11-12
Figure 11-6:	Shortcuts for opening a profile manager.....	11-13
Figure 11-7:	SQL Server Profile Manager window	11-14
Figure 11-8:	Database Profile Manager window.....	11-14
Figure 11-9:	SQL Server Profile Manager with Subscribers container selected	11-16

Figure 12-1:	Create Profiles dialog box.....	12-2
Figure 12-2:	Shortcuts for creating a profile.....	12-3
Figure 12-3:	Delete profile confirmation dialog box.....	12-4
Figure 12-4:	Shortcuts for deleting a profile	12-4
Figure 12-5:	Populate Profile dialog box	12-6
Figure 12-6:	Validation Results window	12-9
Figure 12-7:	A profile manager as a subscriber	12-12
Figure 12-8:	Distribution to a subscribing profile manager	12-13
Figure 12-9:	Subscribers dialog box for a SQL Server profile manager	12-14
Figure 12-10:	Subscribers dialog box for a database profile manager	12-16
Figure 12-11:	Distribute Profiles dialog box for policy region	12-21
Figure 12-12:	Distribute Profile dialog box	12-22
Figure 12-13:	Preview Distribution dialog box	12-24
Figure 13-1:	Default Policy dialog box	13-3
Figure 13-2:	Validation Policy dialog box	13-8
Figure 13-3:	Validation Results window	13-11
Figure 13-4:	Policy Script dialog box	13-13
Figure 14-1:	Illustration of scan back time	14-16
Figure 14-2:	Create Profile Manager dialog box.....	14-20
Figure 14-3:	Policy region window	14-21
Figure 14-4:	Create Profile dialog box	14-21
Figure 14-5:	Profiles in a Profile Manager window	14-22
Figure 14-6:	Sentry Profile Properties dialog box	14-23
Figure 14-7:	Add Monitor to Tivoli/Sentry Profile dialog box.....	14-24
Figure 14-8:	Edit Sentry Monitor dialog box	14-25

List of Tables

Table 1:	Style conventions	xxxiii
Table 2:	Shortcut symbols	xxxiv
Table 3:	Displaying online help information.....	xxxvi
Table 2-1:	Enterprise SQL Server Manager administrator roles	2-2
Table 3-1:	Activities available from standard menus	3-12
Table 3-2:	Shortcut menu commands	3-14
Table 3-3:	Toolbar controls.....	3-17
Table 4-1:	Files created during generation of server-level DDL	4-36
Table 4-2:	Files created during generation of database-level DDL.....	4-37
Table 6-1:	Effects of disabling mirroring	6-16
Table 7-1:	Files created during generation of database-level DDL.....	7-16
Table 8-1:	Object permission dialog boxes.....	8-7
Table 8-2:	Privilege and permission summary	8-43
Table 8-3:	Access granted through tabs and dialog boxes.....	8-48
Table 9-1:	Role requirements for creating database objects.....	9-2
Table 9-2:	Role requirements for deleting database objects.....	9-8
Table 10-1:	Types of table constraints	10-4
Table 11-1:	SQL Server Profile Manager profiles	11-5
Table 11-2:	Database Profile manager profile types.....	11-6
Table 11-3:	Manage menu in a profile manager window	11-18
Table 11-4:	Shortcut menu commands for profile manager windows.....	11-19
Table 12-1:	Subscriber types allowed in profile managers	12-11
Table 12-2:	ESSM and SQL Server roles required for distributing ESSM profiles.....	12-20
Table 13-1:	Application of policy to object changes and creation.....	13-2
Table 13-2:	Object attributes subject to policy	13-4
Table 13-3:	Attributes not supported by Get Defaults	13-6
Table 14-1:	Error codes monitored by Event Monitoring Services	14-18
Table 14-2:	Monitoring Source Names.....	14-27
Table 2-1:	Enterprise SQL Server Manager icons.....	B-1
Table 2-2:	Icon drag-and-drop operations.....	B-4
Table 2-3:	SQL Server release 11.0 icon drag-and-drop operations.....	B-7
Table D-1:	Roles required to create objects	D-2
Table D-2:	Roles required to change object properties	D-3
Table D-3:	Roles required to delete objects	D-5
Table D-4:	Roles required to display object properties	D-6
Table D-5:	Roles required to manage ESSM resources.....	D-7
Table D-6:	Roles required for miscellaneous database management tasks.....	D-8
Table D-7:	Roles required for miscellaneous database management tasks.....	D-9

About This Book

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

Audience

This manual is for experienced Sybase SQL Server administrators planning to use Enterprise SQL Server Manager (ESSM). To understand the information in this manual also requires familiarity with the Tivoli Management Environment (TME) concepts and procedures. See the *Tivoli Management Platform Guide* for more information about the TME.

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

How to Use This Book

To get started with Enterprise SQL Server Manager, read:

- Chapter 1, “Introduction,” which gives an overview of Enterprise SQL Server Manager.
- Chapter 2, “Setting Up the Tivoli Management Environment for ESSM,” which explains how to create managed SQL Server resources in a Tivoli policy region.
- Chapter 3, “Enterprise SQL Server Manager Windows and Dialog Boxes,” which introduces you to the desktop window and to the various types of dialog boxes used in Enterprise SQL Server Manager.

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

To learn more about how to use Enterprise SQL Server Manager, to manage individual SQL Server installations, read Chapters 4-10, in any order.

- Chapter 4, “Managing the SQL Server Operating Environment,” describes how to start, 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.

To learn more about using Enterprise SQL Server Manager, to manage your enterprise, read the following chapters.

- Chapter 11, “Enterprise SQL Server Manager Profile Managers,” describes profile managers and how to create and manage them.
- Chapter 12, “Managing ESSM Profiles,” explains profile concepts and how to create and distribute profiles.
- Chapter 13, “Managing Policy,” describes how to specify and use default and validation policy.
- Chapter 14, “Using Event Monitoring Services,” describes how to use the Enterprise SQL Server Manager Event Monitoring Services to monitor SQL Server events.

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 Enterprise SQL Server Manager Icons,” is a pictorial guide to the icons used in Enterprise SQL Server Manager to represent objects and utilities.
- Appendix C, “SQL Server Profile Operations” lists the actions taken by each profile and dependencies on other profiles.

- Appendix D, “Enterprise SQL Server Manager Role Requirements,” lists the roles required for managing SQL Server, databases, and managed resources.
- The Glossary contains definitions of system administration and Enterprise SQL Server Manager terms.

Related Documents

The Enterprise SQL Server Manager documentation set consists of online help and the following Sybase and Tivoli books:

- *Tivoli Management Platform Release Notes* describes important release-specific information for the release of TME included with your Enterprise SQL Server Manager software.
- *Tivoli Management Platform Guide* describes important concepts and features of the Tivoli Management Environment. Because Enterprise SQL Server Manager runs as an application within TME, it is important that you have a general understanding of TME before you install or use Enterprise SQL Server Manager.
- *Tivoli Management Platform Reference Manual* describes Tivoli Management Environment commands.
- *Enterprise SQL Server Manager Release Bulletin* describes release-specific information, including special installation instructions and known software and documentation issues.
- *Enterprise SQL Server Manager Installation and Planning Guide* describes how to plan an Enterprise SQL Server Manager installation and how to install Enterprise SQL Server Manager.
- *Enterprise SQL Server Manager User's Guide* (this book) describes Enterprise SQL Server Manager features and how to use them.
- Enterprise SQL Server Manager Online Help is a complete source of information about the Enterprise SQL Server Manager graphical user interface (GUI) and provides detailed information about the Enterprise SQL Server Manager desktop features and the functions they perform.
- *Enterprise SQL Server Manager Reference Manual* describes the Enterprise SQL Server Manager command line interface (CLI) and the Enterprise SQL Server Manager commands.

- *Tivoli/Sentry User's Guide* describes the Tivoli/Sentry management application tool. Using Tivoli/Sentry, you can configure Enterprise SQL Server Manager Event Monitoring Services to monitor SQL Server events.

Other Documents

- *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 Enterprise SQL Server Manager. It includes the Transact-SQL commands and system procedures used to perform the functions for which Enterprise 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>exec.ddl</i>	Filenames, variable names, and book titles
<code>check (pub_id in "1389")</code>	Values that you type exactly as shown





Shortcuts

You can perform most Enterprise SQL Server Manager activities by executing a drop-down menu command. For most activities, there are also shortcuts that help you work more efficiently. For example, Enterprise 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 Enterprise SQL Server Manager operations, this manual first presents the procedure for executing the appropriate menu command. If an alternative method 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
	Drag one of the icons shown and drop it on the other
	Select the toolbar button shown
	Execute a shortcut menu command. (Click the right mouse button.)

Roles

Many activities in Enterprise SQL Server Manager are restricted to users with specific TME and SQL Server roles. This manual specifies required roles at the beginning of each procedure, for example:

	TME	ESSM	SQL Server
Required roles	none	security	sso_role

Mouse Buttons

To use Enterprise 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. On a three-button mouse, the button used to execute commands corresponds to the left button on a two-button mouse.

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 that include controls for both release 10.x and 11.0 features, controls that support release 11.0 are not visible.
- When you are managing 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

You can get help for using Enterprise 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 Enterprise 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 Voyager window.	Click the Help button on the toolbar: 
Table of Contents for Enterprise 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.
Information on how to use the 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 Enterprise SQL Server Manager. Displays product version number.	Choose About ESSM 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

What's in This Chapter

This chapter provides an introduction to the major features of Enterprise SQL Server Manager.

Enterprise SQL Server Manager (ESSM) is a tool for administering SQL Server installations, both individually and across the enterprise. Enterprise SQL Server Manager is specifically designed to address the needs of an organization that has implemented an enterprise-based client/server architecture. Enterprise SQL Server Manager offers a highly scalable, interoperable, and extensible solution and adheres to a common, standards-based, object-oriented architecture. Enterprise SQL Server Manager runs as an application in the Tivoli Management Environment (TME) enabling administrators to manage multiple SQL Servers distributed across multiple hosts from a single desktop.

Enterprise SQL Server Manager provides you with two major capabilities:

- The capability to manage individual SQL Server installations
- The capability to manage many SQL Server installations at one time

You can conduct management tasks from the operating system command line or through the Enterprise SQL Server Manager graphical user interface (GUI).

Managing Individual SQL Server Installations

Enterprise SQL Server Manager is a 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

Enterprise SQL Server Manager lets you do the following typical system or database administrator's tasks:

- **Managing SQL Server:**
 - Start and stop SQL Server
 - Display and terminate SQL Server processes
 - Generate and run server Data Definition Language (DDL) scripts
 - Enable and control access to or from remote servers
 - Troubleshoot SQL Server problems
- **Managing data caches:**
 - Create user-defined data caches
 - Define buffer pools
 - Bind databases and objects to a cache
- **Managing SQL Server physical resources:**
 - Create and delete database devices and dump devices
 - Create mirror devices
- **Managing databases:**
 - Create and delete databases and database objects
 - Set database options
 - Set database options
 - Manage database storage allocation across devices and segments
 - Back up and restore databases
 - Issue database checkpoints
 - Generate and execute database DDL scripts
 - Run the `dbcc` database check and repair utility
- **Managing access:**
 - Create and delete SQL Server logins
 - Create and delete database users and user groups
 - Administer Sybase roles
 - Manage object and command permissions

Making It Easy

Enterprise SQL Server Manager makes it easy to perform complex administration tasks. Special features that support ease-of-use are:

- **Visual representation of objects.** The Enterprise SQL Server Manager display, called the Voyager window, represents SQL Server and its objects. It expands and contracts to display information at the level of detail you need.
- **Navigation between related objects.** If a dialog box for a database object references another object, you can navigate directly from the open dialog box to a dialog box for the referenced object. For example, to open the dialog box for a trigger defined on a table, double-click the trigger name in the table dialog box.
- **Drag-and-drop operations.** You can execute many tasks using drag-and-drop. For example, to add a user to a group, drag the icon representing the user over the icon representing the group, and click OK in the confirmation dialog box.
- **The ability to filter the display of objects.** Enterprise SQL Server Manager allows you to limit the display to objects that match a specified string of characters.
- **Toolbar buttons for frequently-performed tasks.** Click a button in the toolbar to perform such tasks as creating or deleting an object or refreshing the current display of SQL Server information.

Managing the Enterprise

In addition to letting you manage individual SQL Server installations, Enterprise SQL Server Manager allows you to manage multiple SQL Server installations. This is because Enterprise SQL Server Manager runs as an application in the **Tivoli Management Environment (TME)**, a product designed for distributed system management.

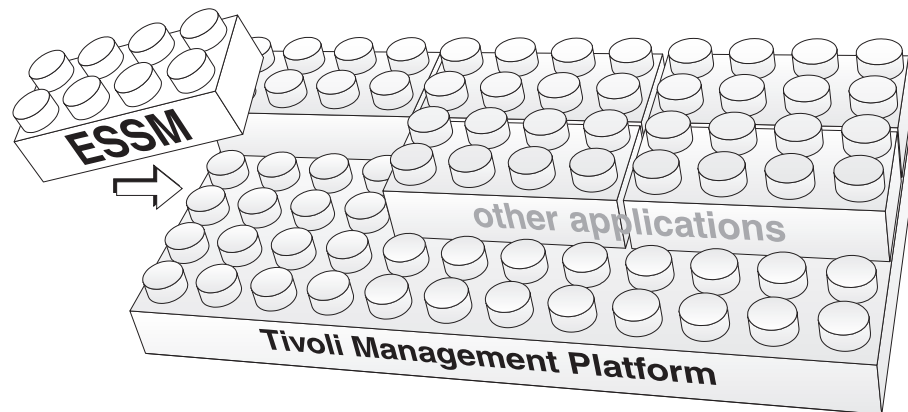


Figure 1-1: Tivoli Management Environment

Running in the TME environment, Enterprise SQL Server Manager shares the same framework services used by other management applications. These shared services include:

- Control over administrator authorization through **role** assignment services.
- Implementation of rules through policies and policy regions.
- Distribution of SQL Server and database objects through TME profile management services.
- A common event monitoring facility through Tivoli Monitoring Technology (TMT) and Tivoli/Sentry profiles.
- A common notification facility through the TME Bulletin Board services.
- Job automation and scheduling through TME Task Library and scheduler services.

The Enterprise SQL Server Manager User Interface

Enterprise SQL Server Manager provides a graphical user interface that you access from the TME desktop and a command line interface that you access from the operating system command line.

Graphical User Interface

Enterprise SQL Server Manager uses both the Tivoli desktop and the following special Enterprise SQL Server Manager windows:

- The SQL Server window
- The SQL Server Profile Manager window
- The Database Profile Manager window

These three windows, also referred to generically as Voyager windows, use a hierarchical list to display all objects managed by Enterprise SQL Server Manager and allow flexibility in managing SQL Server.

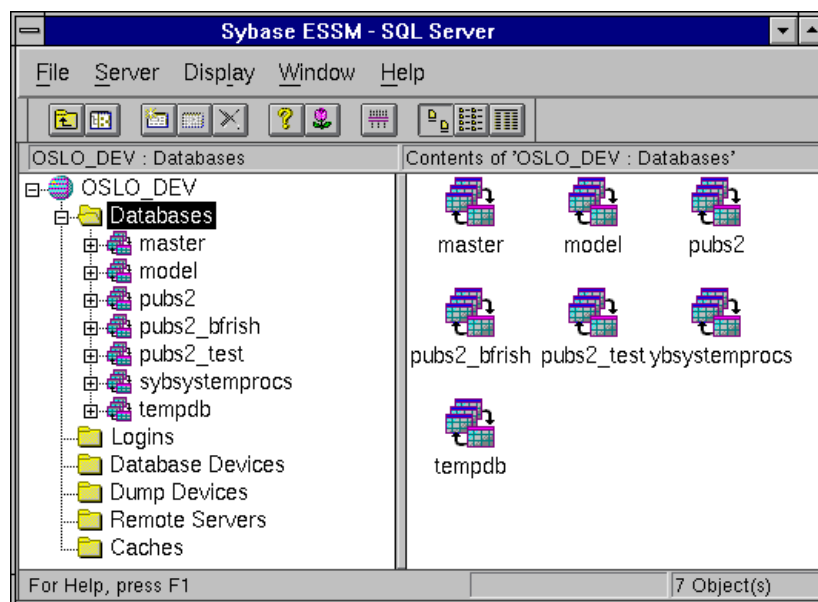


Figure 1-2: The Enterprise SQL Server Manager graphical user interface (GUI)

For more information about using the TME desktop, see *Tivoli Management Platform User's Guide*. For information about the Voyager window, see Chapter 3, "Enterprise SQL Server Manager Windows and Dialog Boxes."

Command Line Interface

Enterprise SQL Server Manager has a comprehensive command set that you can use to perform administration activities from the operating system command line. For example:

```
pine% sgetserver -name PARIS_DEV -list
*** SQL Server: PARIS_DEV ***
Version: SQL Server 10.0.2.1
Platform: HP9000
Operating System: 800
Build Options: HP-UX 9.0 EBF 4010
Build Date: OPT/Wed Nov 16 09:43:57 PST 1994
pine%
```

You can also create scripts, combining Enterprise SQL Server Manager commands and Tivoli commands with virtually any scripting language, to facilitate batch processing of repetitive or large-scale administrative chores. You can automate this work by saving the scripts as tasks in the TME Task Library and scheduling them in the desktop scheduler to execute automatically.

For more information about Enterprise SQL Server Manager commands, see *Enterprise Enterprise SQL Server Manager Reference Manual*.

Enterprise SQL Server Manager Policy Region Resources

To allow you to manage and configure SQL Server and database objects in the TME, Enterprise SQL Server Manager adds resource types to the TMR database. These resource types appear in the list of available TME resources in a **policy region**. Enterprise SQL Server Manager uses the following types of resources:

- Managed SQL Server
- SQL Server Profile Manager
- Database Profile Manager
- The profile types used by Enterprise SQL Server Manager

Profile Management

Enterprise SQL Server Manager **profile** management is a powerful tool that allows you to distribute SQL Server and database objects to many SQL Servers across the enterprise in a single step.

The ability to create and distribute profiles is the key to managing your enterprise. Without profile **distribution**, you can manage only one SQL Server installation at a time. With profile distribution, you can manage hundreds of SQL Server installations at a time.

You use profile managers to define profiles and set up the relationships with subscribers. A profile is, essentially, a description or model of an object or group of objects of the same type in an existing SQL Server or database that you want to distribute. **Subscribers** are the SQL Servers, databases, and other profile managers to which you want to distribute the objects.

When you distribute a profile to a subscriber, Enterprise SQL Server Manager copies the data in the profile to the recipient SQL Server or database. This allows you to create an object on one SQL Server and then distribute the object definition to as many other SQL Server installations or databases as you are managing with Enterprise SQL Server Manager.

You can create SQL Server Profile Managers and Database Profile Managers to define profiles, create distribution hierarchies, and to control distribution of the profiles to their associated subscribers.

For more information about profile management, see Chapter 11, “Enterprise SQL Server Manager Profile Managers” and Chapter 12, “Managing ESSM Profiles.”

Default Policy and Validation Policy

Enterprise SQL Server Manager lets you configure policy for the attributes of SQL Server profiles.

- **Default policy** methods generate default values for attributes of an object in a profile when you create the object
- **Validation policy** methods check that an object’s attribute values comply with those established for that profile type when you change the object or add new objects to the profile

For more information about policy, see Chapter 13, “Managing Policy.”

Event Monitoring Services

Event Monitoring (EMON) Services allows you to monitor SQL Server for specific events and to manage responses to the events. Event Monitoring Services are based on Tivoli Monitoring Technology (TMT). TMT is the same event management technology framework that Tivoli/Sentry uses.

An **event** occurs when a SQL Server characteristic surpasses a user-defined threshold or state. For example, you can use Event Monitoring Services to notify you when the number of network packets sent exceeds a specific number or if an error log file records an access error. Event Monitoring Services allows you to monitor the following SQL Server characteristics:

- Generic events
- Process specifics
- Error log file events

For more information about Event Monitoring Services, see Chapter 14, “Using Event Monitoring Services.”

Notices

Enterprise SQL Server Manager creates two notice groups in the Tivoli Management Environment:

- Sybase Administration
- Sybase Backup/Recovery

All Enterprise SQL Server Manager commands that create, delete, or modify SQL Server objects log a notice to the "Sybase Administration" notification group. Messages are in the following format:

```
Notice-id: ID_number
Date: Day/Date/Time/Year
Priority: Notice
Administrator: administrator_login
[action] [object]: [object path]
    Object Name : object_name
    Object Type : object_type
    Server : server_name
    [Database : db_name]
```

The priority of each of these messages is "Notice."

Enterprise SQL Server Manager transaction rollback does not log notices.

All Enterprise SQL Server Manager commands that affect the backup and recovery of databases log a notice to the Sybase Backup/Recovery notice group. This includes the start and end of a backup, the start and end of a restore, and notification of use of the `sp_volchanged` stored procedure.

Notifications during profile management are rolled into one notification message.

2

Setting Up the Tivoli Management Environment for ESSM

What's in This Chapter

This chapter describes the tasks you must complete after you install Enterprise SQL Server Manager to prepare the TME for managing SQL Server. Before you can use Enterprise SQL Server Manager, you must configure the TME to work with Enterprise SQL Server Manager resources. Chapter 7, “Getting Started with Enterprise SQL Server Manager” in *Enterprise SQL Server Manager Installation and Planning Guide* also describes the procedures for setting up your environment. Therefore, ask the person who installed Enterprise SQL Server Manager if any of these procedures were done.

Preparing the TME to Manage SQL Servers

Before you can use Enterprise SQL Server Manager, you must prepare the TME to manage SQL Server by:

- Assigning the appropriate Enterprise SQL Server Manager roles to administrators to allow them to perform management tasks on managed SQL Servers
- Assigning SQL Server logins to administrators to allow connections to SQL Servers
- Configuring one or more policy regions to allow SQL Servers as valid managed resources
- Adding managed SQL Server resources to the policy regions
- Adding profile managers to the policy regions.

All of these Enterprise SQL Server Manager activities require at least one existing TME administrator and one policy region. If you have not done so already, create these before proceeding with this chapter.

- For information about creating and configuring Tivoli administrators, see Chapter 3, “Tivoli Administrators” in the *Tivoli Management Platform User's Guide*.
- For information about creating and configuring policy regions, see Chapter 5, “Policy and Policy Regions,” in *Tivoli Management Platform User's Guide*.

Assigning Enterprise SQL Server Manager Roles to Administrators

To give an administrator access to SQL Server management functions in Enterprise SQL Server Manager, you must assign the necessary Enterprise SQL Server Manager roles to the administrator. Enterprise SQL Server Manager extends the standard set of TME administrator roles with the following application-specific roles:

Table 2-1: Enterprise SQL Server Manager administrator roles

Role name	Authorizes an administrator to perform...
dump	Database backup operations. Lets an administrator back up databases.
load	Database restore operations. Lets an administrator restore database backups.
server	SQL Server configuration operations. Lets an administrator modify a server configuration, manage SQL Servers, display and stop SQL Server processes, and start and stop managed SQL Servers.
security	Authentication, authorization, and auditing operations. Lets an administrator create, modify, and delete logins, users, groups, remote servers, and remote logins. Lets an administrator modify object permissions, command permission, server auditing, and object auditing.
space	SQL Server device configuration and allocation operations. Lets an administrator create, modify, and delete devices, dump devices, databases, device mirrors, segments, and thresholds.
schema	Schema management operations. Lets an administrator create, modify, and delete defaults, views, user datatypes, triggers, stored procedures, tables, rules, and named caches.
monitor	SQL Server Monitor launch.
cache	Cache operations.

Ideally, a hierarchy of administrator tasks is established for the enterprise. Within each TMR, a supervisory administrator assigns other administrators the roles and SQL Server logins they need to accomplish their assigned tasks. This process ensures that SQL Server configurations created and distributed using Enterprise SQL Server Manager cannot be compromised by administrator actions at the local SQL Server level.

An administrator must have the appropriate combination of Tivoli, Enterprise SQL Server Manager, and SQL Server roles to perform an Enterprise SQL Server Manager administration task. For example, to create a managed SQL Server resource in a policy region, an administrator must have the server role. The combination of roles varies according to the task. For a complete list of the roles required for each Enterprise SQL Server Manager administration task, see Appendix D, “Enterprise SQL Server Manager Role Requirements.”

Tivoli Management Region (TMR) and Resource Roles

Like all TME roles, you can assign Enterprise SQL Server Manager roles at both the TMR and the resource level. Roles assigned at the **TMR level** apply to all resources within a TMR. Roles assigned at the **resource level** apply only to a specific resource on the administrator’s desktop. The following steps describe how to assign TMR roles to another administrator. For more information about assigning administrator authorization roles, see Chapter 3, “Tivoli Administrators” in the *Tivoli Management Platform User’s Guide*.

► **Note**

To assign a TMR role to another administrator, you must have the Administrators collection icon on your TME desktop and you must have **super** or **senior** role over the Administrators resource.

1. Open the Administrators window to see the icons representing the defined TME administrators.
2. Locate the administrator you want to modify, and choose Edit TMR Roles from that administrator icon’s pop-up menu:

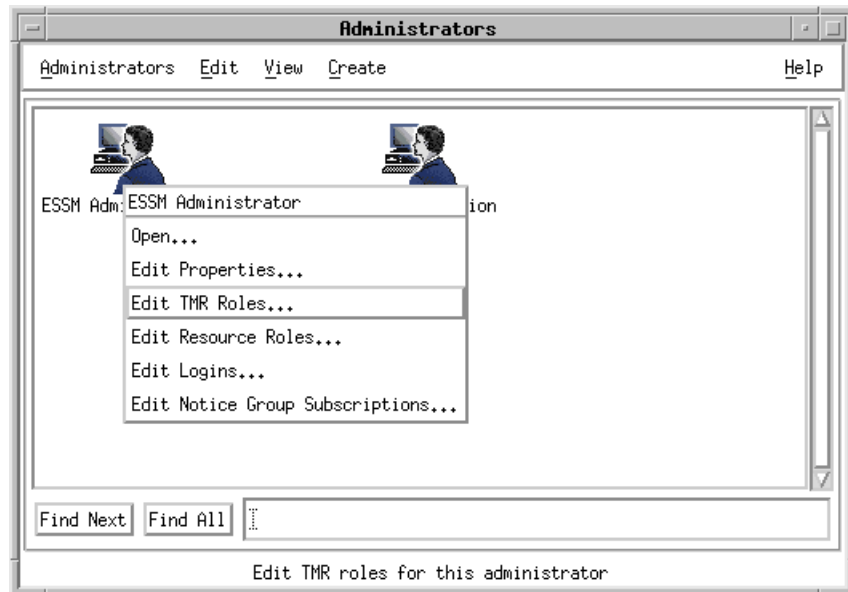


Figure 2-1: Administrator’s popup menu

The Set TMR Roles dialog box opens:

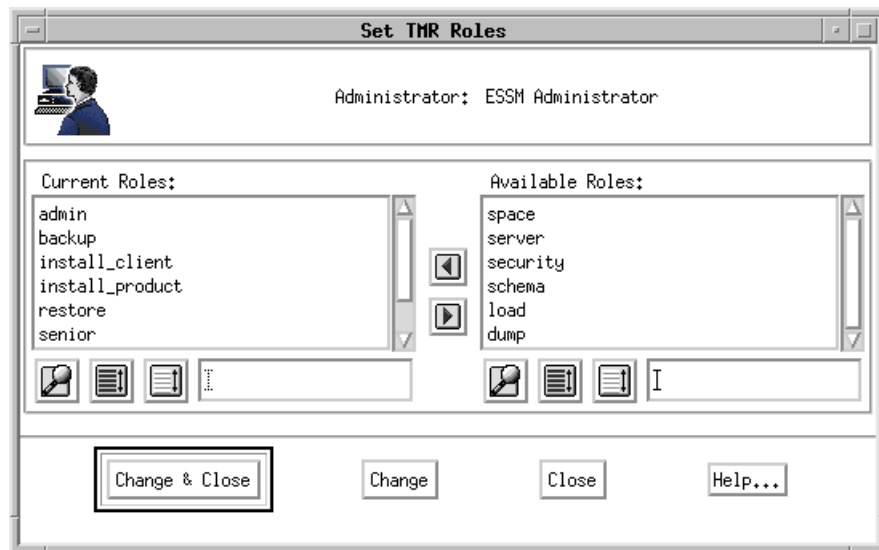


Figure 2-2: Set TMR Roles dialog box

3. In the Available Roles list box, select the roles you want to assign to the administrator. The Enterprise SQL Server Manager roles are:

- server
- space

- security
 - dump
 - load
 - schema
 - monitor
 - cache
4. Click the left-pointing arrow to move the selected roles to the Current Roles list box.
 5. Choose Change & Close to set the roles for the administrator and return to the Administrators window.
 6. Repeat these steps, beginning with step 2, for each Enterprise SQL Server Manager administrator you want to configure.

Assigning SQL Server Logins for Administrators

For each SQL Server in which an administrator is to perform management tasks, the administrator must have a valid SQL Server login assigned to connect to the server. The login must already exist in the SQL Server the administrator is to access.

SQL Server Logins for Multiple UNIX Logins

A single TME administrator can have more than one UNIX login assigned. Enterprise SQL Server Manager allows you to specify a different SQL Server login for each UNIX login associated with the administrator.

SQL Server Logins for Multiple Management Levels

You can configure the login assignments for an administrator's use in connecting to SQL Servers at three levels in the enterprise:

- For all managed SQL Servers within a TMR (enterprise-wide)
- For all managed SQL Servers within a policy region (supersedes any enterprise-wide SQL Server login assignment)
- For an individual managed SQL Server (supersedes any enterprise-wide or policy region SQL Server login assignment)

► *Note*

Event Monitoring Services allows you to monitor the relationship between the Enterprise SQL Server Manager Administrator login and the administrator's SQL Server login. If either login changes, the Enterprise SQL Server Manager Administrator cannot log on to a SQL Server. Therefore, you may want to use EMON Services to contact you if this relationship changes.

SQL Server Roles

You must be sure that the SQL Server login you assign to an administrator has the correct SQL Server roles (System Administrator, System Security Officer, Operator) for the Enterprise SQL Server Manager tasks that the administrator is to perform.

The following sections describe how to assign SQL Server logins to Enterprise SQL Server Manager administrators.

► **Note**

Assigning SQL Server logins to Enterprise SQL Server Manager administrators requires the **super** role.

Assigning ESSM Administrator SQL Server Logins for the Enterprise

To display or set an enterprise-wide SQL Server login for an administrator:

1. Open the Administrators window to see the icons representing the defined TME administrators.
2. Locate the administrator you want to modify, and choose Edit Logins from that administrator icon's pop-up menu.

The Set Login Names dialog box opens.



Figure 2-3: Set Login Names dialog box

3. In the Current Login Names list box, select the UNIX login name for the administrator you want to modify.
4. Choose Set SQL Server Logins. The SQL Server Logins for Administrator dialog box opens, displaying the Enterprise Level tab. The options on this tab let you set the enterprise-wide SQL Server login for the selected administrator UNIX login.

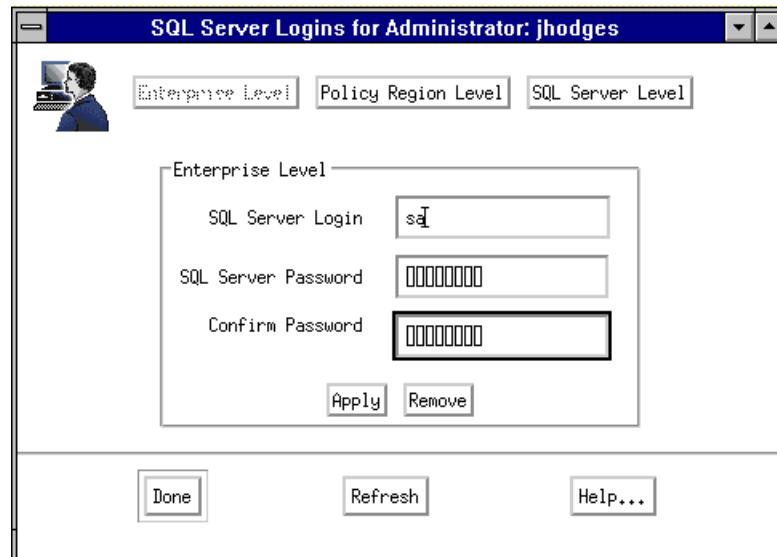


Figure 2-4: SQL Server Logins for Administrators: Enterprise Level dialog box

5. In the SQL Server Login box, enter the SQL Server login name that the administrator is to use when accessing any managed SQL Server in the TMR. In Figure 2-4, the Tivoli administrator “jhodes” is associated with the SQL Server login “sa”.
6. In the SQL Server Password box, enter the password to be used with the SQL Server login name.
7. In the Confirm Password box, retype the password to confirm that it is correct.
8. Choose Apply to record the login assignment in the TMR database.
9. Choose Done to close the dialog box.
10. Repeat these steps, beginning with step 3, for each UNIX login you want to associate with a SQL Server login.

To configure the UNIX logins for another administrator, start again at step 2 on page 2-7.

Assigning ESSM Administrator SQL Server Logins for Policy Regions

Open the SQL Server Logins for Administrators dialog box as described in the procedure on page 2-7.

1. Select the Policy Region Level tab. The Policy Region Level tab shows a list of all current policy region SQL Server login assignments for the administrator.

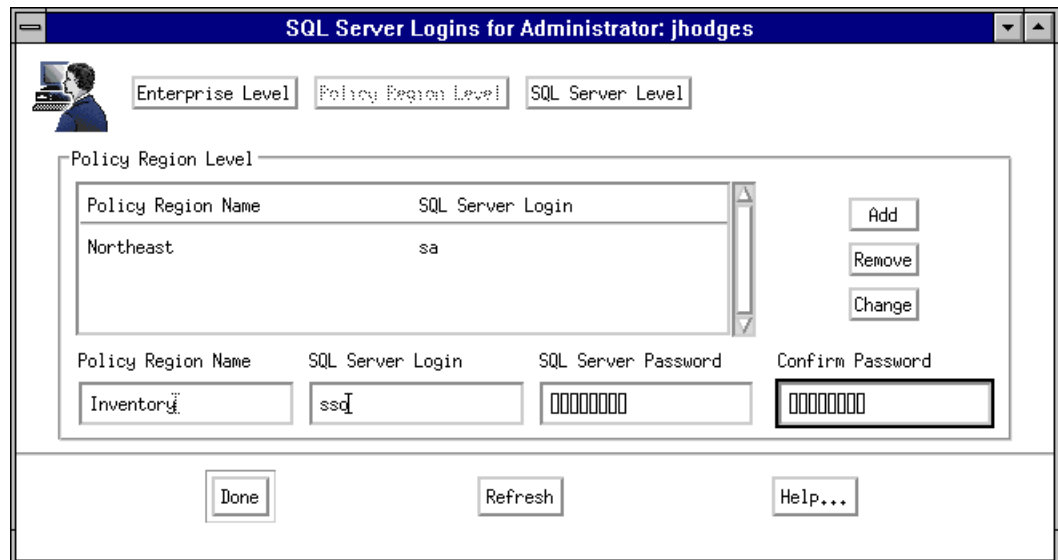


Figure 2-5: SQL Server Logins for Administrators: Policy Level dialog box

2. In the Policy Region Level group box, specify the list of policy regions and associated SQL Server logins you want to assign to the administrator, one pair at a time. You can add, remove, or change policy region SQL Server login assignments until you are satisfied with the list. In Figure 2-5, the Tivoli administrator “jhodes” is associated with the SQL Server login “sa” in the Northeast policy region. Data has been entered to associate “jhodes” with the SQL Server login “sso” in the Inventory policy region.
3. When you finish making SQL Server login assignments for the administrator, choose Done to close the dialog box.

Assigning ESSM Administrator SQL Server Logins for Managed SQL Servers

Open the SQL Server Logins for Administrators dialog box as described in the procedure on page 2-7.

1. Select the SQL Server Level tab. The SQL Server Level tab shows a list of all current policy region SQL Server login assignments for the administrator.

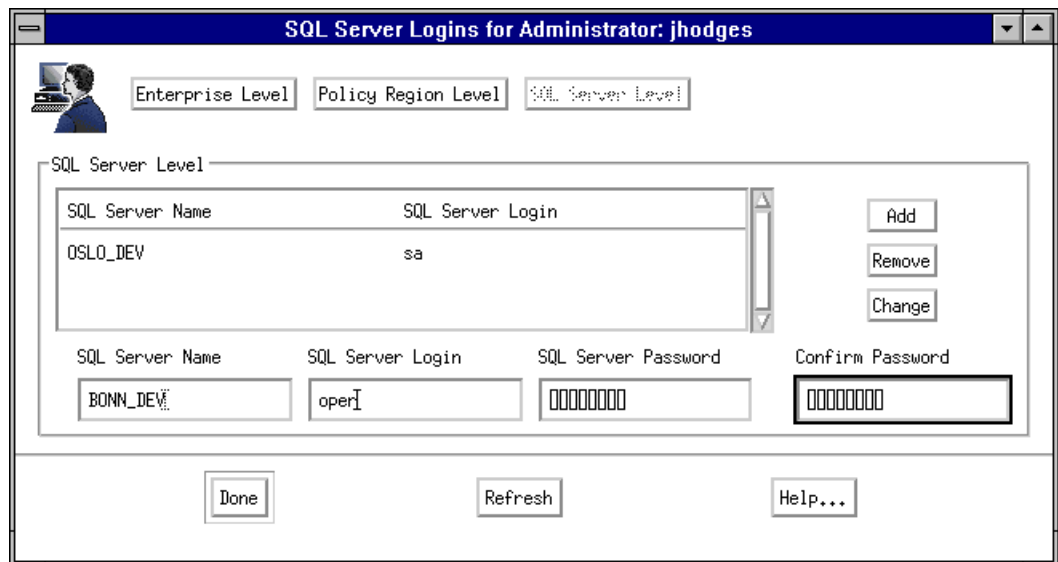


Figure 2-6: SQL Server Logins for Administrators: SQL Server Level dialog box

2. In the SQL Server Level group box, specify the list of managed SQL Servers and associated SQL Server logins you want to assign to the administrator, one pair at a time. You may add, remove, or change managed SQL Server login assignments until you are satisfied with the list. In Figure 2-6, Tivoli administrator “jhodes” is associated with the SQL Server login “sa” in SQL Server OSLO_DEV and data is entered to associate “jhodes” with the SQL Server login “oper” in SQL Server BONN_DEV.
3. When you are done making SQL Server login assignments for the administrator, choose Done to close the dialog box.

Configuring Policy Regions for Managed SQL Server Resources

Administrators manage SQL Server by creating managed SQL Server resources within policy regions. Managed SQL Servers can reside in a policy region along with other types of resources (managed nodes, task libraries, and so forth), which allows you to group and organize resources to suit your needs.

However, before you can create a managed SQL Server resource within a policy region, you must add the *ManagedSQLServer* resource type to the policy region's list of valid resources. Doing this adds the *ManagedSQLServer* command to the policy region's Create menu.

In addition to the *ManagedSQLServer* resource type, you should add all other Enterprise SQL Server Manager resources too. Doing so enables all Enterprise SQL Server Manager management functions for managing SQL Servers in the policy region.

The list of available Enterprise SQL Server Manager resource types includes:

- *ManagedSQLServer*
- *SQLDatabase*
- *SQLDatabaseProfile*
- *SQLDatabaseProfileMgr*
- *SQLDataTypeProfile*
- *SQLDbDeviceProfile*
- *SQLDefaultProfile*
- *SQLDumpDeviceProfile*
- *SQLGroupProfile*
- *SQLIndexProfile*
- *SQLLoginProfile*
- *SQLProcedureProfile*
- *SQLRemoteServerProfile*
- *SQLRuleProfile*
- *SQLSegmentProfile*
- *SQLServerProfile*
- *SQLServerProfileMgr*
- *SQLTableProfile*

- *SQLTriggerProfile*
- *SQLUserProfile*
- *SQLViewProfile*

► **Note**

Adding a resource type to a policy region's list of valid resources requires the **senior** role.

To add all the Enterprise SQL Server Manager resource types to a policy region's list of valid resources, follow these steps:

1. Open the policy region you want to modify.
2. From the policy region's Properties menu, choose Managed Resources. The Set Managed Resources dialog box opens:

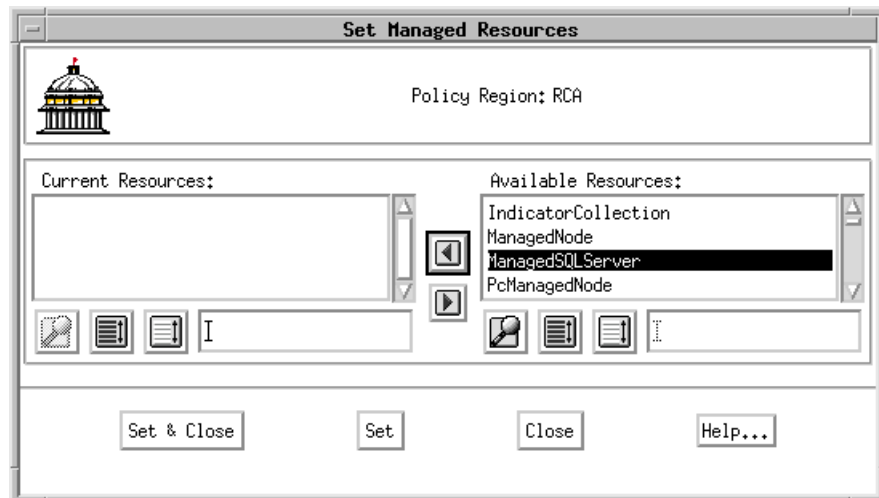


Figure 2-7: Set Managed Resources dialog box

3. In the Available Resources list box, select the *ManagedSQLServer* resource type, along with the remaining Enterprise SQL Server Manager resource types (listed on page 2-11). You can hold down the Shift key as you select each resource type to add it to the current selection.
4. Select the left-pointing arrow to move the selected resource types to the Current Resources list box.
5. Choose Set & Close to add the resource type to the policy region's list of valid resources and close the Set Managed Resources dialog box.

6. Repeat these steps for each policy region in which you want to create managed SQL Server resources.

Creating a Managed SQL Server Resource in a Policy Region

To perform Enterprise SQL Server Manager administration tasks on a SQL Server, you must create a corresponding managed SQL Server resource in a policy region. Doing this registers the managed resource in the TMR database. Creating a managed SQL Server resource also establishes which machine will serve as the management host for the managed SQL Server.

Enterprise SQL Server Manager Management Hosts

A **management host** is the TME client on which the Enterprise SQL Server Manager processing is to occur on behalf of the managed SQL Server. Creating a managed SQL Server resource also establishes a logical connection between the managed SQL Server and its management host. The management host may be any TME client on which Enterprise SQL Server Manager is installed.

The following figures show different configurations you should consider when choosing a management host.

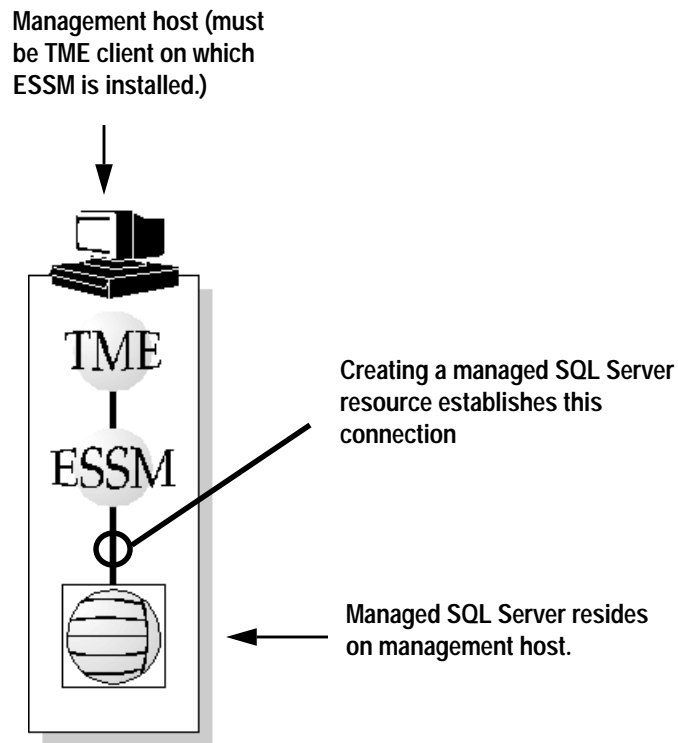


Figure 2-8: Managed SQL Server resides on management host

In Figure 2-8, the managed server's host and its management host are the same machine. SQL Server is running on a TME client with Enterprise SQL Server Manager installed. This is the most common configuration.

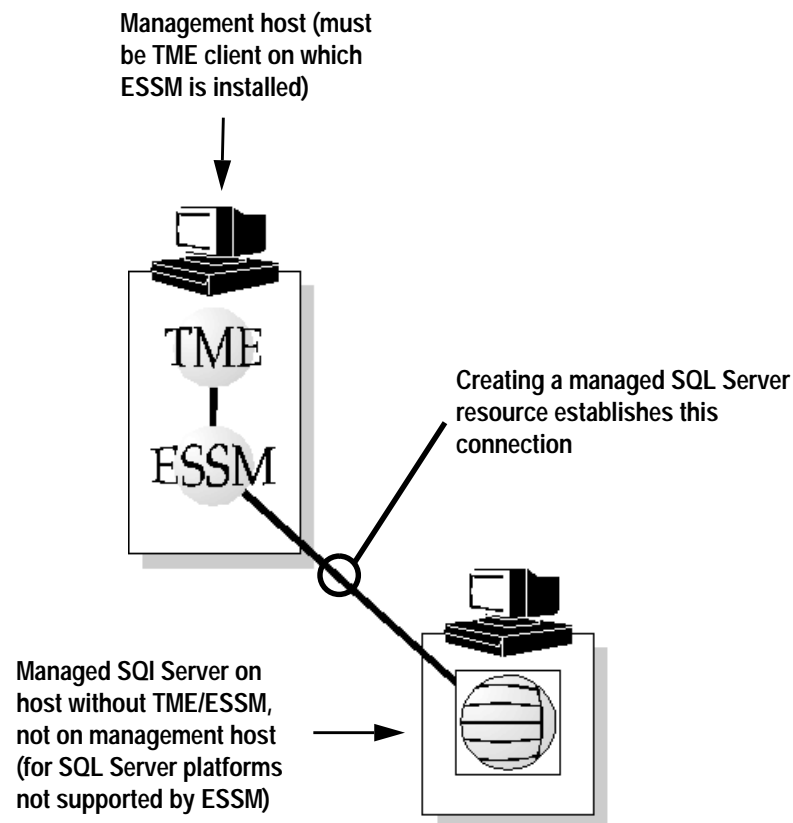


Figure 2-9: Managed SQL Server and remote management host

In Figure 2-9, the managed server's host and its management host are two different machines. SQL Server is running on a computer without TME, and its management host is a TME client running Enterprise SQL Server Manager.

The advantage to this configuration is that you can use Enterprise SQL Server Manager to manage SQL Servers running on platforms not supported by Enterprise SQL Server Manager. This is possible because communication between Enterprise SQL Server Manager and the managed SQL Server occurs using Sybase connectivity.

Creating a Managed SQL Server Resource

To create a managed SQL Server resource in a policy region, follow these steps:

► **Note**

Creating a managed SQL Server resource in a policy region requires the server role.

1. Open the policy region window in which you want to create the managed SQL Server resource. The policy region must be configured to allow this resource type (see “Configuring Policy Regions for Managed SQL Server Resources” on page 2-11).
2. From the Create menu, choose ManagedSQLServer. The Manage SQL Server dialog box opens:

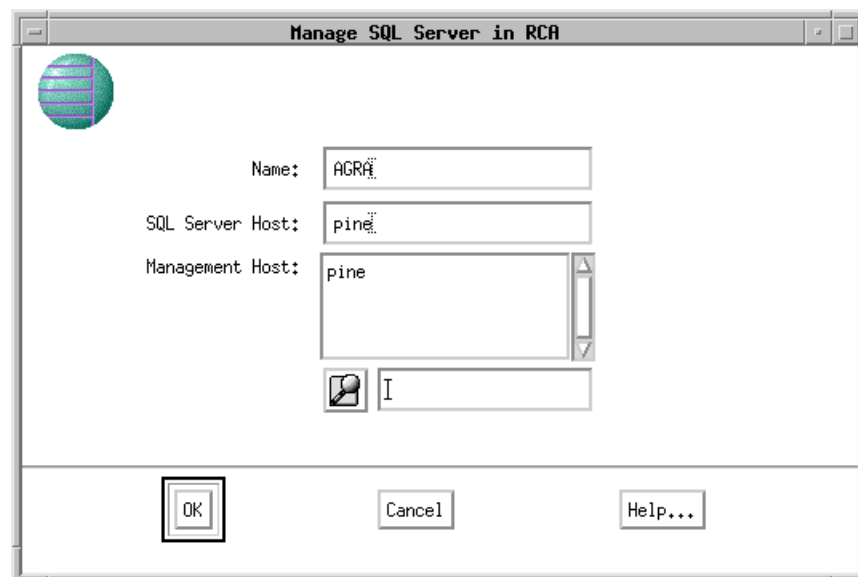


Figure 2-10: Manage SQL Server dialog box

3. In the Name box, enter the name of the SQL Server you want to add to the policy region as a managed resource.
4. In the SQL Server Host box, enter the name of the machine on which SQL Server is running.
5. In the Management Host box, select the TME client that is to be the server’s management host (see “Enterprise SQL Server Manager Management Hosts” on page 2-13).

6. Choose OK to create the new managed SQL Server resource in the policy region.
7. Repeat these steps, beginning with step 2, for each SQL Server resource you want to add to the policy region.

After you create a managed SQL Server, the TMP adds a SQL Server icon to the policy region. Figure 2-11 illustrates a policy region window with several managed SQL Servers and profile managers. For information about creating profile managers, see Chapter 11, “Enterprise SQL Server Manager Profile Managers.”

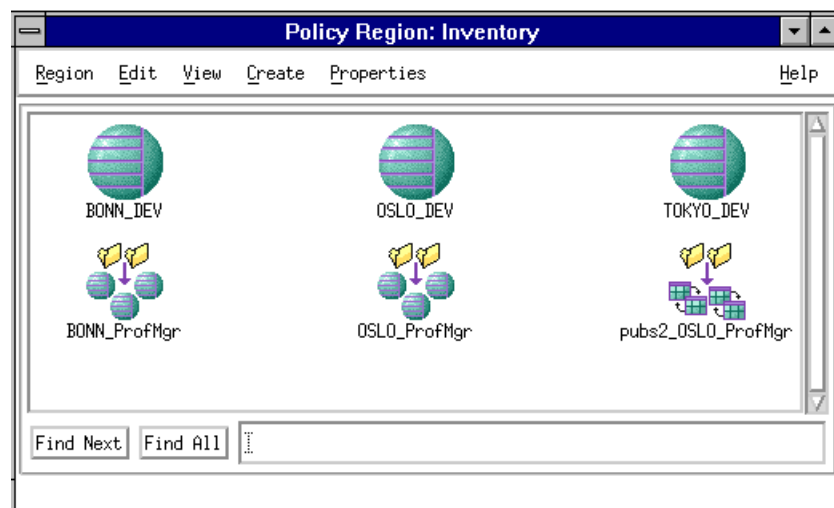


Figure 2-11: Policy region with managed ESSM resources

You start Enterprise SQL Server Manager from any managed SQL Server or profile manager. For information about starting Enterprise SQL Server Manager and using the SQL Server window, see Chapter 3, “Enterprise SQL Server Manager Windows and Dialog Boxes.”

3

Enterprise SQL Server Manager Windows and Dialog Boxes

What's in This Chapter

In addition to the TME desktop, Enterprise SQL Server Manager uses three custom windows, the SQL Server window, the SQL Server Profile Manager window, and the Database Profile Manager window. These windows are generically called Voyager windows. A Voyager window lets you see and operate on all the objects in SQL Server or a profile manager.

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

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

Starting Enterprise SQL Server Manager

To start Enterprise SQL Server Manager, in a Tivoli policy region window, double click the icon of a managed SQL Server or a profile manager. A Voyager window opens. The Voyager window for each managed SQL Server or profile manager is independent of other Voyager windows and the Tivoli desktop. You can close the Tivoli desktop and continue to work in open Voyager windows.

Stopping Enterprise SQL Server Manager

To stop Enterprise SQL Server Manager, close all open Voyager windows. Closing the Tivoli desktop does not close the Voyager windows. You must close them separately.

Using the Voyager Window

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

Figure 3-1 shows a view of the SQL Server window and indicates the main features of the Voyager display. A description of each feature follows.

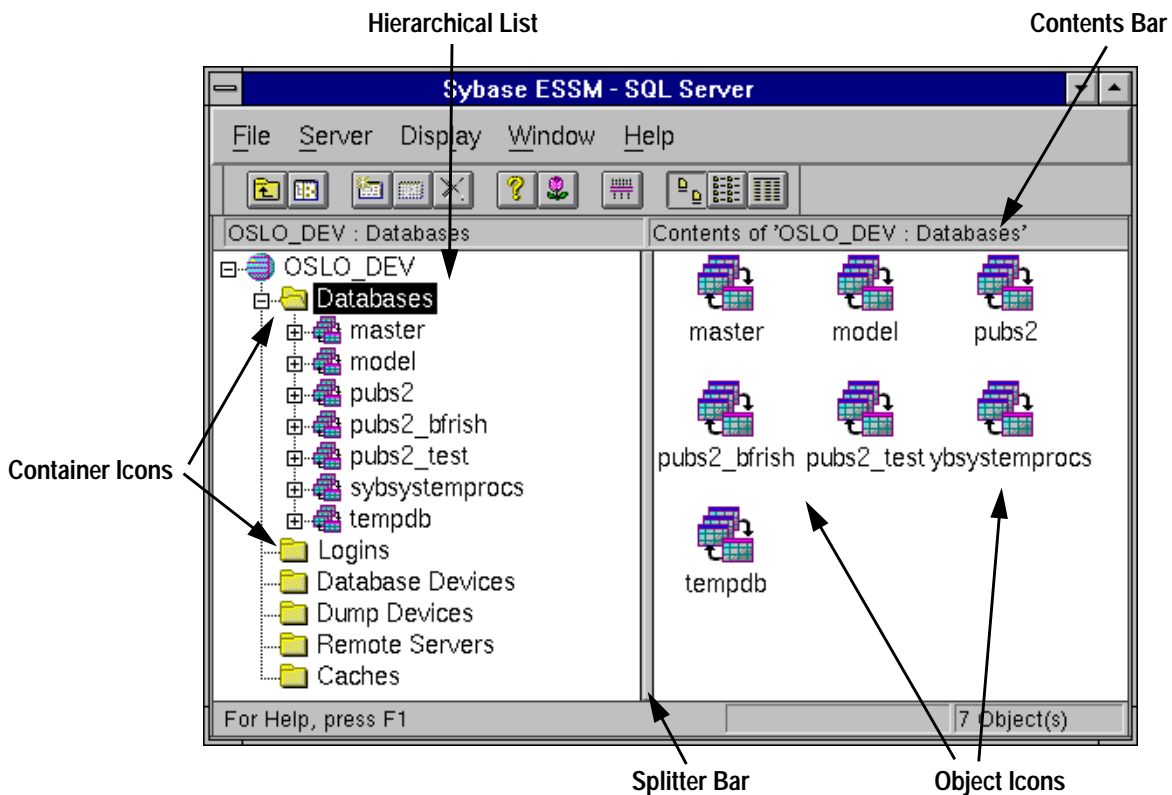


Figure 3-1: Enterprise SQL Server Manager Voyager window

- The window is split into left and right panes:
 - The left pane contains a hierarchical list, which shows SQL Server and its 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 represents all databases in SQL Server. 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 displays objects for only one managed SQL Server at a time.

Moving Through a Voyager Window

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 the list 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. Some reasons for opening additional Voyager windows are:

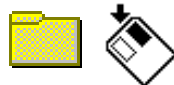
- To view different parts of a long Voyager hierarchy at the same time
- To perform a drag-and-drop operation involving two different types of objects in the same database or involving an object that is scrolled out of view
- To view objects in multiple containers simultaneously

To open a new window, choose **New Window...** from the **Window** menu. Enterprise SQL Server Manager creates a copy of the entire Voyager window. 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.

Shortcuts



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. Enterprise 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.

Figure 3-2 shows an example of each format:

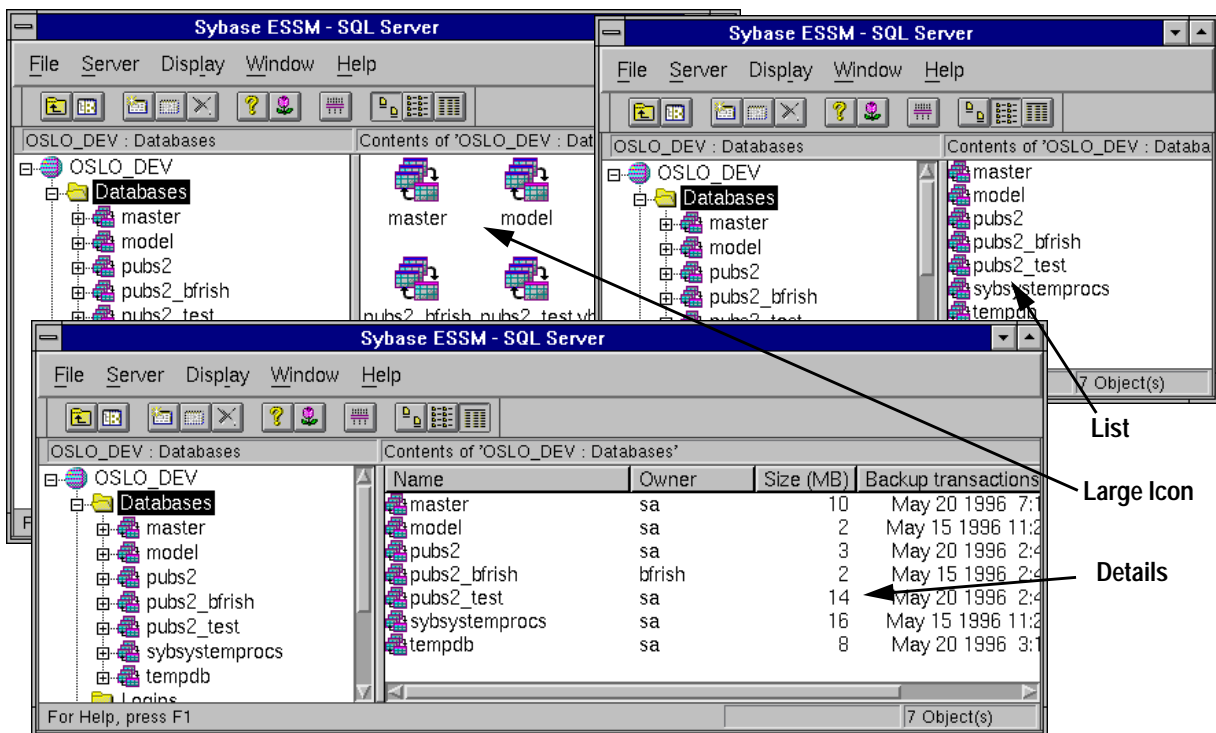


Figure 3-2: Object display formats

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, Figure 3-3 shows the Details view of the databases in SQL Server sorted by database name and by size.

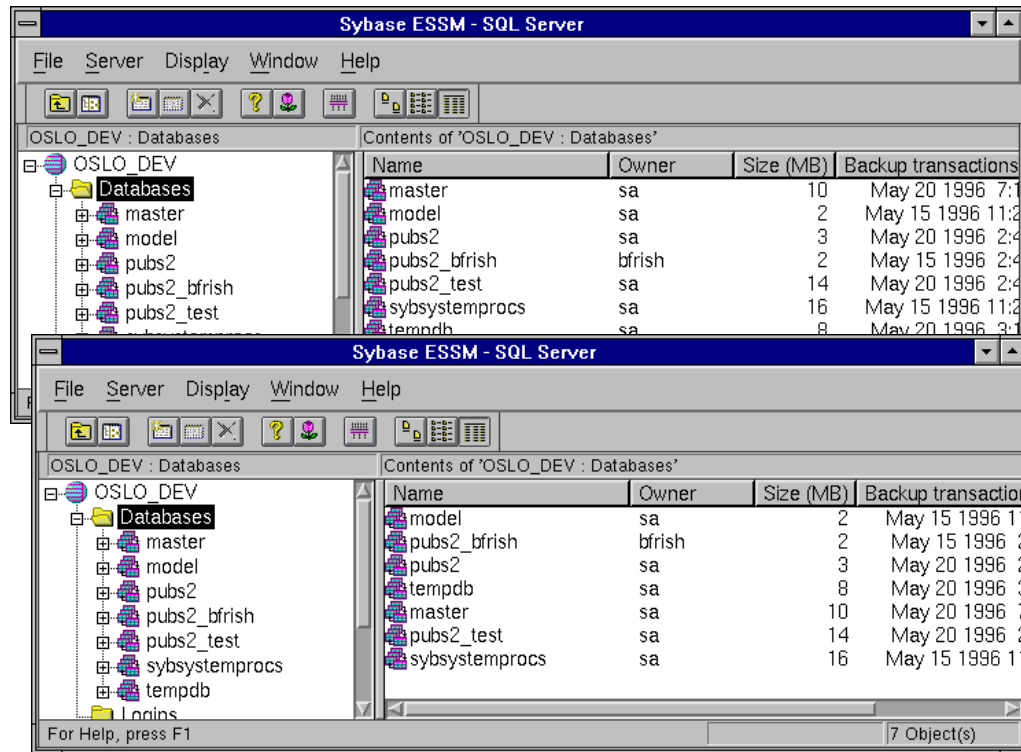


Figure 3-3: Details view of databases, sorted by name and by size

Shortcuts



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 the Display

You can limit the display to objects that match a specific character string. 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.

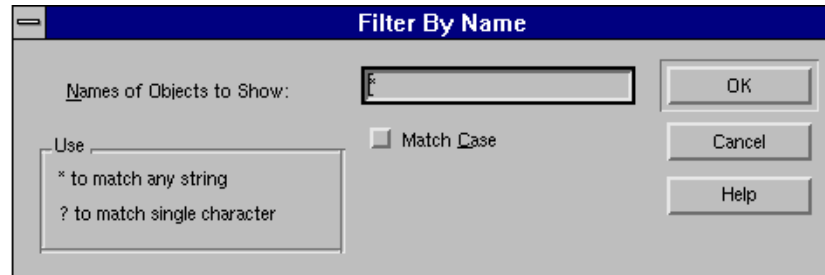


Figure 3-4: Filter by Name dialog box

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 wildcard 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, select the Match Case button. For example, the string *gary** with Match Case unselected might return icons with names beginning with *Gary*, *GARY*, and *gary*. With Match Case selected, it returns only strings that begin with *gary*.
4. Click OK. Enterprise 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 Enterprise 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.

Enterprise 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. For example, to add a user to a group you can drag a user icon onto a group icon, or you can drag a group icon onto a user icon. In either case, the Group dialog box opens.

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 (unless your intended target is illegal, in which case, you cannot complete the drag-and-drop operation).

In the Figure 3-5, 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.

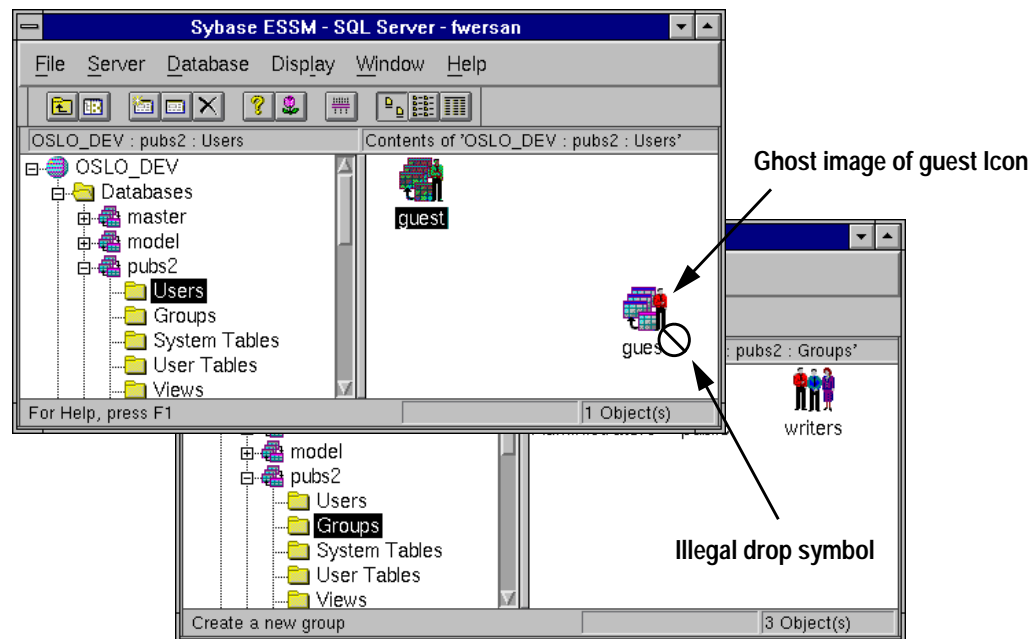


Figure 3-5: Drag-and-drop symbol

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

Using Drag-and-Drop Between Windows

You can use drag-and-drop between windows only if the two windows are views of the same SQL Server. You cannot drag-and-drop between:

- Windows for different SQL Servers
- Voyager windows and TME windows
- SQL Server windows and Profile Manager windows
- Profile manager windows

Menus and Toolbars

This section describes the menus and toolbars in the Voyager window.

Standard Menus

The menu bar in a Voyager window always has the following menus:

- File
- Display
- Window
- Help

A SQL Server window (the basis for the discussion in this chapter) has an additional standard menu – Server. Profile manager windows have a different set of standard menus. Profile manager menus are described in “Profile Manager Menus” in Chapter 11, “Enterprise SQL Server Manager Profile Managers”.

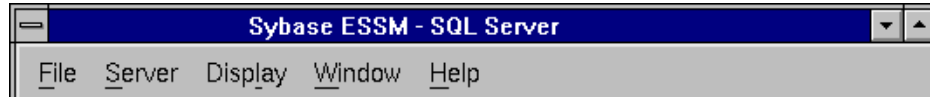


Figure 3-6: Standard menu bar

Unlike **context-sensitive menus**, which change in the menu bar according to the context of your current activity, standard menus are always present on the menu bar. The following table summarizes the activities to which the standard menus provide access.

Table 3-1: Activities available from standard menus

Menu	Activities
File	<ul style="list-style-type: none"> • Close the active Voyager window.
Server	<ul style="list-style-type: none"> • 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 • Open a new Voyager window • Stop 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 • Filter the display using a character string
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 Enterprise SQL Server Manager online help contents • Search for an online help topic • Display information about how to use online help • Display the About dialog box for Enterprise 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 creating, deleting, updating, and administering databases.

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 Menu

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, which was activated when the *authors* table icon was selected.

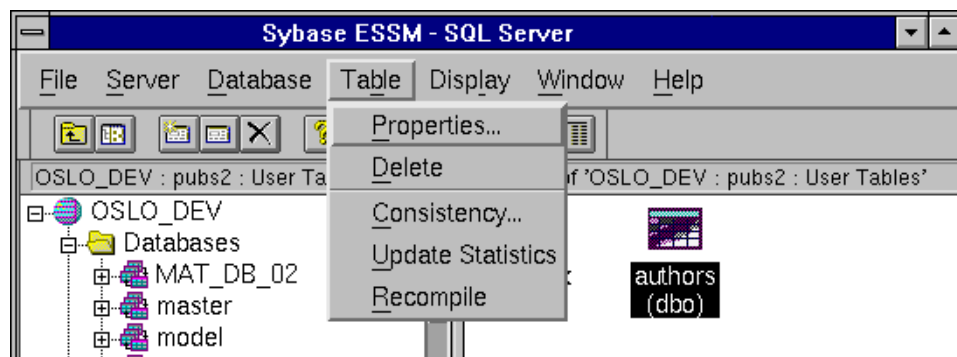


Figure 3-7: Pull-down menu for a table

Shortcut Menus

To activate a shortcut (pop-up) menu, click the right mouse button over a container icon or object icon representing a SQL Server or database object. A menu is displayed. From this menu, choose the appropriate menu command.

Shortcut menus have different menu commands according to their context:

Table 3-2: Shortcut menu commands

Context	Commands
SQL Server icon	<ul style="list-style-type: none"> • Same commands as the Server menu
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 • List • 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.

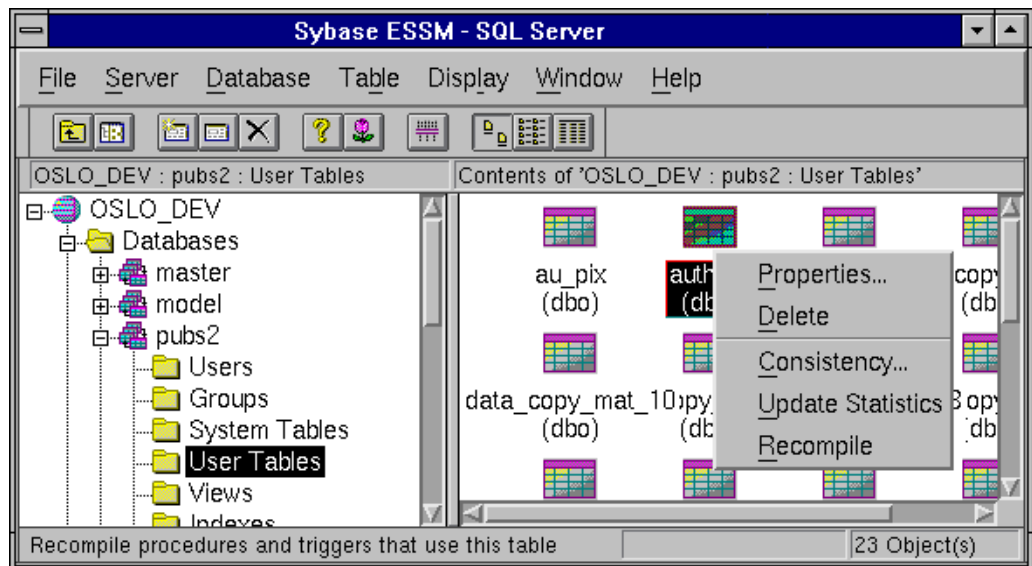


Figure 3-8: Shortcut menu for a table

Toolbar and Status Bar

The Enterprise 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.

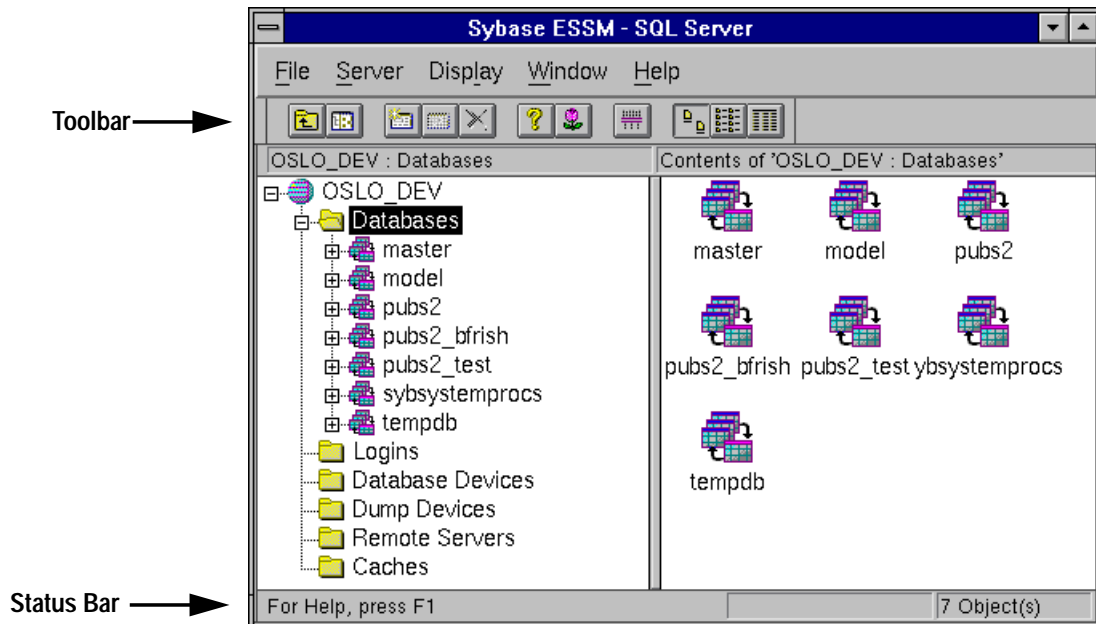


Figure 3-9: Toolbar and status bar

You can hide and redisplay the toolbar or status bar. To display or hide the toolbar or status bar, toggle the **Toolbar** or **Status Bar** command on the **Display** menu.

Standard Toolbar

The standard toolbar consists of the following controls:

- A text box that identifies the SQL Server you are viewing
- Buttons that provide a quick way to execute menu commands

Figure 3-10 illustrates the toolbar. Table 3-3 describes the controls.

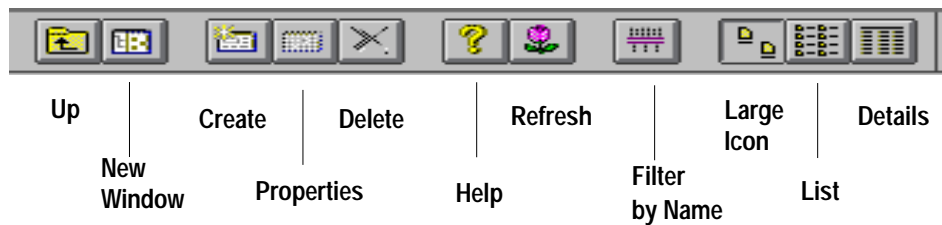


Figure 3-10: Enterprise SQL Server Manager toolbar

Table 3-3: Toolbar controls

Control	Function
Up	Moves the Voyager display up one level in the object hierarchy. For example, if the current selection is the Groups container in a database, clicking the Up button moves the selection up to the database icon.
New Window	Opens a 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, Enterprise SQL Server Manager displays a **Tooltip**—a floating label that appears 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.

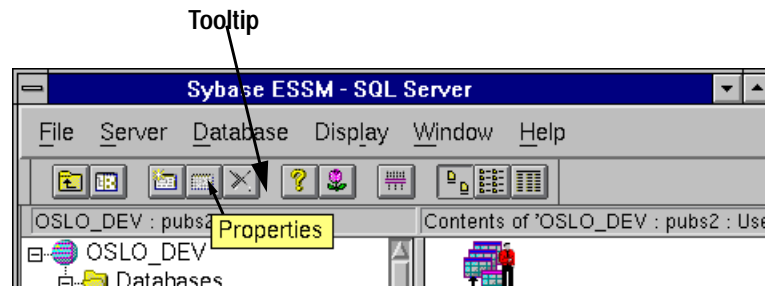


Figure 3-11: Tooltip for the Refresh tool bar button

Status Bar

The status bar is an information display bar located at the bottom of the application window. In Enterprise 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.

Figure 3-12 shows the status bar for a display in which the Properties command is selected in the Table menu and filtering by name is activated.

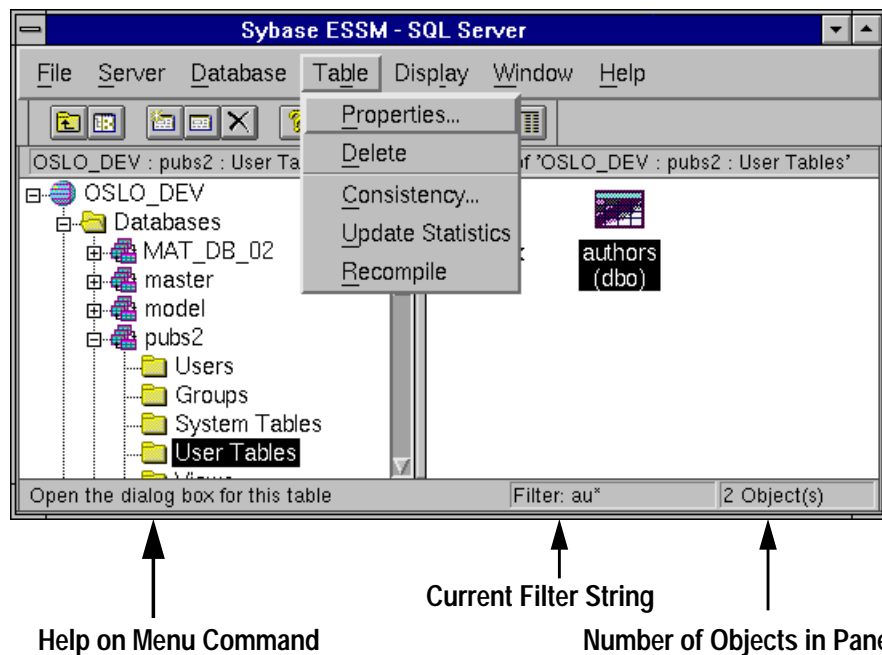


Figure 3-12: Status bar

Types of Dialog Boxes

Enterprise SQL Server Manager has the following types of dialog boxes:

- Object dialog boxes, which hold information about SQL Server and database objects
- Command dialog boxes, for specifying how to execute menu commands
- Confirmation dialog boxes, which ask you to confirm an action, such as deleting an object
- Message dialog boxes, which 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 Properties tab on the Table dialog box is an example.

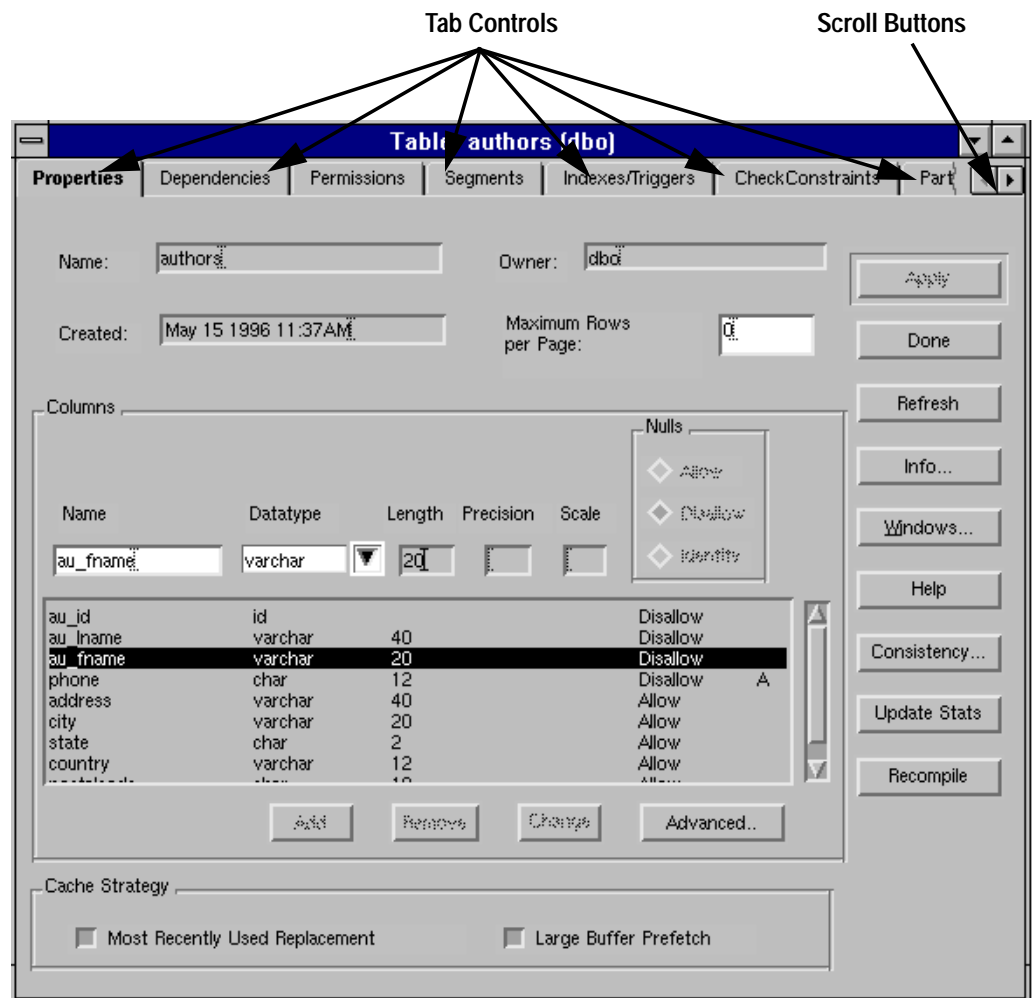


Figure 3-13: Properties tab of a multiple-tab Table dialog box

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:

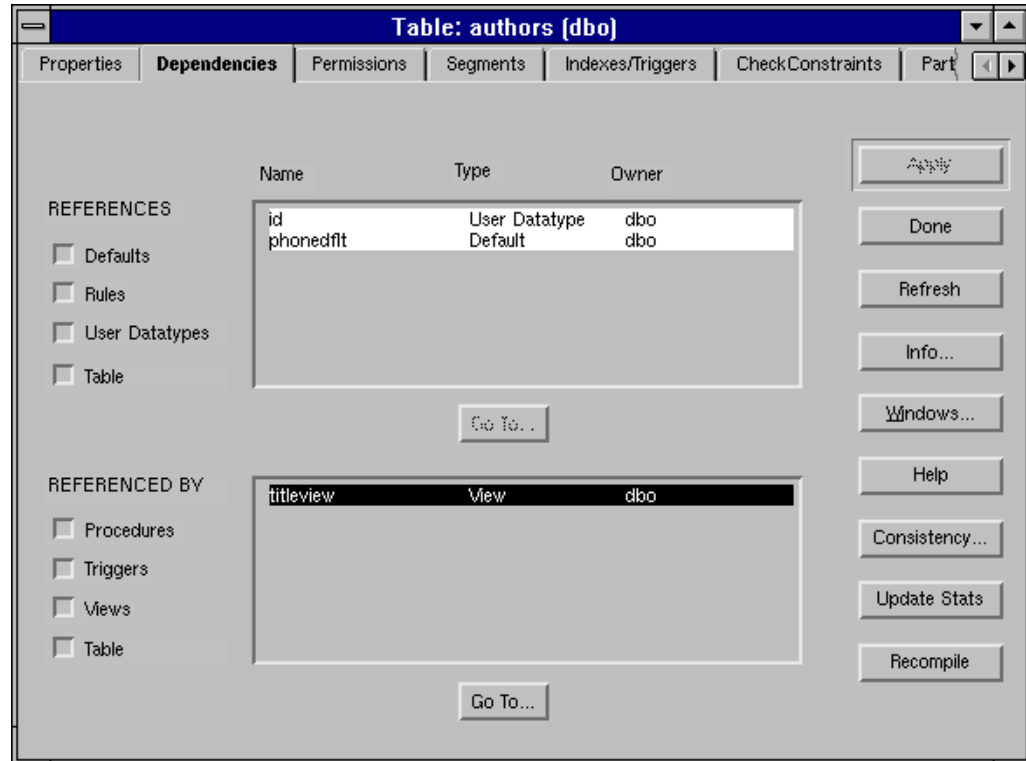


Figure 3-14: Dependencies tab of the multiple-tab Table dialog box

Selecting Multiple Objects

Some dialog boxes have multi-column list boxes. When you select objects to move from one column to the other, you can select ranges of objects or multiple non-contiguous objects.

To select a range of objects:

1. Select the first object in the list.
2. Locate the last item in the range. This may involve scrolling the list.
3. Hold down the Shift key and select the last item in the range. All items selected are highlighted.

To select or clear non-contiguous objects, hold down the Control key as you click on each object you want to select or clear.

Navigating Between Objects

Enterprise 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..., Enterprise SQL Server Manager opens the Properties tab of the *titleview* view.

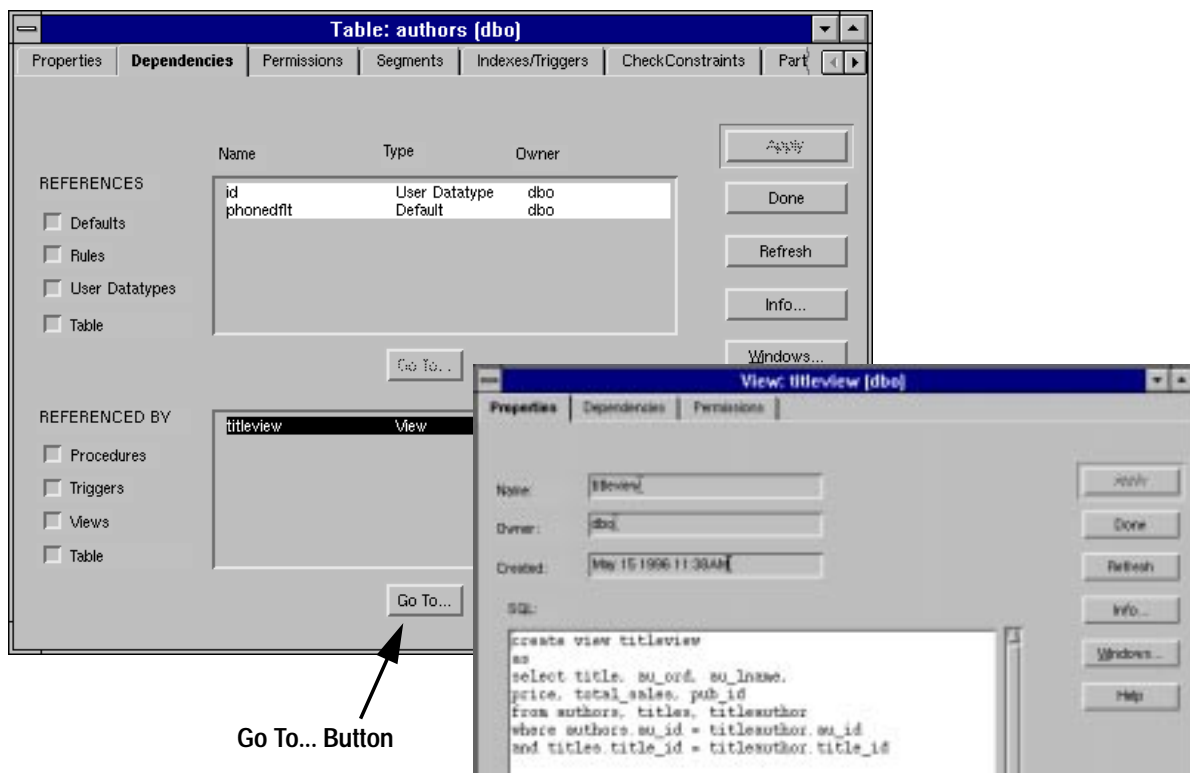


Figure 3-15: Navigating to a related object

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:

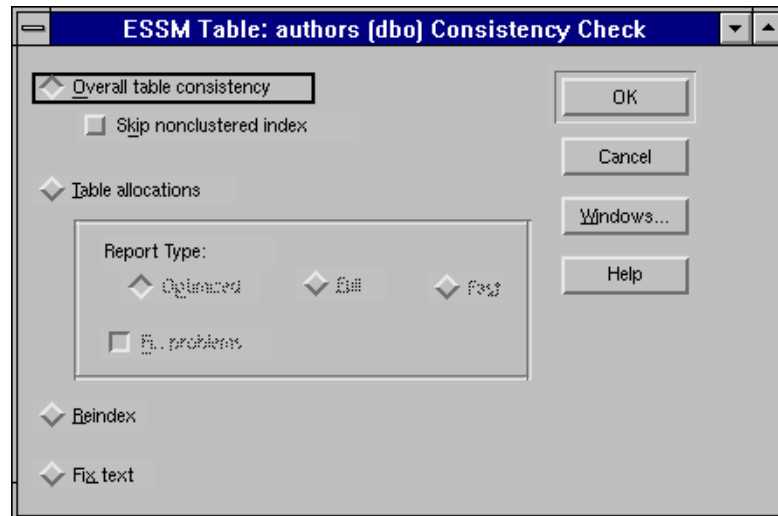


Figure 3-16: Example of a command dialog box

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.

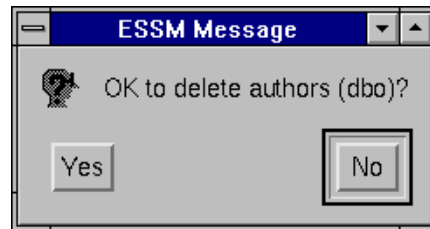


Figure 3-17: Example of a confirmation dialog box

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 Enterprise SQL Server Manager or SQL Server. Enterprise 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 3-18: 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.

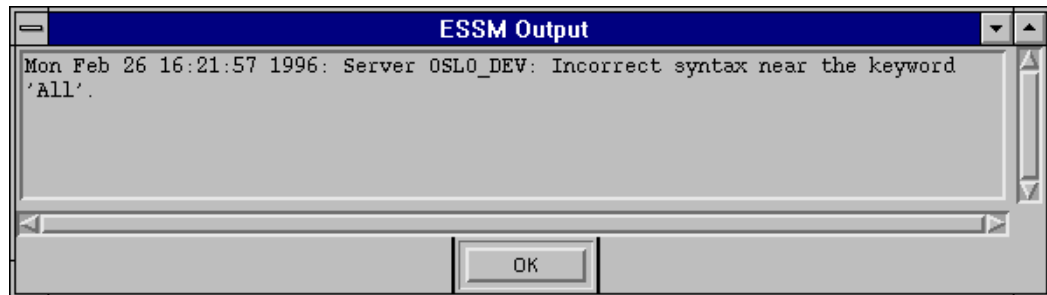


Figure 3-19: Output dialog box

► **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.

Closing a Message Dialog Box

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

Refreshing Window and Dialog Box Displays

As you work, information in open dialog boxes and the Voyager window can get unsynchronized. To update the contents of the active window or a dialog box, refresh the display. Use either of the following methods:

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

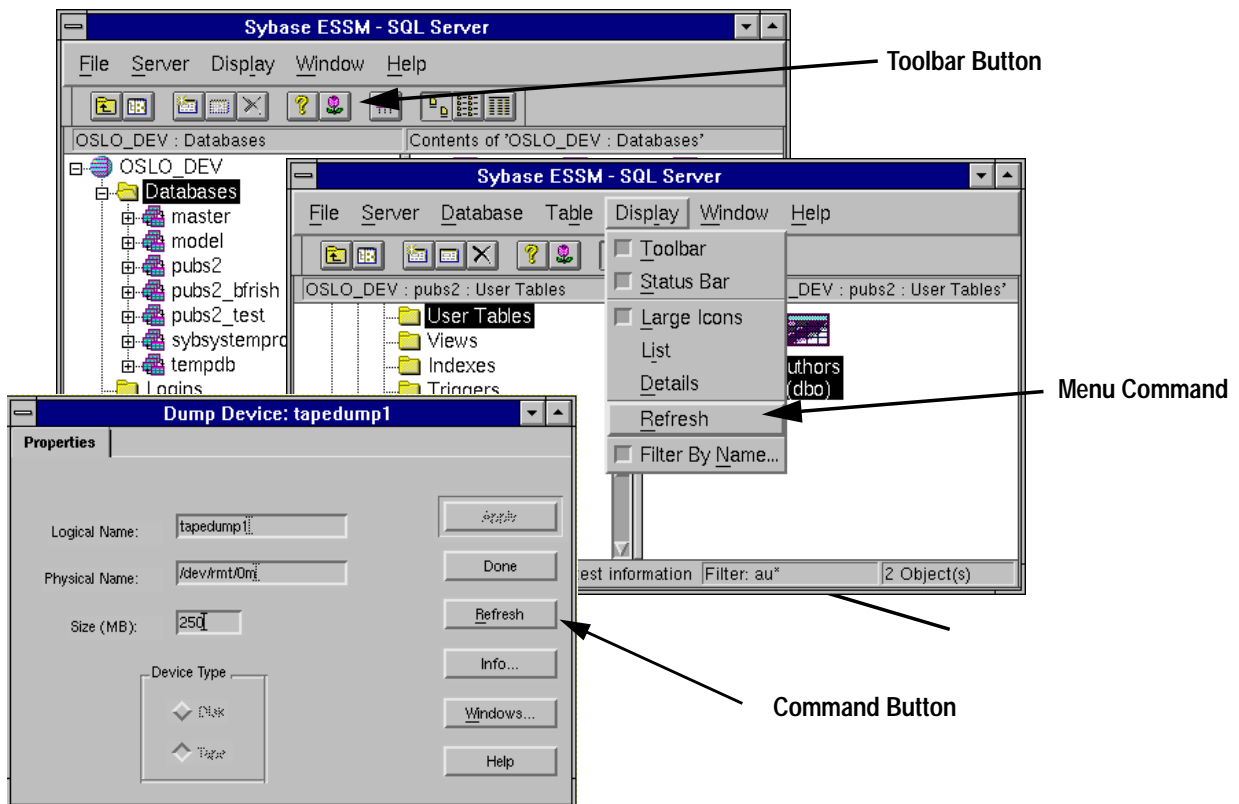


Figure 3-20: Methods of refreshing a window or dialog box

Moving Between Open Windows and Dialog Boxes

Enterprise 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.

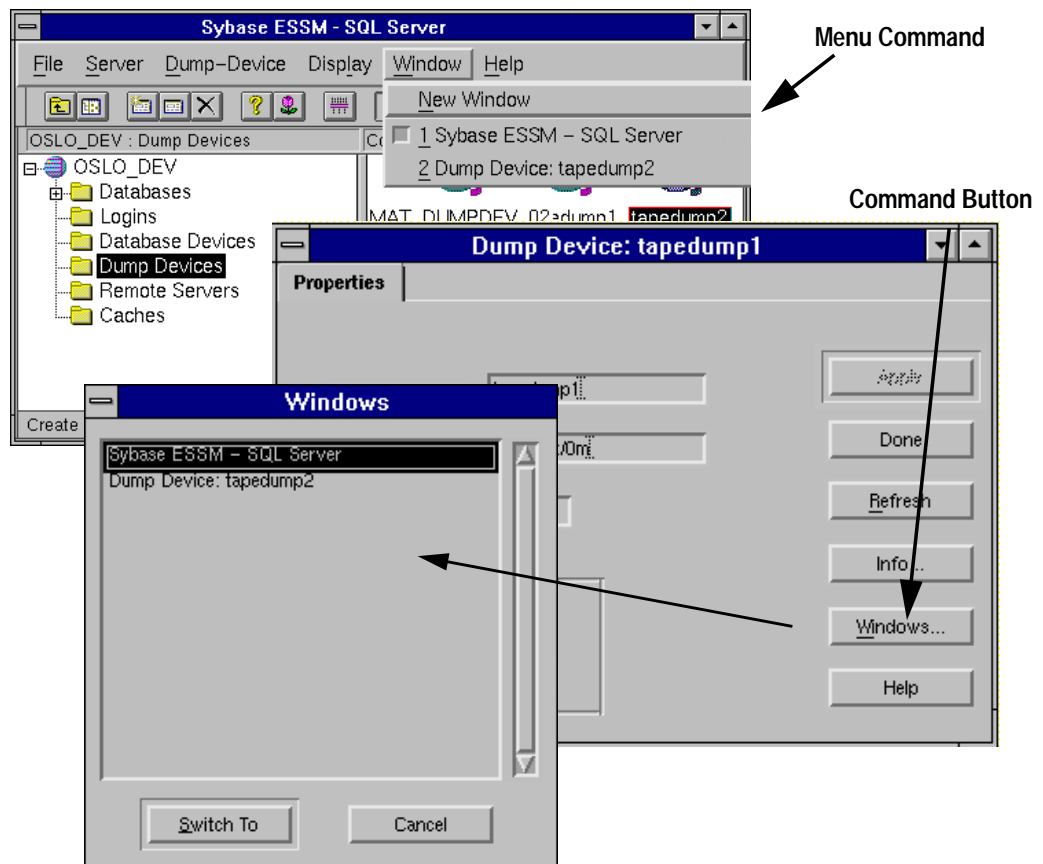


Figure 3-21: Methods of moving between open windows and dialog boxes

4

Managing the SQL Server Operating Environment

What's in This Chapter

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

- Start and stop SQL Server
- Get SQL Server status
- Start SQL Server monitor
- Display information about 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

Managing SQL Servers on Non-ESSM Hosts

Enterprise SQL Server Manager allows you to manage SQL Server installations on host machines on which Enterprise SQL Server Manager is not installed, including hosts that Enterprise SQL Server Manager does not support. However, the following restrictions apply:

- You cannot use Enterprise SQL Server Manager to start SQL Server. This includes starting SQL Server using:
 - Start SQL Server dialog box
 - `sstartserver` command
 - Restart Server After Shutdown check box on the Stop SQL Server dialog box
 - `-restart` option of the `sstopserver` command
- Enterprise SQL Server Manager does not automatically delete operating system files designated as a device's physical device name when you delete database devices.

Starting SQL Server or Backup Server

You start SQL Server or Backup Server from a policy region window.

	TME	ESSM	SQL Server
Required roles	any	server	System Administrator

To start SQL Server:

1. In the policy region window, click the right mouse button over the icon of the SQL Server you want to start. From the popup menu, choose Start. The Start SQL Server dialog box opens.

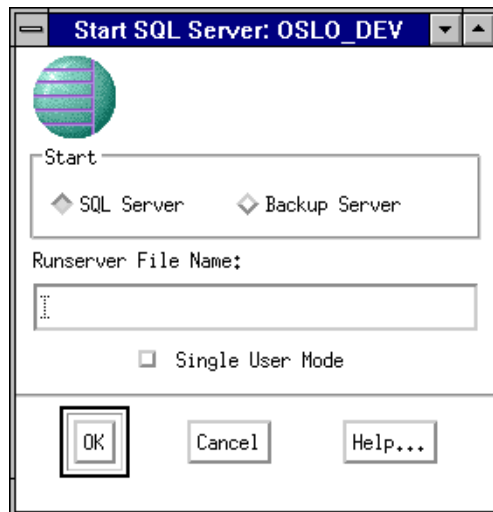


Figure 4-1: Start SQL Server dialog box

2. In the Start group box, select SQL Server or Backup Server.
3. In the Runserver File Name box, type the full pathname of the SQL Server you want to start.
4. To run SQL Server or Backup Server in single user mode, check the Single User Mode box.
5. Click OK.

Displaying the Status of SQL Server

	TME	ESSM	SQL Server
Required roles	any	any	none

To display the status of SQL Server:

1. In the policy region window, click the right mouse button over the icon of the SQL Server for which you want status.
2. From the popup menu, choose Status. The SQL Server Status dialog box opens. The dialog box shows the current status of the SQL Server. The possible statuses are:
 - Alive
 - Dead
 - Hung
 - Host not found
 - Server not found



Figure 4-2: SQL Server Status dialog box

Stopping SQL Server or Backup Server

You can stop SQL Server or Backup Server from a policy region window or from a SQL Server window.

	TME	ESSM	SQL Server
Required roles	any	server	System Administrator

Stopping SQL Server from a Policy Region Window

1. In the policy region window, click the right mouse button over the icon of the SQL Server you want to stop. From the popup menu, choose Stop. The Stop SQL Server dialog box opens.

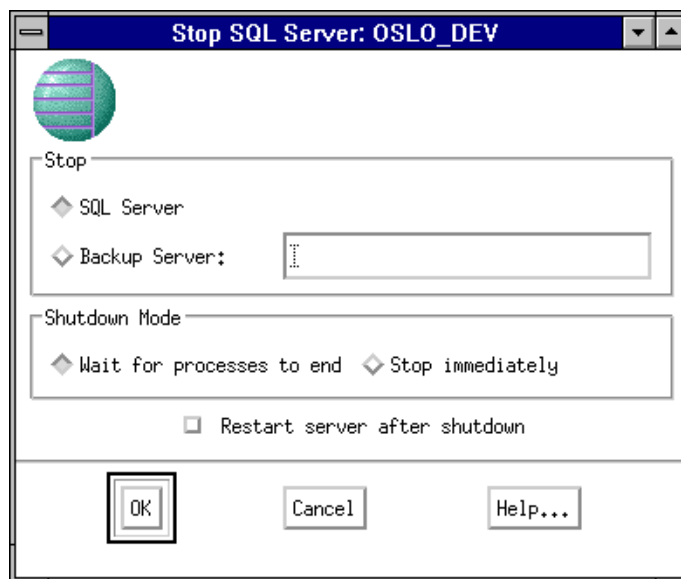


Figure 4-3: Stop SQL Server dialog box

2. In the Stop group box, select SQL Server or Backup Server. If you select Backup Server, enter the name of the Backup Server you want to stop.
3. In the Shutdown Mode group box, select an option.

Stop Immediately—Specifies that shutdown is to stop the server immediately.

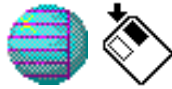
Wait for Processes to End—Specifies that shutdown is to wait for existing processes to exit before stopping the server.

4. If you want to restart SQL Server after the shutdown, check the Restart Server After Shutdown box.
5. Click OK.

Stopping SQL Server from a SQL Server Window

From the Server menu, choose Stop→ Immediately or Stop→ When Ready.

Shortcuts



Click the right mouse button over the SQL Server icon and choose Stop from the shortcut menu.

Starting SQL Server Monitor from Enterprise SQL Server Manager

To make it easier for you to monitor SQL Server, you can start SQL Server Monitor from the Tivoli desktop. The default process for starting SQL Server Monitor is based on the following assumptions:

- SQL Server Monitor client is installed
- SQL Server Monitor client executable is located under the directory specified as the Sybase environment when you installed Enterprise SQL Server Manager
- SQL Server Monitor uses the same interfaces file as Enterprise SQL Server Manager
- The location of the *locales* and *charsets* files is the same as for Enterprise SQL Server Manager

If any of these assumptions are wrong, you can change the scripts that Enterprise SQL Server Manager uses to start SQL Server Monitor. See “Customizing SQL Server Monitor Launch” on page 4-8.

	TME	ESSM	SQL Server
Required roles	none	monitor	System Administrator

To start SQL Server Monitor, in a policy region window, click the right mouse button over the icon of the SQL Server you want to monitor. From the popup menu, choose Monitor. The main dialog box for SQL Server Monitor opens. It uses the SQL Server login and password assigned to you in the SQL Server Logins for Administrator dialog box.

The client is started on the machine on which the Tivoli desktop is running. Once started, the SQL Server Monitor client process is independent of the Voyager and the Tivoli Desktop.

► **Note**

Enterprise SQL Server Manager cannot start SQL Server Monitor Historical Server.

Starting the SQL Server Monitor Server

If the SQL Server Monitor server is not running, Enterprise SQL Server Manager starts it. The server runs as the user “sybase” and has the same environment as a SQL Server started by Enterprise SQL Server Manager.

When Enterprise SQL Server Manager starts the SQL Server Monitor server, it makes the following assumptions:

- The name of the SQL Server Monitor server is *SQL_Server_name_SMS*
- The run file name is *RUN_SQL_Server_name_SMS*

If these assumptions are not true, you must start the SQL Server Monitor server and choose the SQL Server Monitor server that you want to connect to in the SQL Server Monitor client login screen. You can also edit the SQL Server Monitor Server startup script to have it reflect your installation.

If you want SQL Server Monitor Server to use your resource file, you must append the path of your home directory to the variable *XUSERFILESEARCHPATH*. By default, Enterprise SQL Server Manager sets *XUSERFILESEARCHPATH* to */work/framework/usr/lib/X11/app-defaults*.

When you exit SQL Server Monitor client, Enterprise SQL Server Manager does not stop the SQL Server Monitor server.

Customizing SQL Server Monitor Launch

If your installation of SQL Server Monitor does not follow the assumptions used by Enterprise SQL Server Manager, you can edit the SQL Server Monitor startup scripts. If the location of the locales and charsets files is different, create the variable *SQLMON_INSTALLATION*.

To create *SQLMON_INSTALLATION*, use the Tivoli command *odadmin* as follows:

1. Get the current Tivoli environment:

```
odadmin environ get > environfile
```

The file has contents similar to the following:

```
LIBPATH=
TZ=US/Eastern
NLSPATH=/work/tivoli/usr/local/Tivoli/msg_cat/%L/%N
.cat:/work/tivoli/usr/local/Tivoli/msg_cat/%L/%N.ct
ESSM_INSTALLATION=/work/tivoli/usr/local/Tivoli/bin
/solaris2/SSM/SYBASE
ESSM_SYBASE=/usr/local/system1002
ESSM_INTERFACES=/usr/local/system1002/interfaces
PATH=/work/tivoli/usr/local/Tivoli/bin/solaris2/bin
:/bin:/usr/bin
```

2. Edit the file to add a definition of *SQLMON_INSTALLATION*.

► **Note**

Do not change the definitions of the variables *ESSM_INSTALLATION*, *ESSM_SYBASE*, or *ESSM_INTERFACES*, or Enterprise SQL Server Manager will not work.

3. Update the Tivoli environment:

```
odadmin environ set < environfile
```

The scripts that Enterprise SQL Server Manager uses to start SQL Server Monitor are:

- *\$BINDIR/ssm/start_monitor_client.sh*
- *\$BINDIR/ssm/sms_startserver.sh*

If the name of the SQL Server Monitor server is not *SQL_Server_name_SMS*, or if the runfile name is not *RUN_SQL_Server_name_SMS*, edit the scripts to meet the specific needs of your installation.

The SQL Server Window

The SQL Server window is a collection of all the managed resources in SQL Server.

To open the SQL Server window:

- Double-click the SQL Server icon in the policy region window, or
- Choose **Open** from the SQL Server icon pop-up menu.

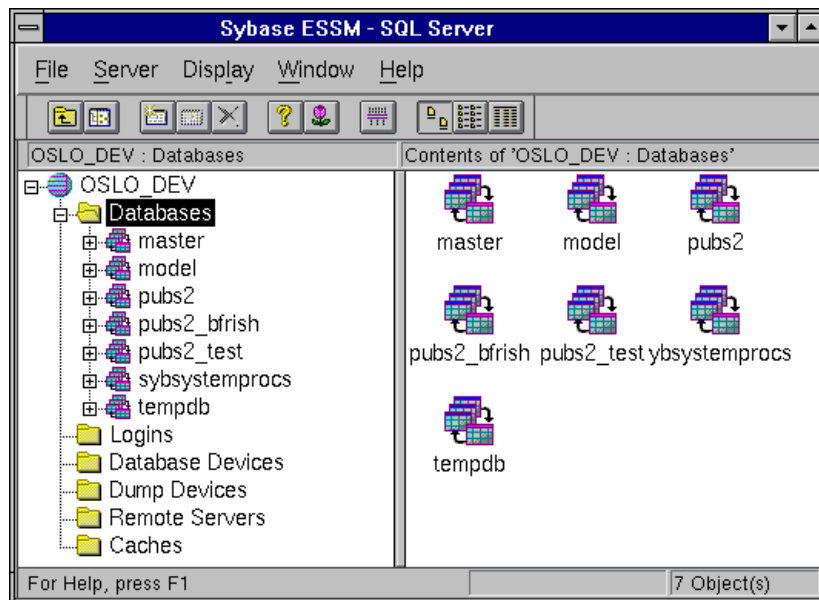



Figure 4-4: SQL Server window

The window contains icons for the following SQL Server objects:

- SQL Server
- Databases
 - Users
 - Groups
 - System Tables
 - User Tables
 - Views
 - Indexes
 - Triggers
 - Procedures

- Rules
- Defaults
- User datatypes
- Segments
- Logins
- Database devices
- Dump devices
- Remote Servers
-  Caches

Displaying SQL Server Details

In the SQL Server window, you can display details about SQL Server or any of its objects.

Examining SQL Server Properties

	TME	ESSM	SQL Server
Required roles	any	any	none

To display a dialog box that shows the properties of SQL Server, from the Server menu, choose Properties.

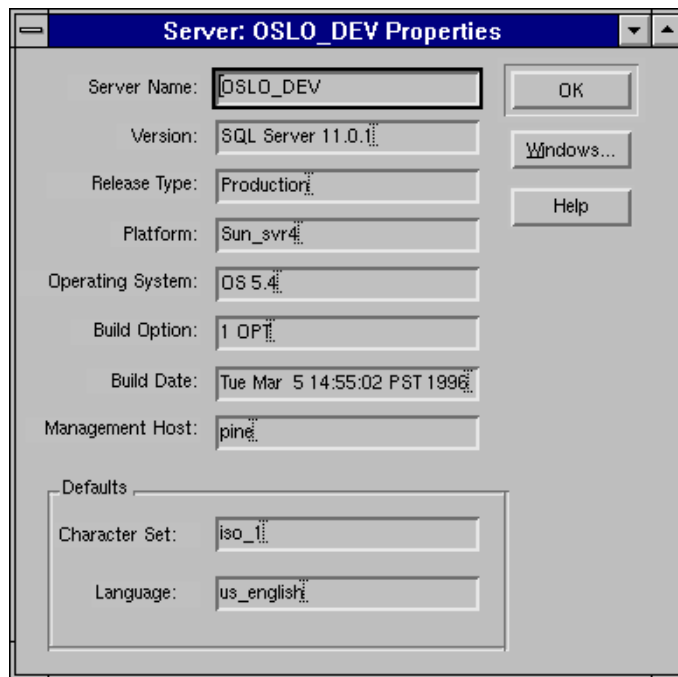


Figure 4-5: Server Properties dialog box

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

Examining SQL Server Objects

The Voyager window contains icons representing the objects that SQL Server manages. For a picture of what each type of icon looks like, see Appendix B, “Guide to Enterprise 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.

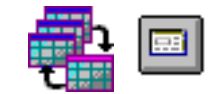
For example, to display the properties of a database device managed by SQL Server, select the database device icon and choose Properties from the Database Device menu.

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

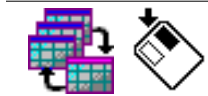
Shortcuts



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.

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 SQL Server release 10.x and SQL Server release 11.0 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 *System Administration Guide* for the appropriate SQL Server release. For a discussion of configuration issues to consider when determining optimal settings, see *Sybase Performance and Tuning Guide*.

	TME	ESSM	SQL Server
Required roles	none	server	System Administrator

In addition to the System Administrator, you must have the System Security Officer to set any of the following configuration variables:

- allow updates
- audit queue size
- password expiration interval
- remote access

► **Note**

The default character set id parameter is set during SQL Server installation and cannot be reset from within Enterprise SQL Server Manager.

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 SQL Server Release 10.x

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

1. From the Server menu, choose Configuration. The Configuration Parameters dialog box opens.

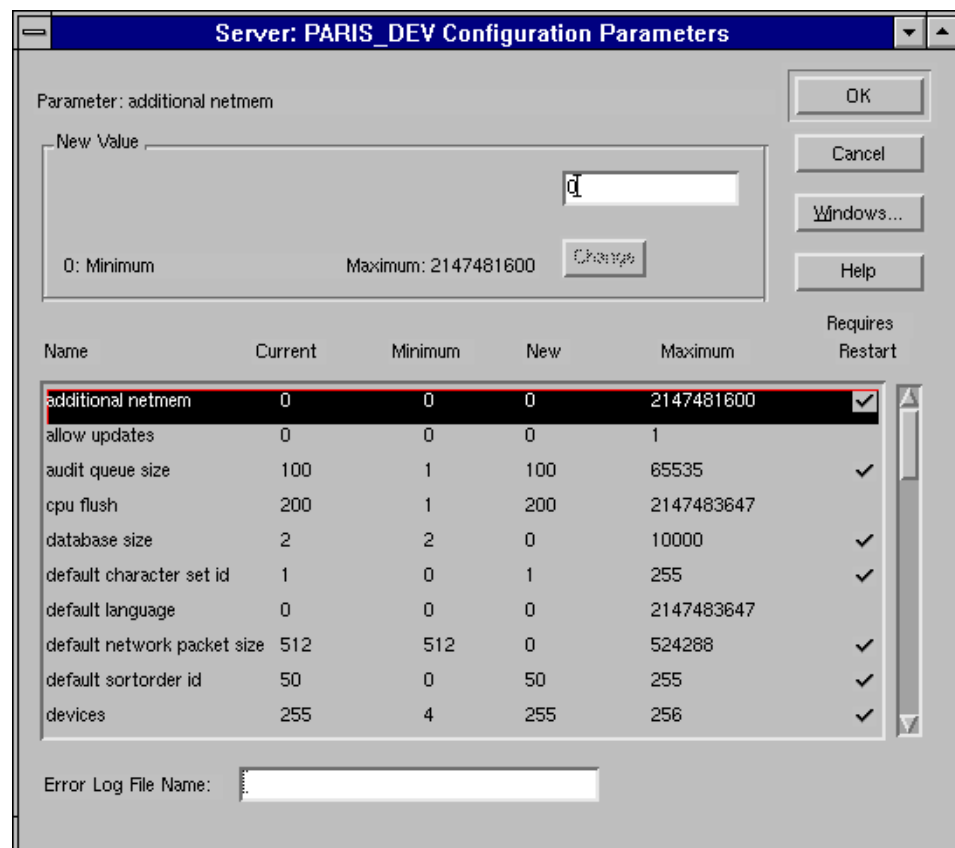


Figure 4-6: Configuration Parameters dialog box for SQL Server release 10.x

2. 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.
3. In the edit box, enter the new value of the configuration parameter.
4. Click the Change button. Enterprise 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.
5. Repeat steps 2 through 4 for each parameter you want to update.
6. Optionally, enter a pathname for the error log file.
7. Click OK. Enterprise 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, Enterprise SQL Server Manager copies the value in the New column to the Current column.
 - If the parameter you reset requires a SQL Server reboot, Enterprise 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 SQL Server Release 11.0

In SQL Server release 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 SQL Server release 11.0:

1. From the Server menu, choose Configuration. The Configuration Parameters dialog box opens.

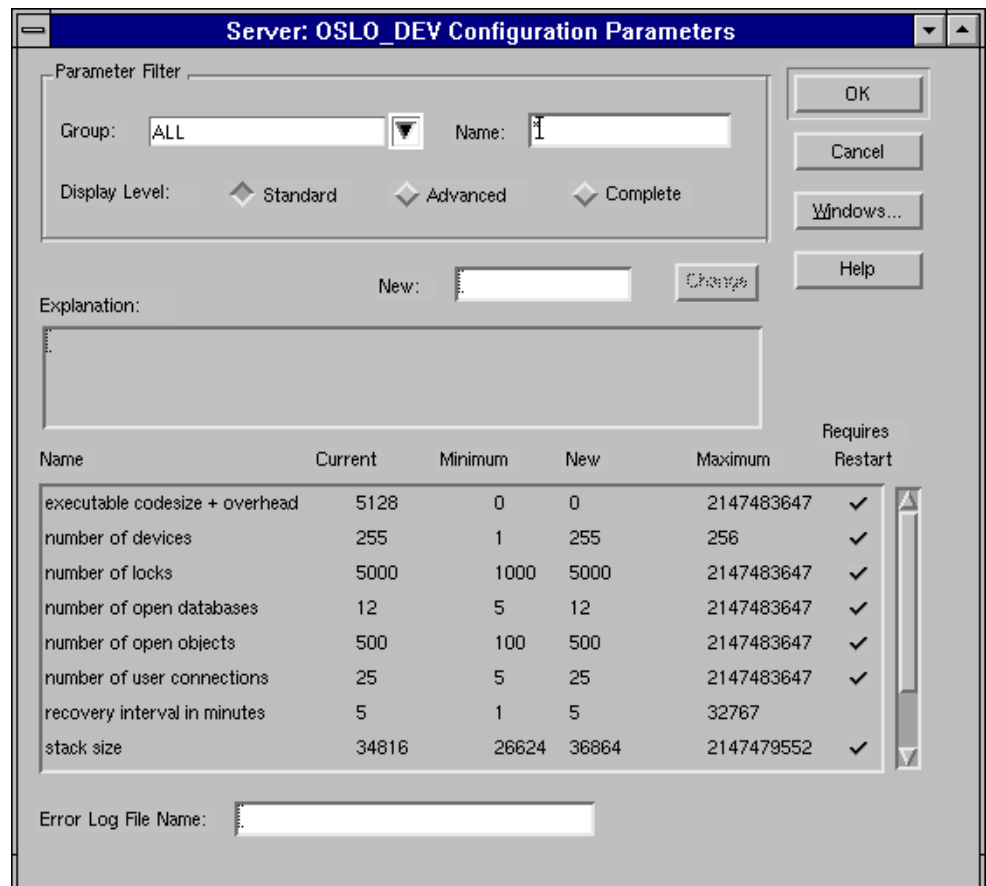


Figure 4-7: Configuration Parameters dialog box for SQL Server release 11.0

2. 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 wildcard characters “*” and “?” are valid. For example, to display parameters within the Group list and Display Level whose name begins with “p”, enter:

p*

3. 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.
4. In the New box, enter the new value of the parameter, and click Change.
5. Repeat steps 2 through 4 for each parameter you want to update.
6. Optionally, enter a pathname for the error log file.
7. Click OK. Enterprise SQL Server Manager updates the configuration values in the dialog box as follows:
 - If the parameter you reset takes effect immediately, Enterprise SQL Server Manager copies the value in the New column to the Current column.
 - If the parameter you reset requires a SQL Server reboot, Enterprise 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 Enterprise SQL Server Manager to view the status of user processes in SQL Server and to kill processes that are interfering with other users, blocking access to database objects, or consuming excessive system resources.

Displaying Processes

	TME	ESSM	SQL Server
Required roles	any	server	none

To display user processes in SQL Server, choose Processes from the Server menu. The Server Processes dialog box opens.

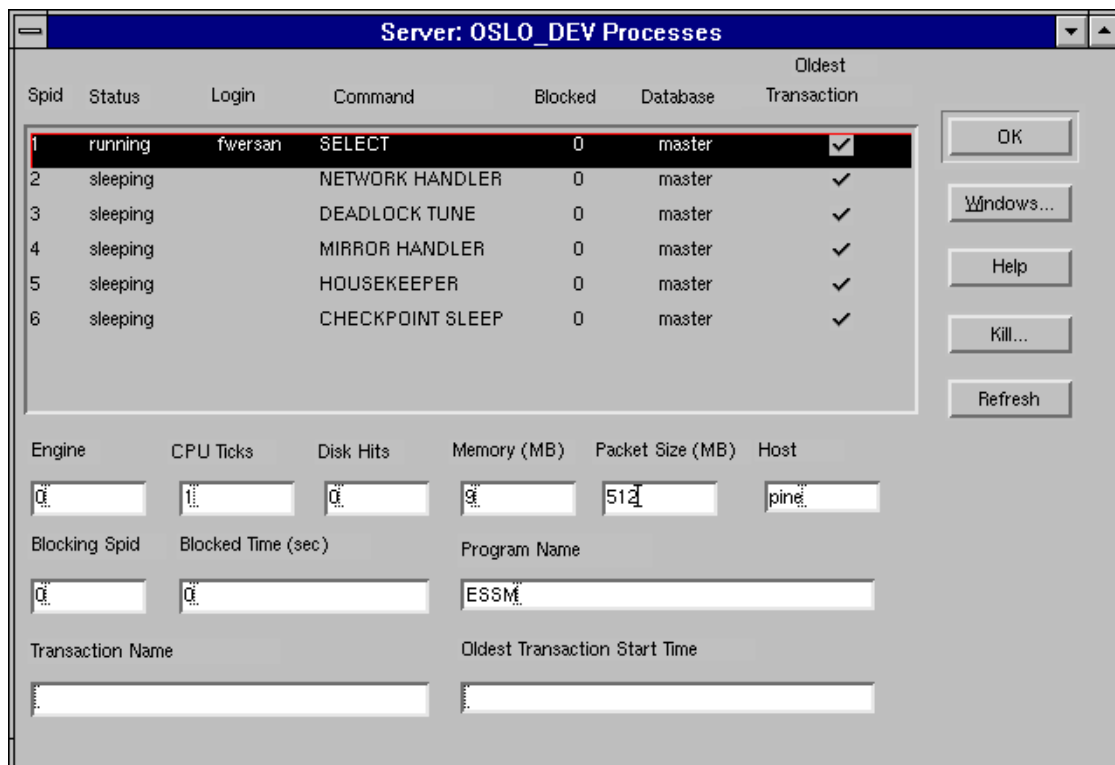


Figure 4-8: Server Processes dialog box

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 Release 11.0 Process Information

When you are managing SQL Server release 11.0, 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

	TME	ESSM	SQL Server
Required roles	any	server	System Administrator

To kill a process:

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

► **Note**

You cannot kill your own login process from a Voyager window. To kill your process, get another administrator to kill it, or use a tool external to Enterprise SQL Server Manager such as isql.

Displaying SQL Server Locks and Statistics

Enterprise SQL Server Manager does not report SQL Server locks and statistics. Use SQL Server Monitor to report this information.

Enabling Communication with Other Servers

Enterprise 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 Enterprise SQL Server Manager; for details, see *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, you must create a definition of the remote server in the local server.

	TME	ESSM	SQL Server
Required roles	any	security	System Security Officer

To create a remote server:

1. From the Server menu, choose Create; then, choose Remote Server. The Create Remote Server dialog box opens.

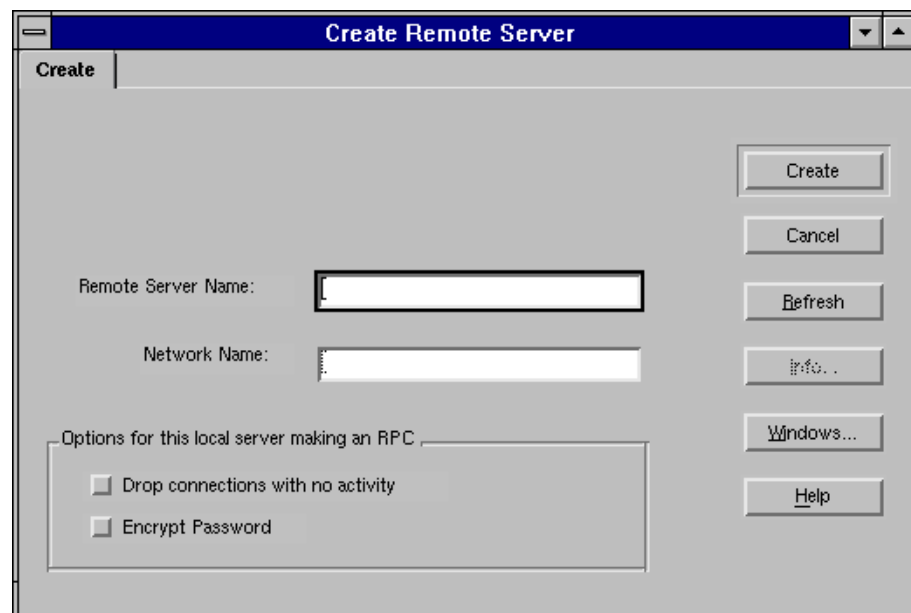


Figure 4-9: Create Remote Server dialog box

2. 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.
3. 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.

4. 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.
 - 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.
5. Click Create. Enterprise 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; then, select the Create toolbar button.



Click the right mouse button over the Remote Server container icon. 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

	TME	ESSM	SQL Server
Required roles	any	security	System Security Officer

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. Enterprise SQL Server Manager deletes the remote server and its login mappings.

Shortcuts



Select the remote server container icon; then, select the Delete toolbar button.



Click the right mouse button over the Remote Server container icon. 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.

Adding or deleting the remote logins of a remote server requires the following roles:

TME	ESSM	SQL Server
any	security	System Administrator

Changing the remote logins of a remote server requires the following roles:

TME	ESSM	SQL Server
any	security	System Security Officer and System Administrator

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 local SQL Server) 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.

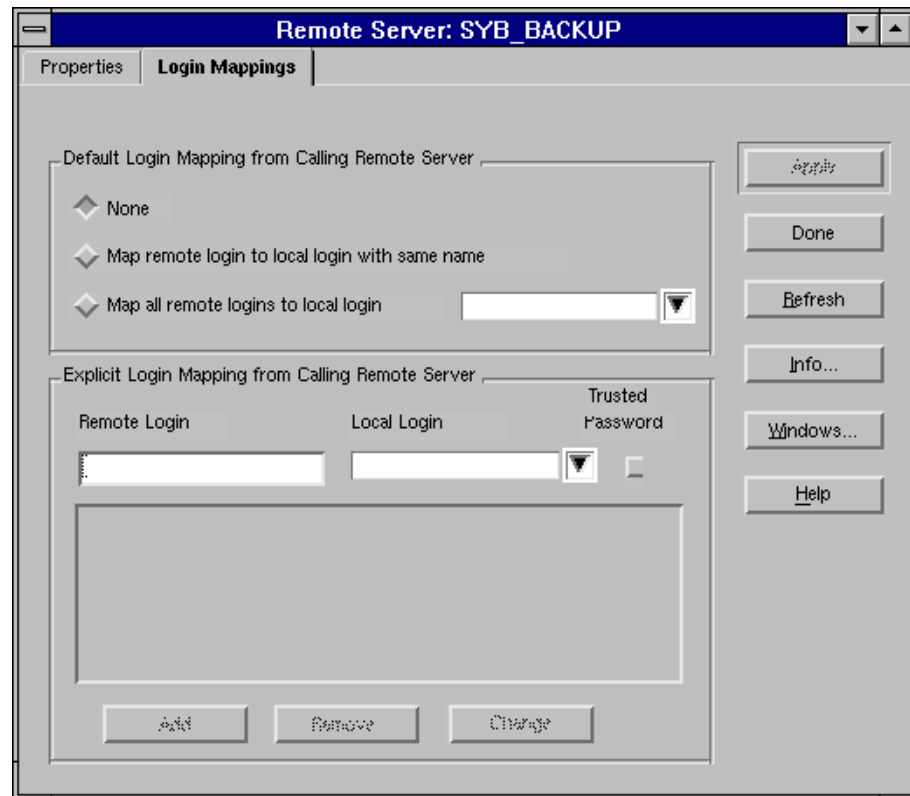





Figure 4-10: Remote Server Login Mappings tab

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. See “Setting Up Explicit Mappings” for this procedure. 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.
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.

Setting Up Explicit Mappings

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, choose 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.

Displaying Remote Server Details

Displaying the properties of a remote server requires the following roles:

TME	ESSM	SQL Server
any	any	none

Changing the properties of a remote server requires the following roles:

TME	ESSM	SQL Server
any	security	System Security Officer

To display information about an existing remote server:

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

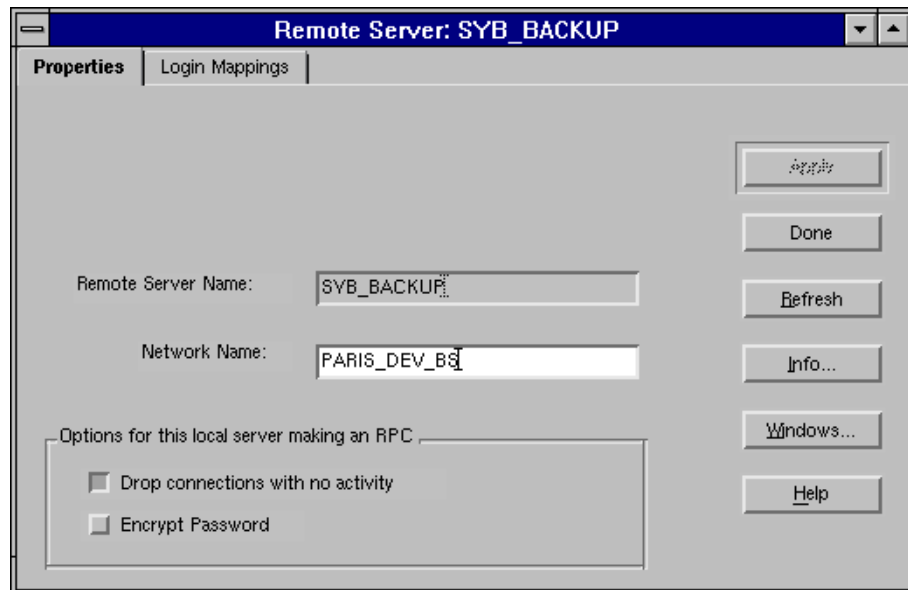


Figure 4-11: Remote Server Properties dialog box

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.

Enterprise 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, Enterprise 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.

You can create DDL in or execute DDL from a file system on a managed node. It does not have to reside on the client running the desktop. If the DDL is not on the client running the desktop, it must be accessible via NFS when it is executed.

Generating SQL Server DDL

	TME	ESSM	SQL Server
Required roles	any	dump	System Administrator

To generate SQL Server DDL:

1. Choose Generate DDL from the Server menu. The Server Generate DDL dialog box opens.

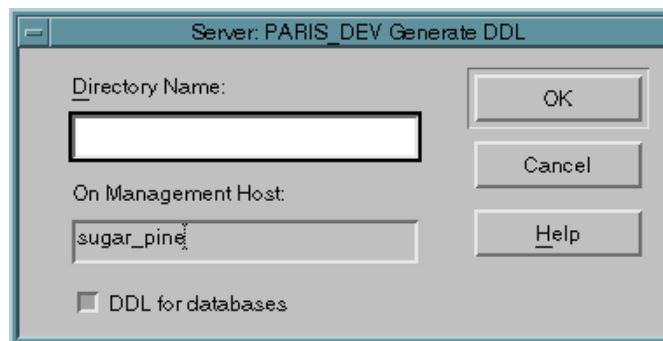


Figure 4-12: Server Generate DDL dialog box

2. In the Directory Name text box, enter the location in the directory hierarchy on the management host where you want Enterprise SQL Server Manager to create the top-level DDL script directory.
3. To generate DDL for all the databases on the Server, select DDL for Databases.
4. Click the OK button. Enterprise 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 4-35.

Executing SQL Server DDL

	TME	ESSM	SQL Server
Required roles	any	load	System Administrator

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. For information about modifying *.ddl* files, see the next section, “Examining and Editing DDL Script Files”.

To execute SQL Server DDL:

1. From the Server menu, choose Run DDL. The Server Run DDL dialog box opens.

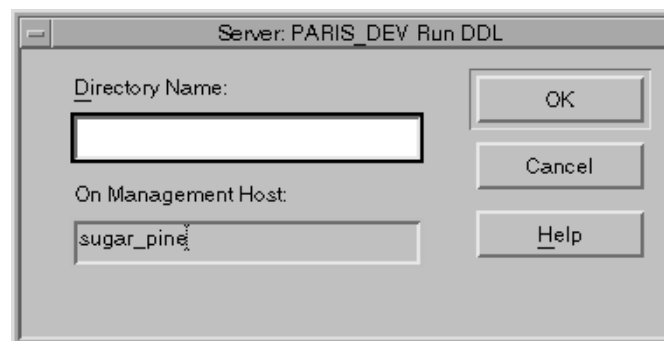


Figure 4-13: Server Run DDL dialog box

2. In the Directory Name box, enter the directory on the management host where the SQL Server DDL scripts are located.
3. Click the OK button. Enterprise SQL Server Manager executes the appropriate SQL Server DDL scripts. For a list of SQL Server DDL script files, see the following section, “Examining and Editing DDL Script Files”.

► **Note**

When you execute DDL, *cache.ddl* executes, but *cachebp.ddl* does not, because you can create buffer pools only on active caches and the caches created by *cache.ddl* are not active yet. To run *cachebp.dll*, stop SQL Server and restart it. Then use *isql* to run *cachebp.dll*.

Examining and Editing DDL Script Files

When Enterprise 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 Enterprise 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 Enterprise 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.

Table 4-1 lists the files in the *srv* subdirectory of the directory specified when the scripts were generated.

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



File name	What the file contains
<i>stats.dll</i>	Statistics about server configuration. Enterprise SQL Server Manager uses this information when attempting to execute DDL scripts on another server.
<i>exec.dll</i>	List of remaining server-level DDL files in correct execution order. At execution time, Enterprise SQL Server Manager uses this file to determine which scripts to execute and in what order.
<i>config.dll</i>	DDL to configure a server the way <i>srv</i> was configured when you generated DDL.
<i>devices.dll</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.dll</i>	DDL to create all databases that existed on <i>srv</i> , with database options set as on <i>srv</i> .
<i>logins.dll</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, Enterprise SQL Server Manager creates user accounts with password set to "password" when you execute DDL.
<i>remote.dll</i>	DDL to configure remote servers and remote logins.
 <i>cache.dll</i>	DDL to define named caches.
 <i>cachebp.dll</i>	DDL to define buffer pools and cache bindings for the named caches defined by <i>cache.dll</i>

Table 4-2 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. Enterprise 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> .

► **Note**

DDL generation does not create scripts for *syskeys* data inserted by *sp_primarykey* or *sp_foreignkey* or for user messages stored in the *sysusermessages* table.



Checking Consistency in SQL Server Release 11.0

When you are managing SQL Server release 11.0 , 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.”

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator or database owner

To run dbcc against a SQL Server:

1. From the Server menu choose Consistency. The Server Consistency Check dialog box opens.

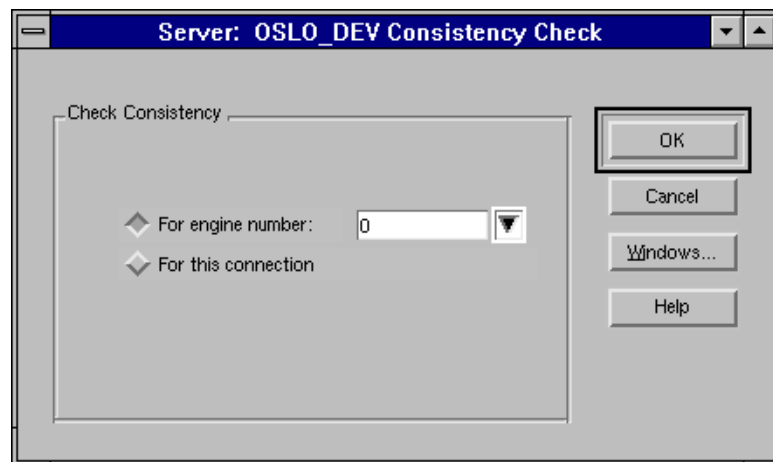
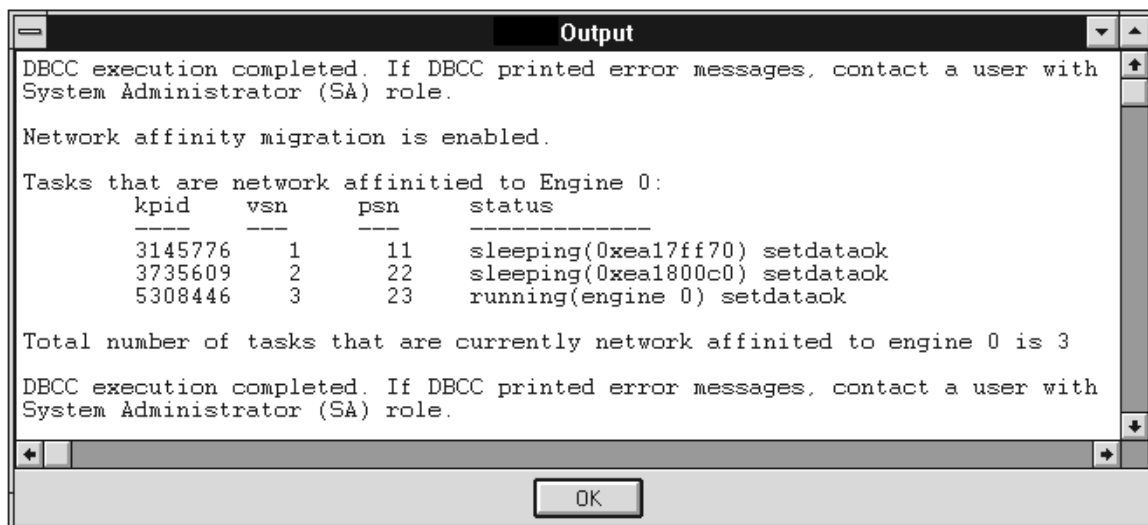


Figure 4-14: Server Consistency Check dialog box

2. 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.
3. Click OK. SQL Server executes the `dbcc` command, and Enterprise 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.



```

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

Network affinity migration is enabled.

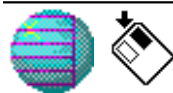
Tasks that are network affinitied to Engine 0:
      kpid      vsn      psn      status
      ----      -
      3145776    1        11      sleeping(0xea17ff70) setdataok
      3735609    2        22      sleeping(0xea1800c0) setdataok
      5308446    3        23      running(engine 0) setdataok

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.

Auditing SQL Server

To audit SQL Server, use Enterprise SQL Server Manager commands. For information about the auditing commands, see *Enterprise SQL Server Manager Reference Manual*.

5

Managing Cache



What's in This Chapter

When you are managing SQL Server release 11.0, 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 about:

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

Cache operations require the following roles:

TME	ESSM	SQL Server
any	cache	System Administrator

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.

Creating a Cache

To create a cache:

1. From the Server menu, choose Create. Then, choose Cache from the cascading menu. The Create Cache dialog box opens.

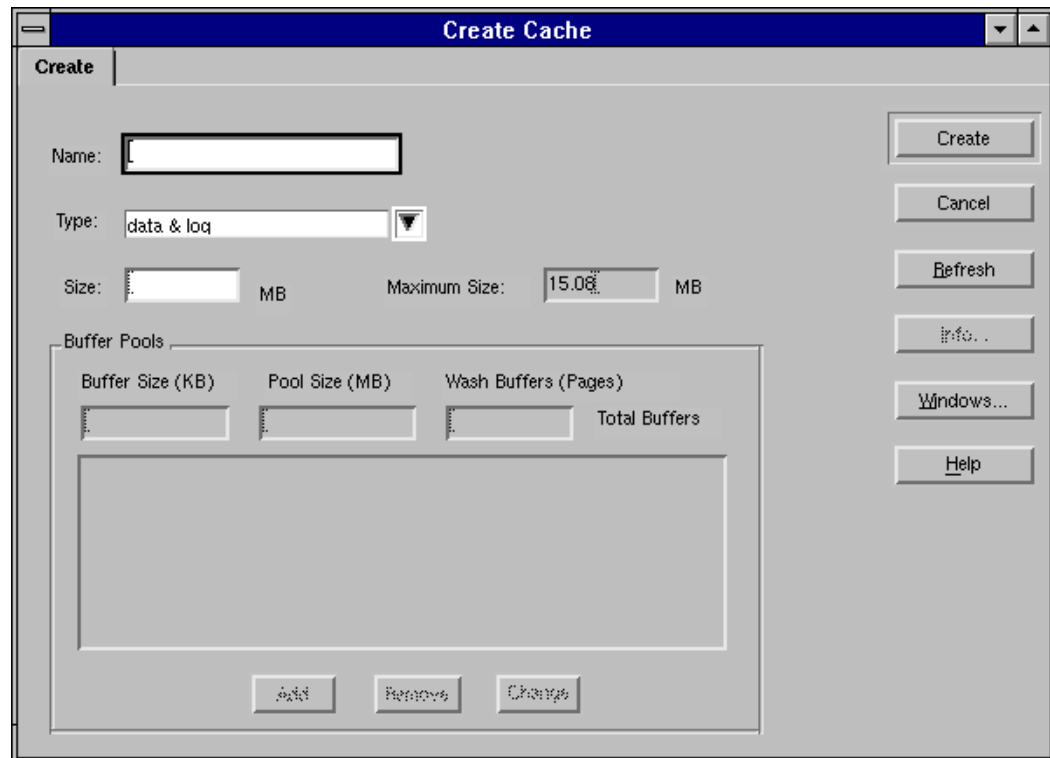


Figure 5-1: Create Cache dialog box

2. In the Name box, enter the name of the cache.
3. 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.”

4. 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.

5. 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 stop and restart SQL Server.

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

Shortcuts



Select the Cache container icon, 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:

1. Select the icon of the cache to delete.
2. From the Cache menu, choose Delete.
3. In the confirmation dialog box, click Yes. Enterprise SQL Server Manager 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.

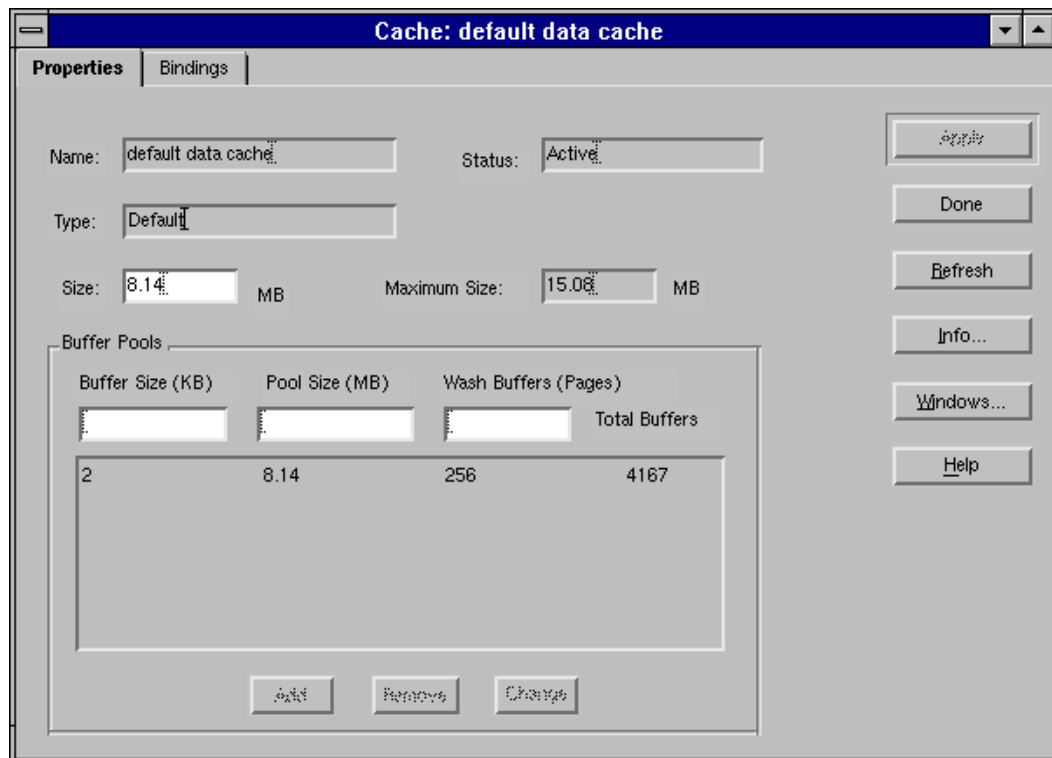
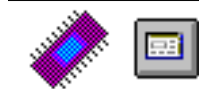


Figure 5-2: Cache Properties tab

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 5-7.
 - 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

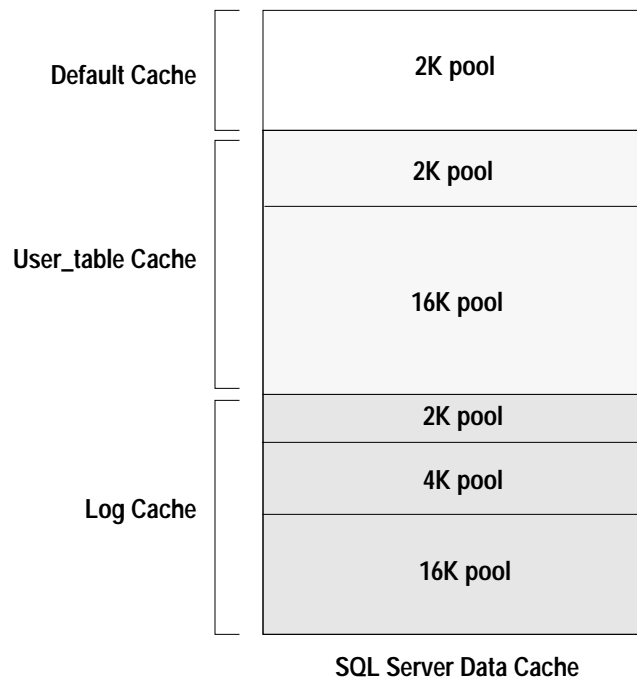


Figure 5-3: Default cache and two named data caches

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.

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.

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.

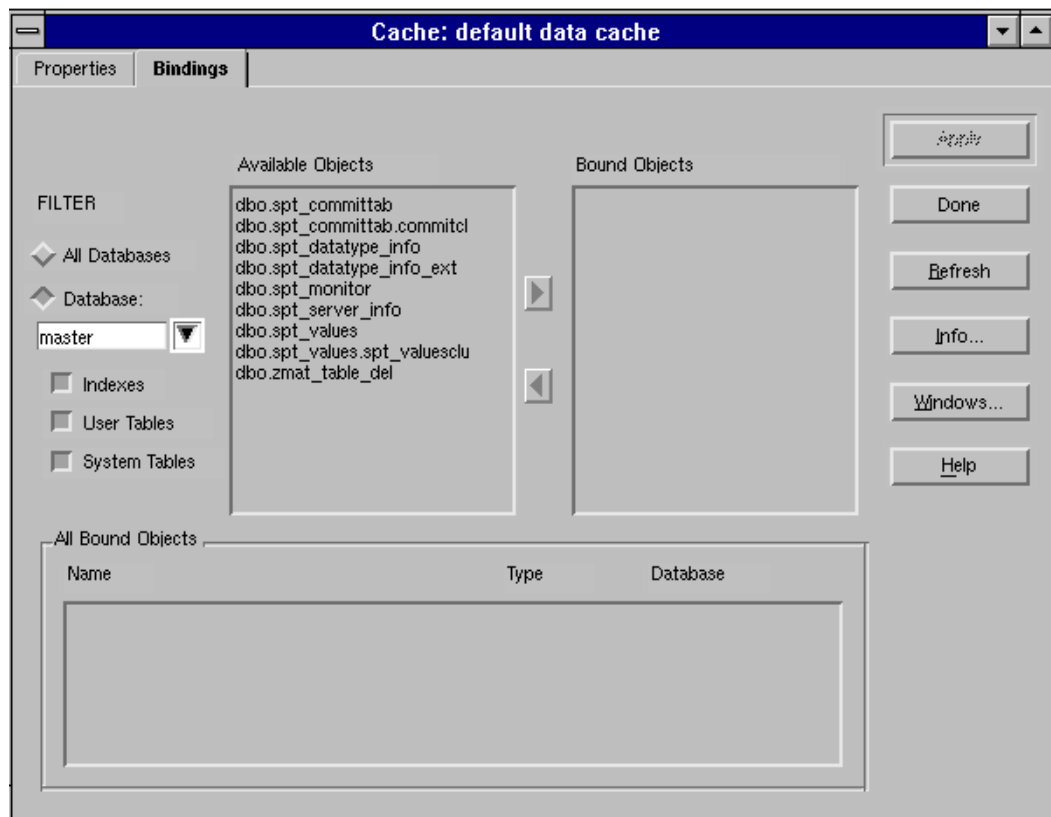
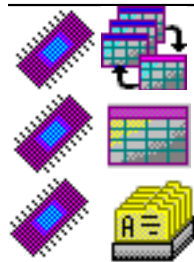


Figure 5-4: Cache Bindings tab

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.

Shortcuts



Drag and drop the cache icon onto the database, table, or index icon to display the Bindings tab.

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

Enterprise 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 Enterprise 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.

This chapter describes how to use Enterprise 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. Enterprise 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 6-12 and “Storage Management Considerations and Tips” on page 6-19.

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)

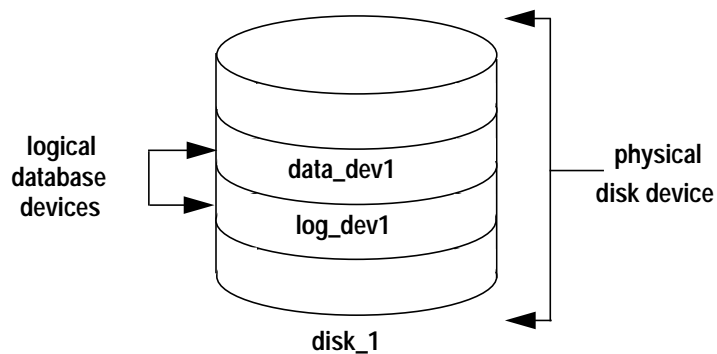


Figure 6-1: Logical database devices on a physical disk device

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 6-19 describes how.
- For ease of recovery, reserve the master device for the system databases, and create additional devices for your application databases.

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator

To create a database device:

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

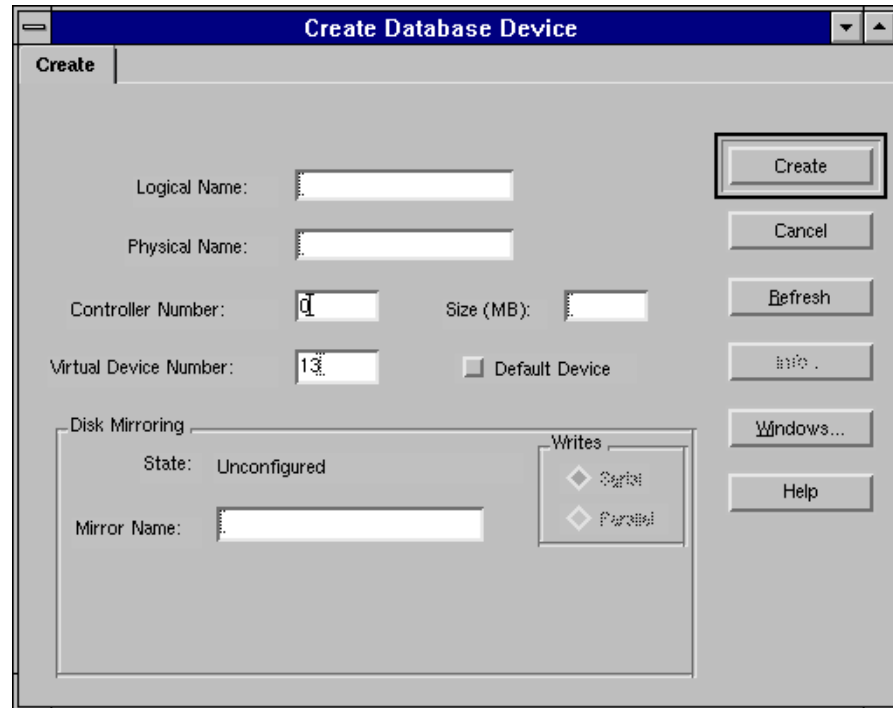


Figure 6-2: Create Database Device dialog box

2. 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, Enterprise 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.
3. 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 about default devices, see “Changing Default Device Designation” on page 6-19.
 4. 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 6-12. 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.
 5. Click Create. SQL Server creates the database device, and a new database device icon appears in Voyager.

Shortcuts



Select the database device container icon. Select the Create toolbar button.



Click the right mouse button over the database device container icon. 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

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator

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.

Notes:

- UNIX files associated with a database device are not deleted when the database device is deleted if the management host for SQL Server is not the same as the SQL Server host.
- 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 about the databases allocated on the device.

Examining Database Device Properties

Displaying database device properties requires the following roles:

TME	ESSM	SQL Server
any	any	none

Modifying database device properties requires the following roles:

TME	ESSM	SQL Server
any	space	System Administrator

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.

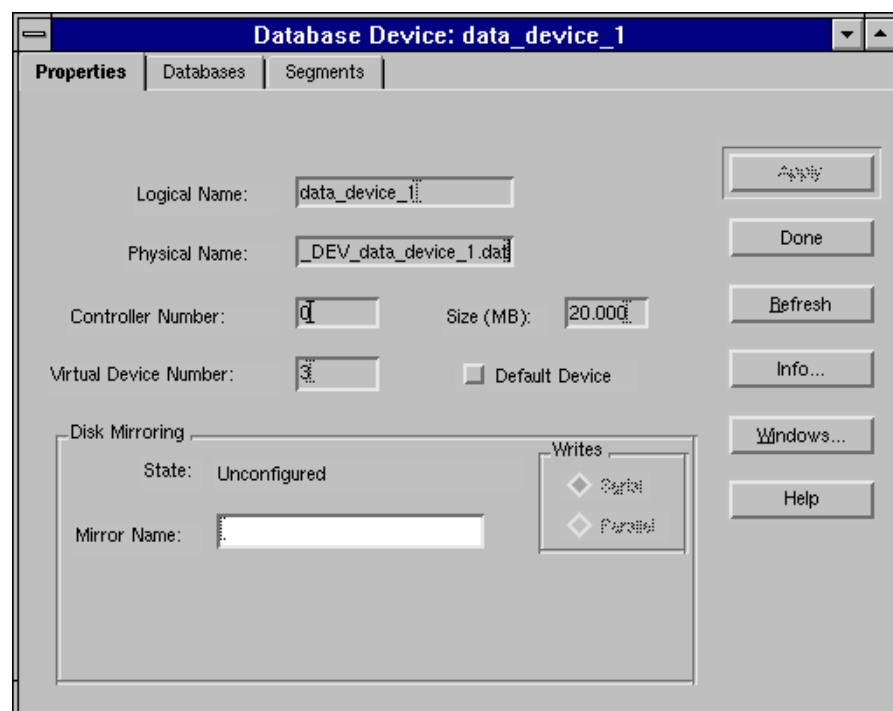

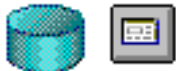
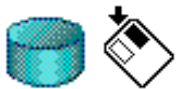


Figure 6-3: Properties tab of Database Device dialog box

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 6-7.
2. Click the Databases tab. The dialog box display changes to the Databases tab.

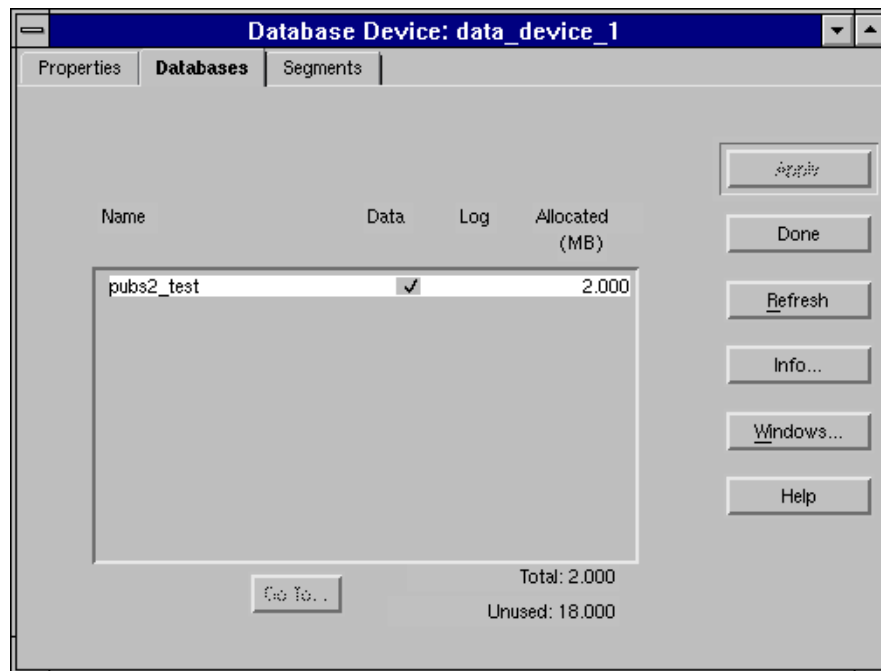


Figure 6-4: Databases tab of Database Device dialog box

The Databases group box shows:

- The name of each database allocated to the device
- What the database stores on the device: data, transaction log, or both data and 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 about database segments, see “Creating and Deleting Segments” on page 7-24.

To display the Segments tab:

1. Open the Database Device dialog box, as described in “Examining Database Device Properties” on page 6-7.
2. Select the Segments tab. The dialog box display changes to the Segments tab

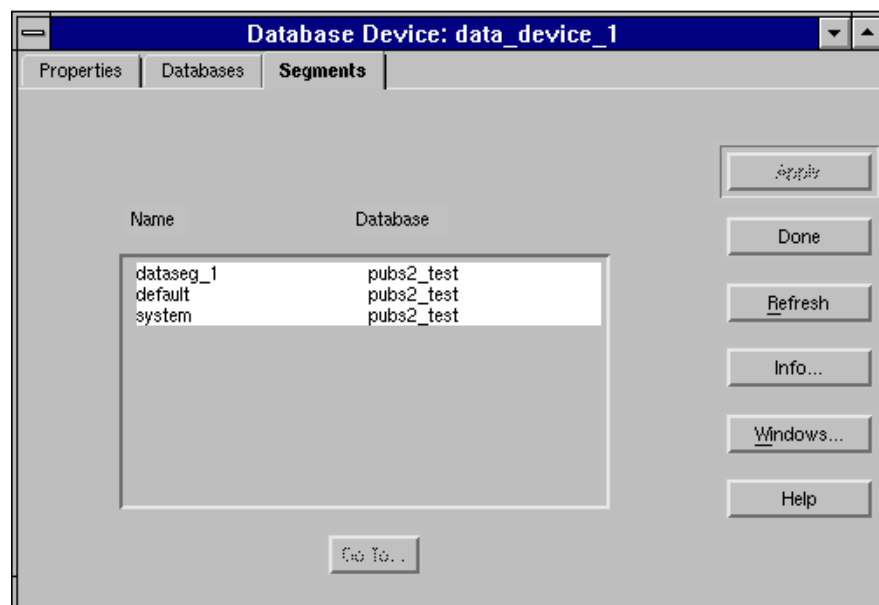


Figure 6-5: Segments tab of Database Device dialog box

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

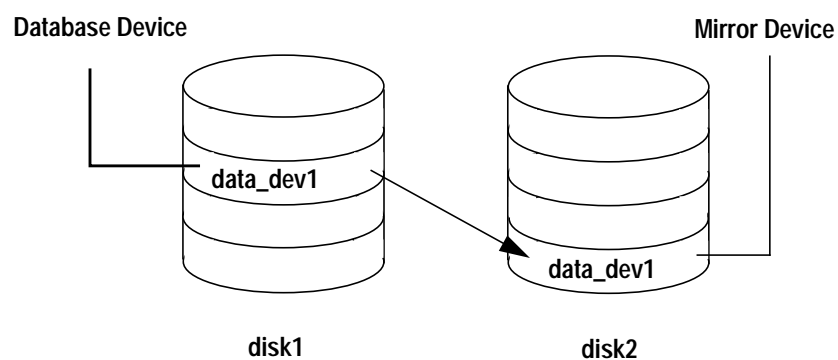


Figure 6-6: Database device on disk1 mirrored on disk2

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 6-3 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, as described in “Examining Database Device Properties” on page 6-7.
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.”

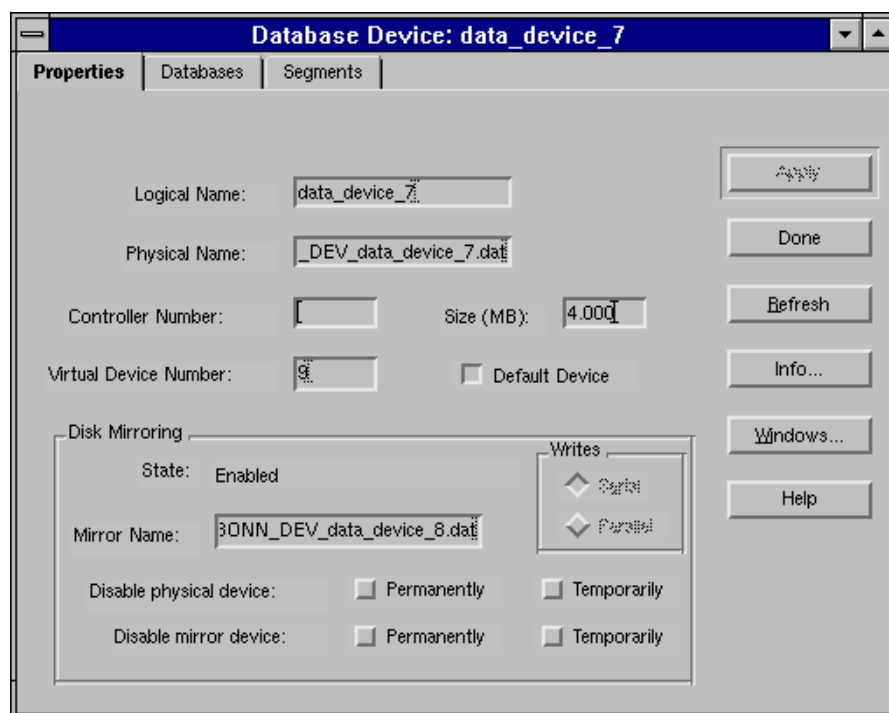
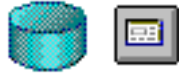


Figure 6-7: Database Device dialog box with disk mirroring enabled

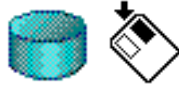
Shortcuts



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

Enterprise 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, as described in “Examining Database Device Properties” on page 6-7.
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

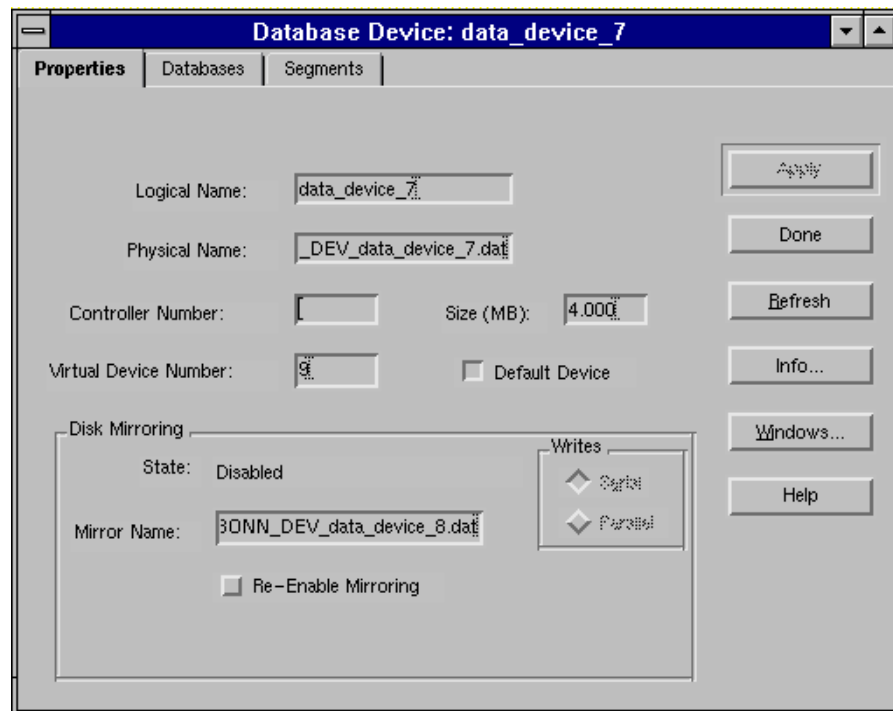
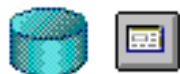


Figure 6-8: Database device with mirroring disabled

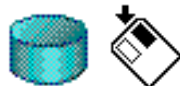
Shortcuts



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.

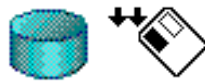
Re-Enabling Disk Mirroring

1. Open its Database Device dialog box, as described in “Examining Database Device Properties” on page 6-7.
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.

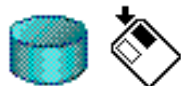
Shortcuts



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 6-7.
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, another method to protect data 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`:

1. On a nonproduction database, create the tables and indexes you anticipate the new database will contain. For information about creating tables and indexes, see Chapter 9, “Managing Database Objects.”
2. From the command line, run `sp_estspace` for each table, and add the results. For details on running `sp_estspace`, see *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.

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator

To create a database:

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

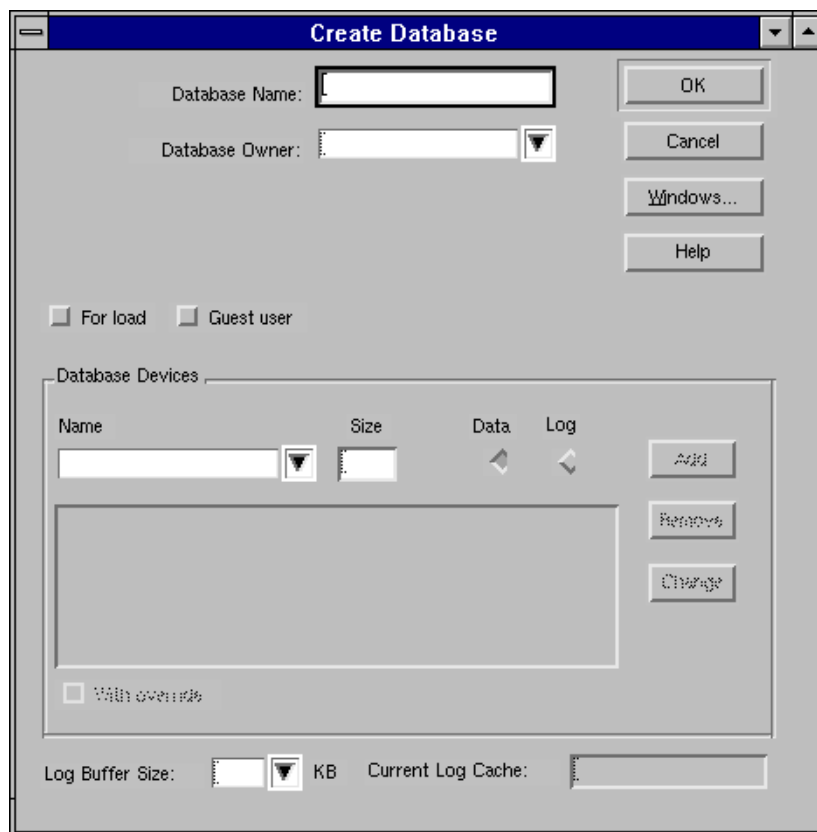


Figure 7-1: Create Database dialog box

2. 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.
3. 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.
4. To create a guest user in the database, check the Guest User check box.
5. 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.
6. 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:


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

To remove a database device allocation from the list:

- Select the device in the list of database devices.
- Click Remove.

► Note

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

7.  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.

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

Shortcuts

Select the database container icon; then, select the Create toolbar button.



Click the right mouse button over the database 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 Database.

Deleting Databases

This section describes how to delete a database, including 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.

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator or Database Owner

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

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator or Database Owner

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

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

	TME	ESSM	SQL Server
Required roles	any	any	System Administrator or valid database user role

To open the Database Properties dialog box:

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

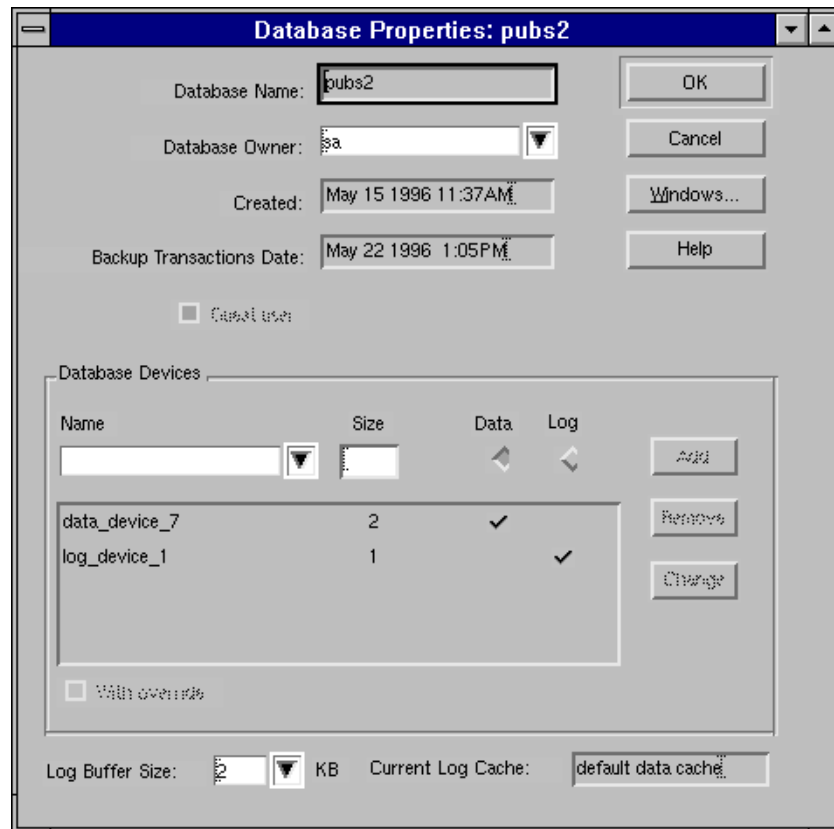




Figure 7-2: Database Properties dialog box


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.

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

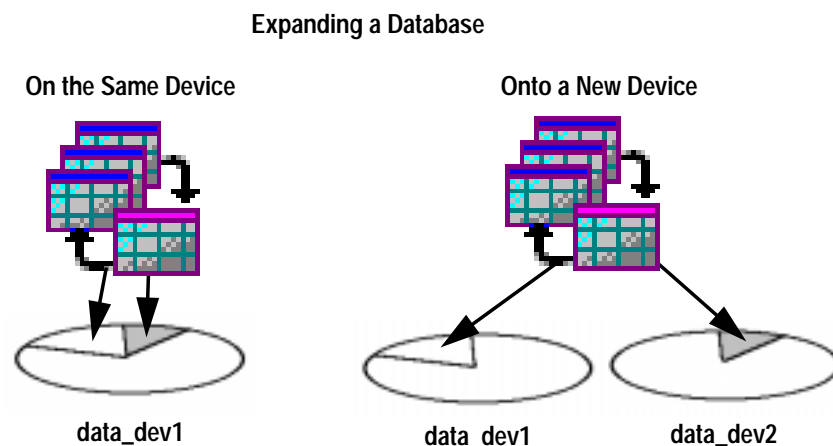


Figure 7-3: Expanding a database

► **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 expand a database by allocating space on the same device:

1. Open the Database Properties dialog box as described in “Examining Database Details” on page 7-7.
2. 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.

3. Click the Add button to move the data from the entry fields into the list box.
4. 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 expand a database by allocating space on a new device:

1. Open the Database Properties dialog box as described in “Examining Database Details” on page 7-7.
2. 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.
3. Click the Add button to move the data from the entry fields into the list box.
4. 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.

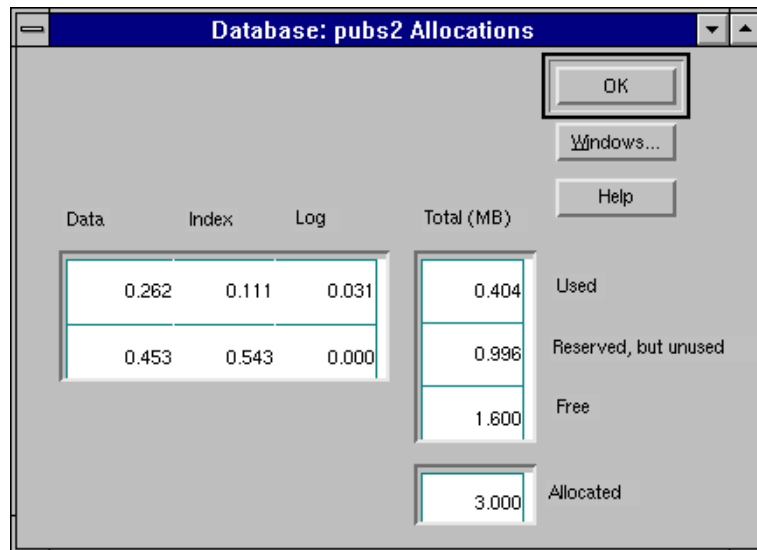
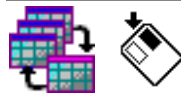


Figure 7-4: Database Allocations dialog box

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.

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

In SQL Server 11.0, 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.

Enterprise 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, Enterprise SQL Server Manager creates or executes the scripts for the database and its objects.

Location of DDL Script Files

By default, when you create DDL, it is generated on the management host for the SQL Server you are managing. When you execute DDL, it is executed on the host from which you started the Tivoli desktop.

To allow access to DDL from all managed nodes, generate DDL in directories that are NFS-mounted and accessible to all hosts.

Generating Database DDL

	TME	ESSM	SQL Server
Required roles	any	dump	System Administrator

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.

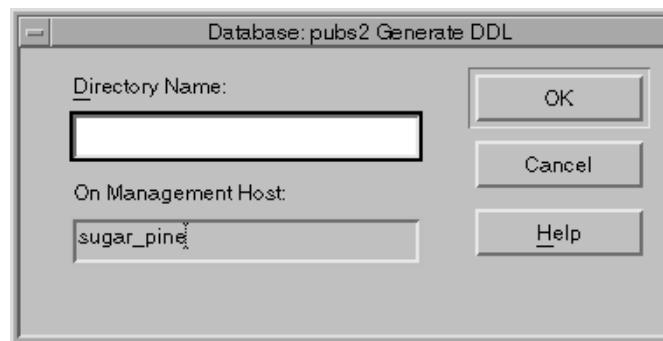


Figure 7-5: Database Generate DDL dialog box

3. In the Directory Name box, enter the location in the directory hierarchy on the management host where you want Enterprise SQL Server Manager to create the top-level DDL script directory.
4. Click the OK button. Enterprise 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

	TME	ESSM	SQL Server
Required roles	any	load	System Administrator

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.

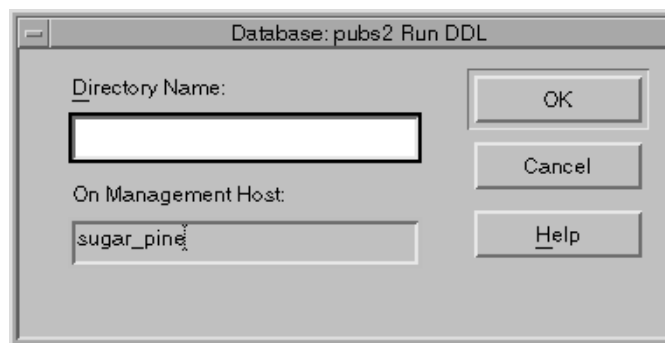
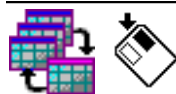


Figure 7-6: Database Run DDL dialog box

3. In the Directory Name box, enter the directory (on the management host) named for the database for which you generated DDL scripts. This directory is a subdirectory in the directory named for SQL Server.
4. Click OK. Enterprise 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 Enterprise 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 Enterprise 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.

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. Enterprise 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> .
<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> .

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

File Name	What the File Contains
<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> .

► Note

DDL generation does not create scripts for *syskeys* data inserted by *sp_primarykey* or *sp_foreignkey* or for user messages stored in the *sysusermessages* table.

Changing Database Options

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

	TME	ESSM	SQL Server
Required roles	any	none	System Administrator

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*.

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.

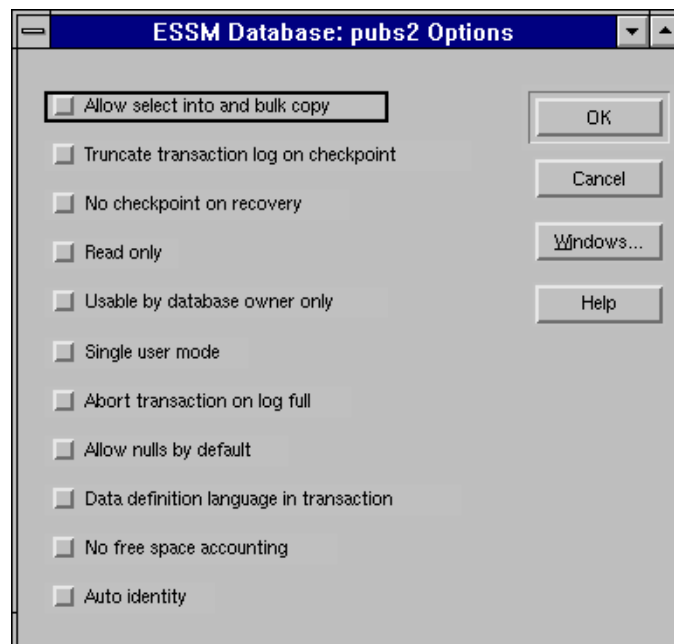
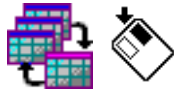


Figure 7-7: Database Options dialog box

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.
- **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 managing 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

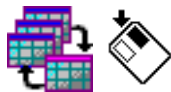
Through this mechanism, SQL Server regularly synchronizes the database and its transaction log, thereby shortening the recovery time in the event of a system failure.

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.

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.

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator

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.

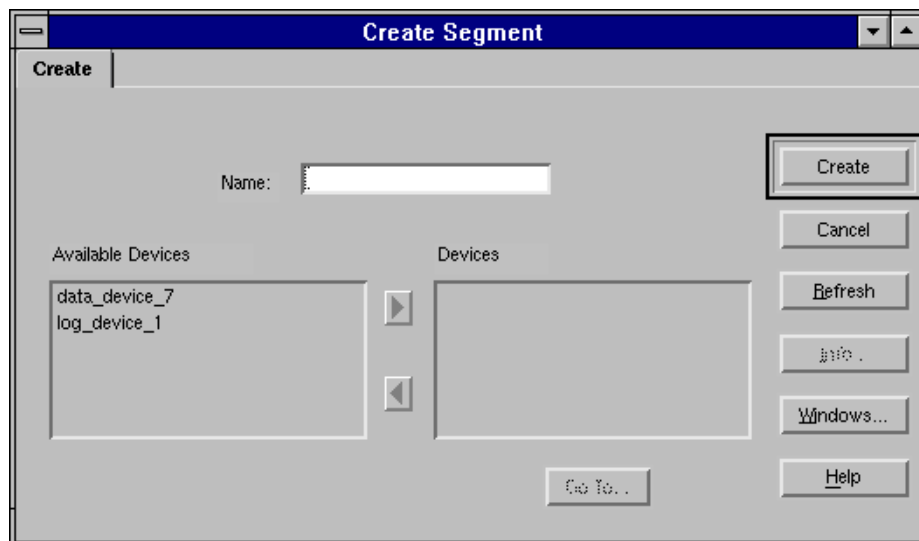


Figure 7-8: Create Segment dialog box

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.

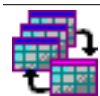
Shortcuts



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.

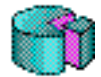
Deleting a Segment

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator

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

Enterprise 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.

Displaying segment properties requires the following roles:

TME	ESSM	SQL Server
any	any	System Security Officer, or valid user in database. Displaying the threshold option requires System Administrator.

Modifying segment properties requires the following roles:

TME	ESSM	SQL Server
any	space	System Administrator

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.

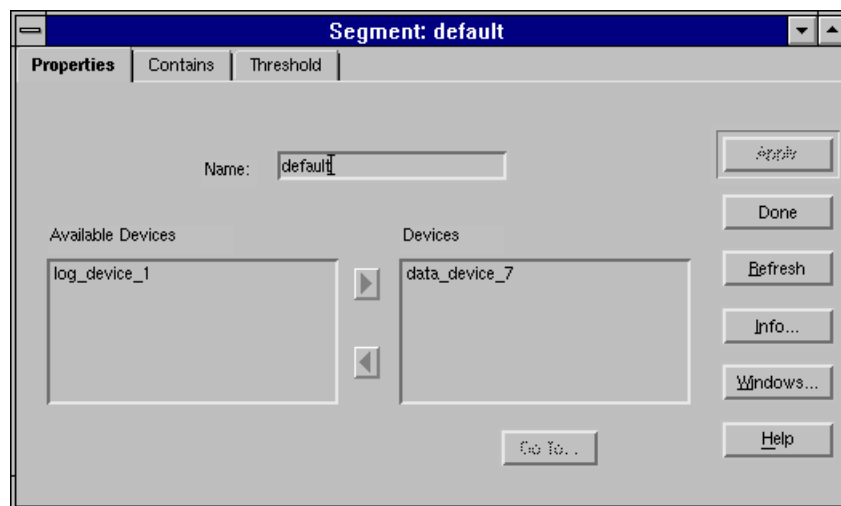


Figure 7-9: Properties tab of Segment dialog box

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 7-28.
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 a 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 7-28.
2. Click the Contains button to display the Contains tab.

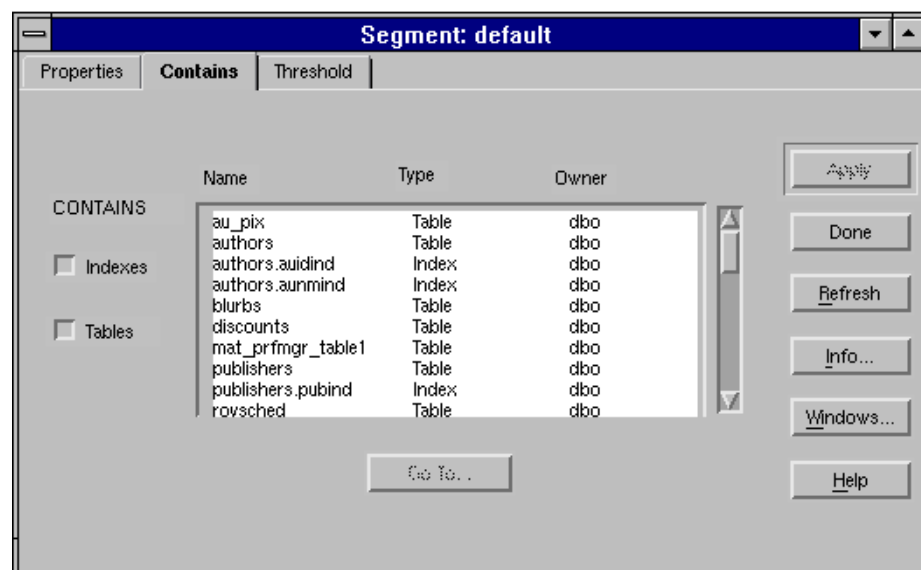


Figure 7-10: Contains tab of the Segment dialog box

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 Enterprise 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.

Creating, deleting, or changing a threshold requires the following roles:

TME	ESSM	SQL Server
any	space	System Administrator

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 Tolerance (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 7-28.
3. Click the Thresholds control to display the Thresholds tab

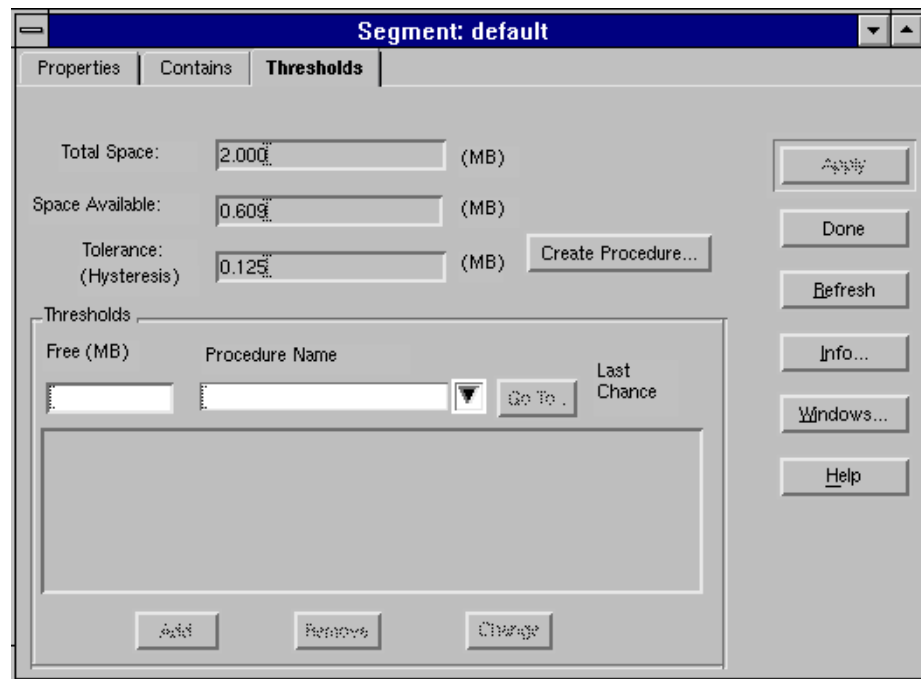


Figure 7-11: Thresholds tab of Segment dialog

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 9-31.

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 7-32.
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 7-32.
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.

Starting Backup Server

	TME	ESSM	SQL Server
Required roles	none	server	System Administrator

To start Backup Server:

1. In a policy region window, click the right mouse button over the icon of the SQL Server for which you want to start Backup Server. From the popup menu, choose Start. The Start SQL Server dialog box opens.
2. In the Start group box, select Backup Server.
3. In the Runserver File Name box, type the name of the runserver file associated with the Backup Server you want to start.
4. Click OK.

Stopping Backup Server

	TME	ESSM	SQL Server
Required roles	none	server	System Administrator

To stop Backup Server:

1. In a policy region window, click the right mouse button over the icon of the SQL Server for which you want to stop Backup Server. From the popup menu, choose Stop. The Stop SQL Server dialog box opens.
2. In the Stop group box, select Backup Server.
3. In the Backup Server text box, type the name of the Backup Server you want to stop. The default is the last Backup Server you started.
4. In the Shutdown Mode group box, select one of the options:
 - Stop Immediately**—stops the server immediately.
 - Wait for processes to end**—specifies that shutdown is to wait for existing processes to exit before stopping the server.
5. Click OK.

► **Note**

If you accept the default for the Backup Server name, remember that there can be more than one Backup Server per SQL Server, so you must be sure that the Backup Server you are stopping is the most recently started Backup Server.

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

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator

To create a dump device:

1. From the Server menu, choose Create; then, choose Dump Device. The Create Dump Device dialog box opens.
2. 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.

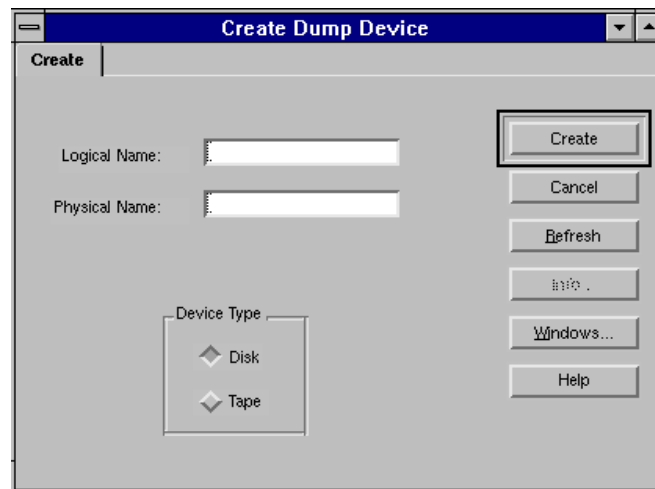


Figure 7-12: Create Dump Device dialog box

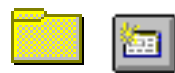
3. Enter the following information:

For Logical Name, enter your name for the device.

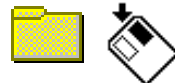
For Physical Name, enter the name of the physical device in your environment to map the logical name to. For a **tape dump device**, it must be the name of a tape drive. For a **disk dump device**, it must be the name of an operating system file, and you must specify the full path name.

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.

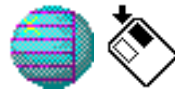
Shortcuts



Click the Dump Device container icon; then, select the Create toolbar button.



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



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

Deleting a Dump Device

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator

To 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.

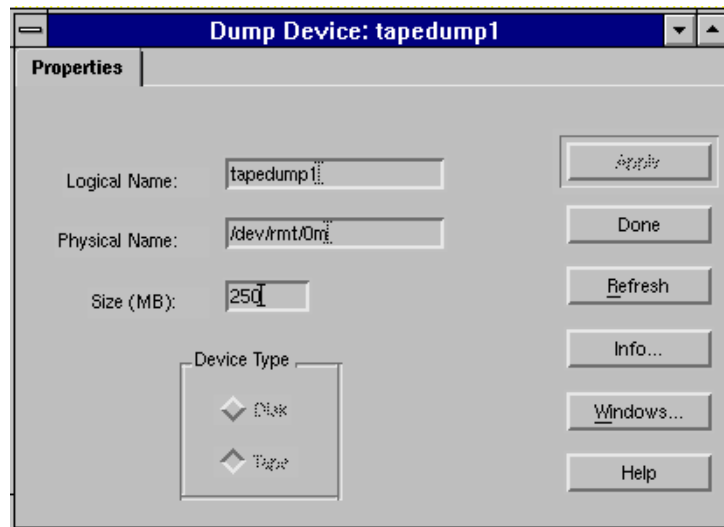


Figure 7-13: Properties tab of the Dump Device dialog box

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

Enterprise 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.

Using the Tivoli Scheduler to Schedule Backups

To use the Tivoli Scheduler to schedule backup and restore tasks, you must write a script and make the script a Tivoli task. Then you can schedule the task. The Tivoli Scheduler is not accessible from the Enterprise SQL Server Manager Database Backup dialog box or the Database Restore dialog box. For information about writing scripts, see *Enterprise SQL Server Manager Reference Manual*.

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

	TME	ESSM	SQL Server
Required roles	admin, senior, or super	dump	Operator

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.

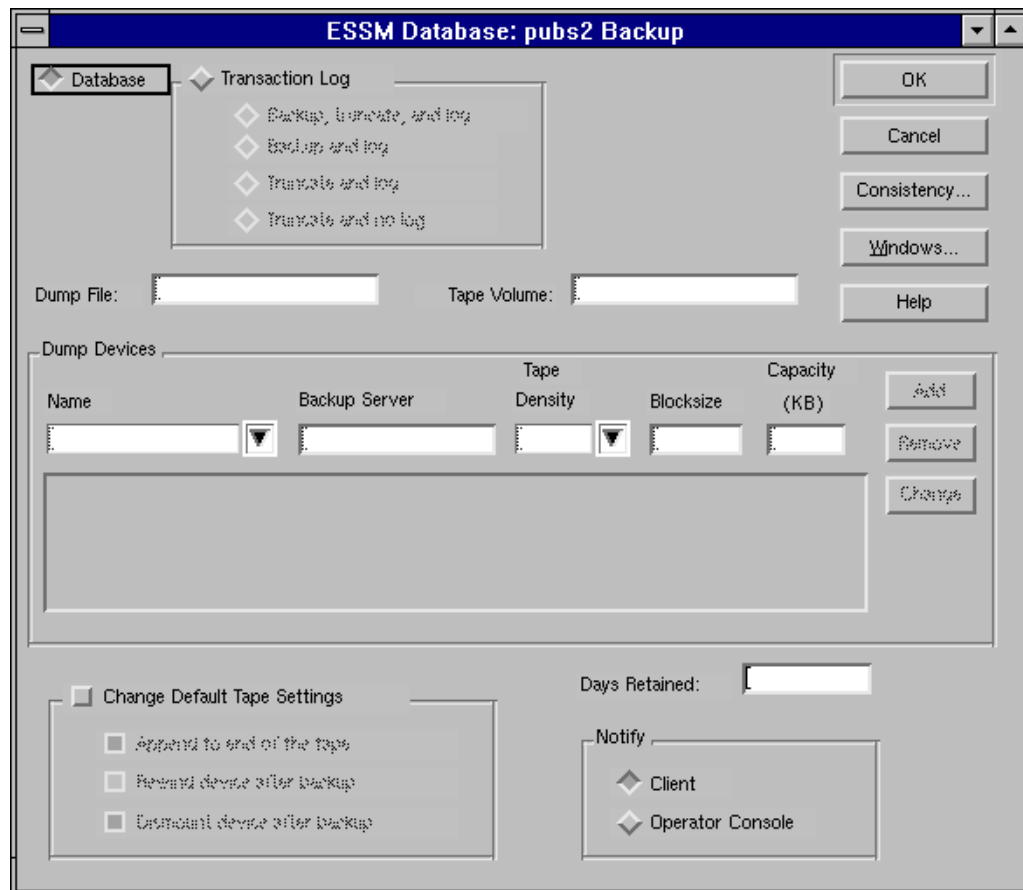


Figure 7-14: Database Backup dialog box

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.

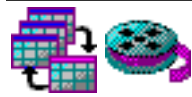
To add a set of dump device specifications to the Database Devices list, click Add. To change an entry, highlight it, make changes, and click Change. To delete an entry, highlight it and click Remove.

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.

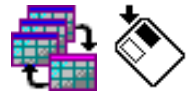
When you select Change Default Tape Settings, the following options become available. Indicate the options you want by selecting or clearing the corresponding check boxes:

- 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 Enterprise 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

	TME	ESSM	SQL Server
Required roles	any	dump and load	Operator and System Administrator

If a backup operation requires intervention, a message is sent to the Sybase Backup/Recovery Notice Group. Use the Volume Change dialog box to respond to requests for volume changes.

To proceed with the backup :

1. Mount a new tape volume.
2. From the Databases menu, choose Change Volume. The Volume Change dialog box opens.

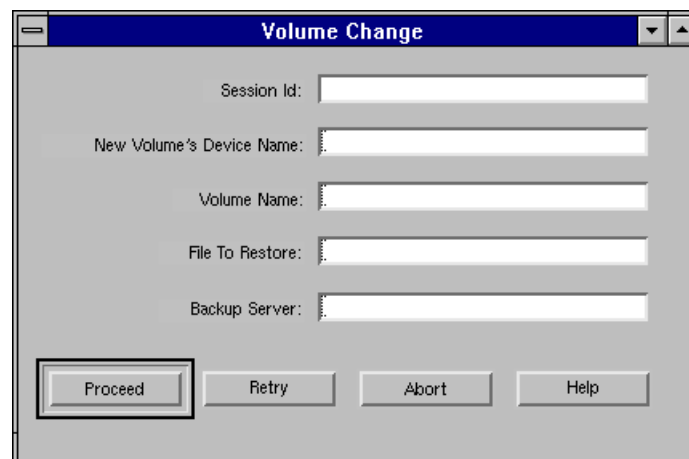


Figure 7-15: Volume Change dialog box

3. Using the information in the message that was sent to the Notice Group, fill in the dialog box.
 - Enter the session id.
 - If the new volume is mounted on a device other than the original one specified, enter the new device name in the New Volume's Device Name box.
 - Optionally, enter the name of the new tape volume in the Volume Name box.
 - Optionally, enter the name of the Backup Server.
 4. Click Proceed.
- To stop the backup, click Abort.
- To retry the backup, click Retry.

Restoring a Database

Enterprise 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 *System Administration Guide* and *Sybase Troubleshooting Guide*.

Using the Database Restore Dialog Box

	TME	ESSM	SQL Server
Required roles	any	load	System Administrator

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.

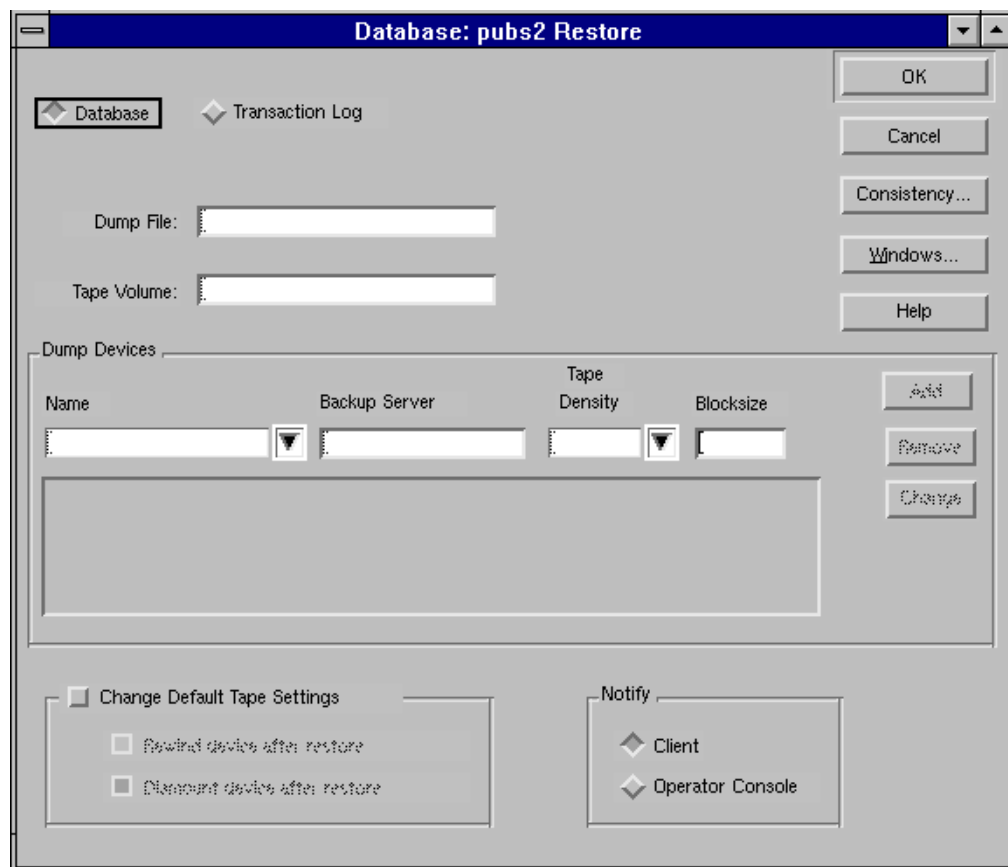


Figure 7-16: Database Restore dialog box

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.

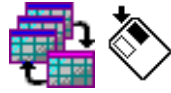
7. Optionally, if the dump device is a tape, select Change Default Tape Settings to indicate that you want to change the way SQL Server handles the tape restore by default. If you leave this box unselected, SQL Server dismounts the tape device when the restore is complete and does not rewind the device.

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 Enterprise 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

	TME	ESSM	SQL Server
Required roles	any	dump and load	Operator and System Administrator

If a restore operation requires intervention, a message is sent to the Sybase Backup/Recovery Notice Group. Use the Volume Change dialog box to respond to requests for volume changes.

To proceed with the restore:

1. Mount a new tape volume.
2. From the Databases menu, choose Change Volume. The Volume Change dialog box opens.

Figure 7-17: Volume Change dialog box

3. Using the information in the message that was sent to the Notice Group, fill in the dialog box.
 - Enter the session id.
 - If the new volume is mounted on a device other than the original one specified, enter the new device name in the New Volume's Device Name box.
 - Optionally, enter the name of the new tape volume in the Volume Name box.
 - If you entered a name in the Volume Name box, enter the name of the file to restore on that volume in the File to Restore box.

- Optionally, enter the name of the Backup Server.

4. Click Proceed.

To stop the restore, click Abort.

To retry the restore, click Retry.

To Get Information About Backup Server Messages

Enterprise 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*.

Checking Database Consistency

Enterprise 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."

	TME	ESSM	SQL Server
Required roles	any	space	System Administrator or Database Owner

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.

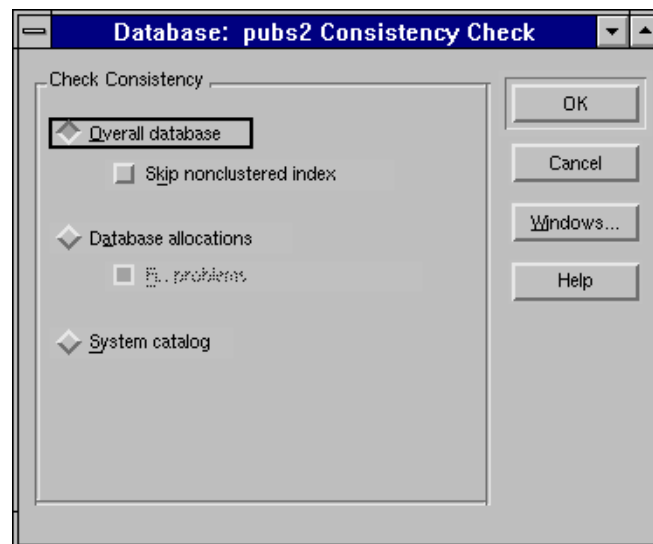
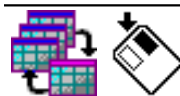


Figure 7-18: Database Consistency Check dialog box

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 7-57.
4. Click OK. SQL Server executes the `dbcc` command. When it completes, Enterprise 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

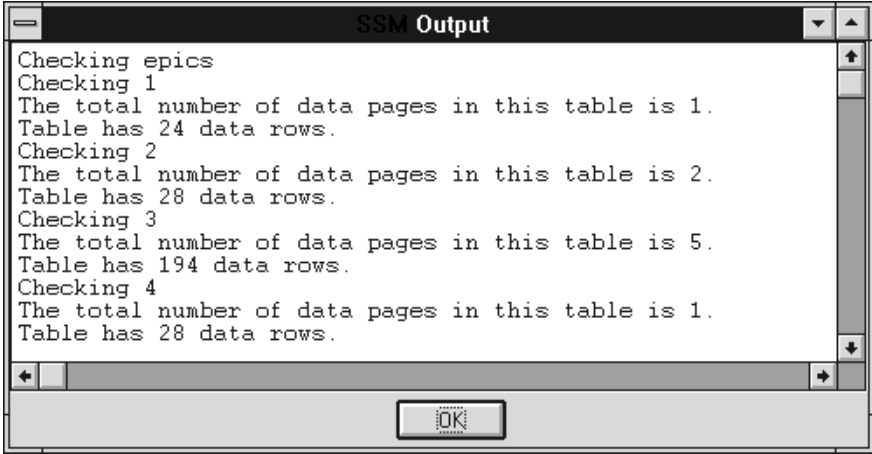
Overall Database Option

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 about 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 7-19: **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. Enterprise 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.

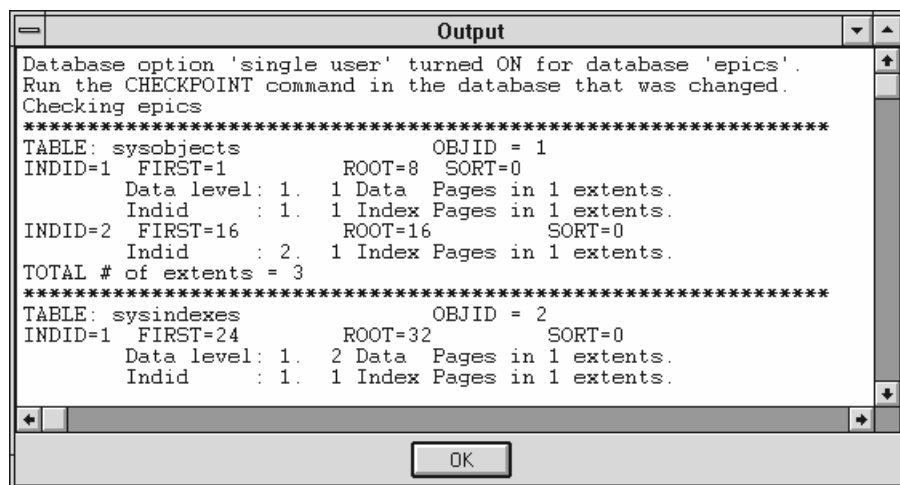
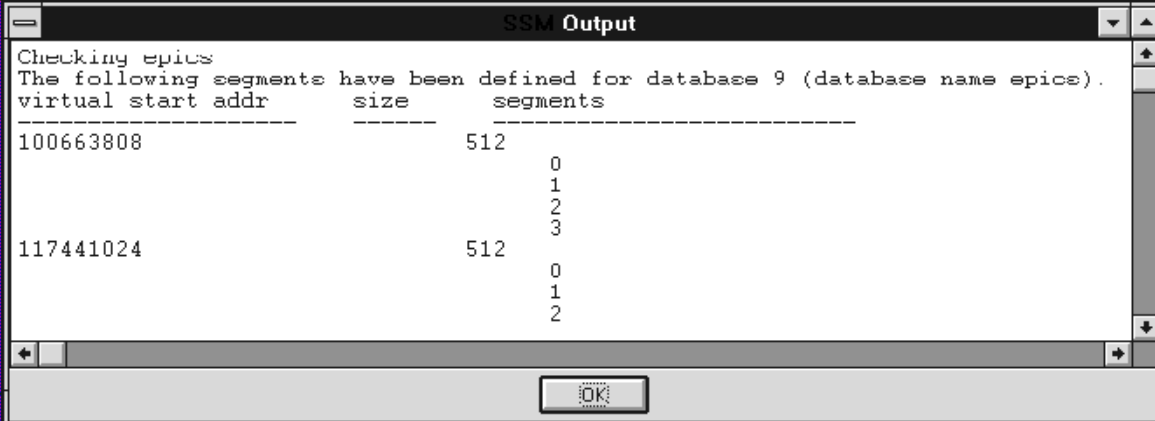


Figure 7-20: `dbcc checkalloc` output

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.



```
SSM Output
Checking epics
The following segments have been defined for database 9 (database name epics).
virtual start addr      size      segments
-----
100663808                512
                        0
                        1
                        2
                        3
117441024                512
                        0
                        1
                        2
```

Figure 7-21: dbcc checkcatalog output

 **Setting a Database Online**

Restoring a database or transaction log requires that the database in which it is performed be set off-line. When managing SQL Server 11.0, Enterprise SQL Server Manager automatically sets the database off-line. 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 Enterprise SQL Server Manager can leave a database in the off-line state. If you receive an error message in Enterprise 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:

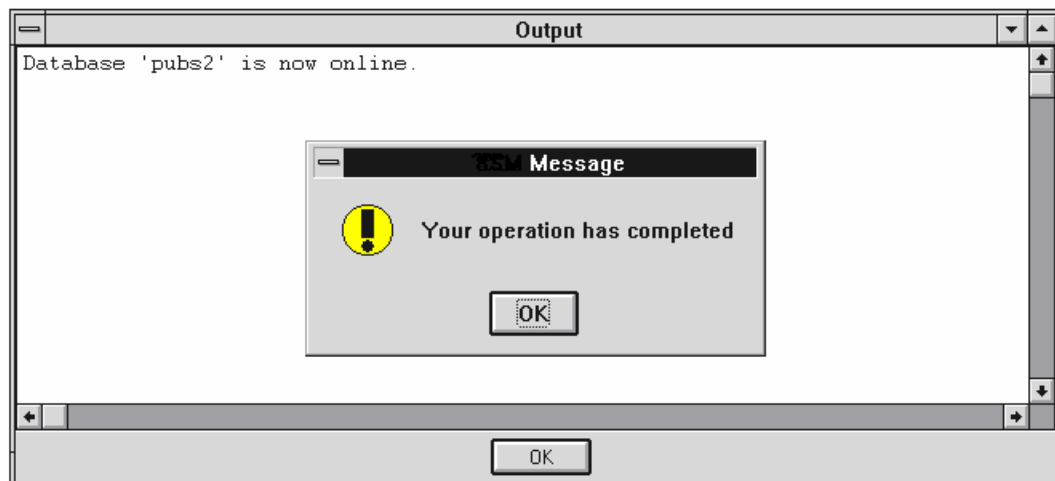


Figure 7-22: Confirmation message: setting a database online

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, an Enterprise SQL Server Manager user must have a SQL Server login. For access to a database or database objects, an Enterprise 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 Enterprise 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 8-43 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 Enterprise 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.

► *Note*

Dialog boxes in Enterprise SQL Server Manager do not display permissions granted as a result of role, group membership, or special user status (implicit permissions). The command and object permission tabs display only those permissions that you set explicitly.

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 Enterprise 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 Enterprise 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 8-9 and “Modifying Login Information” on page 8-12.

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 8-6. 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 Enterprise 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 8-29.
- 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 8-41.

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 Enterprise SQL Server Manager, this form of revoking permissions is called revoking **with cascade**.

You can use Enterprise 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?

Y
N: No Access

Y

Is Homer associated with a login that has the System Administrator role?

Y: Access

N

Is Homer the owner of Epics?

Y: Access

N

Is Homer a member of a group that has been granted permission to create a table in Epics?

Y: Access

N

Has Homer been granted permission to create a table in Epics?

N: No Access

Y: Access

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 Enterprise SQL Server Manager user access to SQL Server. This section describes how to create and delete a login.

Creating a SQL Server Login

	TME	ESSM	SQL Server
Required roles	any	security	System Security Officer

To create a SQL Server login:

1. Choose Login from the Create menu that cascades from the Server menu. The Create Login dialog box opens:

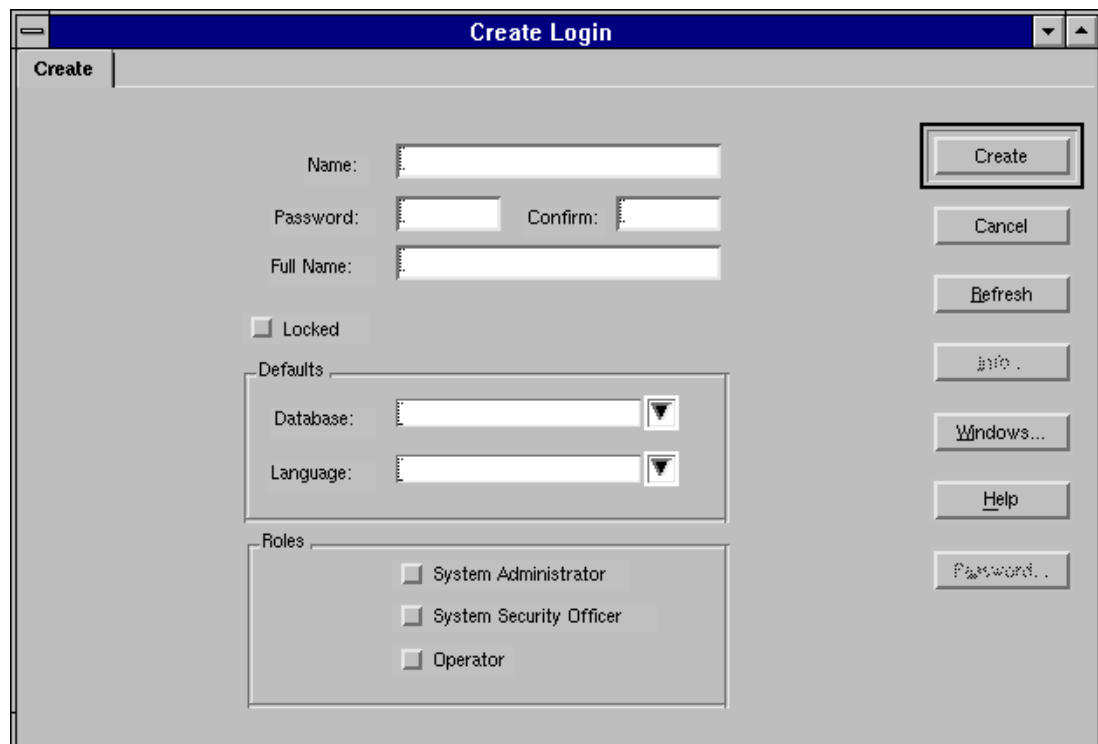


Figure 8-2: Create Login dialog box

2. Enter the login in the Name edit box.
3. Enter the login password in the Password edit box.

4. Re-enter the password in the Confirm box.
5. Enter the full name of the person represented by the login in the Full Name edit box.
6. To lock a login (that is, to prevent it from obtaining any SQL Server access), check the Locked check box. For information about why you might want to lock a login, see “Locking a SQL Server Login” on page 8-13.
7. Select the database that the login points to automatically 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.

8. Choose a language for the login from the list in the Language box.
9. 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 8-3.
10. Click Create to add the new login.

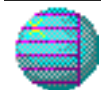
Shortcuts



Click the Login container icon; then, select the Create toolbar button.



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



Click the right mouse button over the 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.

	TME	ESSM	SQL Server
Required roles	any	security	System Administrator

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 is 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

	TME	ESSM	SQL Server
Required roles	any	security	System Security Officer and System Administrator

To change the full name, defaults, or role assignments of a login:

1. Select the login icon you want to modify.
2. From the Login menu, choose Properties.

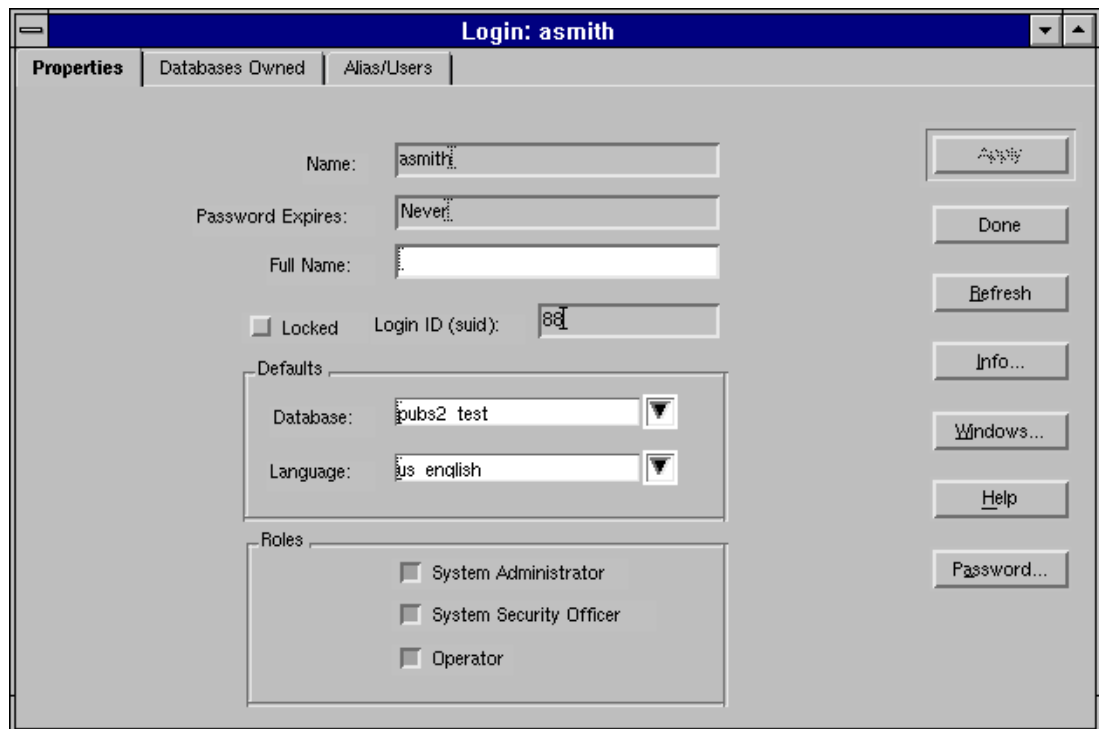





Figure 8-3: Login Properties tab

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 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 8-3.
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 Enterprise 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 8-12.
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 8-12.
2. Click Password. The Change Password dialog box opens.

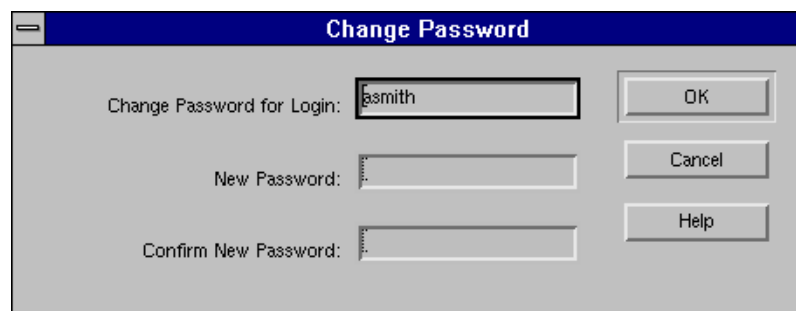


Figure 8-4: Change Password dialog box

3. Enter a new password in the New Password box.
4. Re-enter the password in the Confirm New Password box.
5. 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 8-12.
2. Select Databases Owned to display the Databases Owned tab.

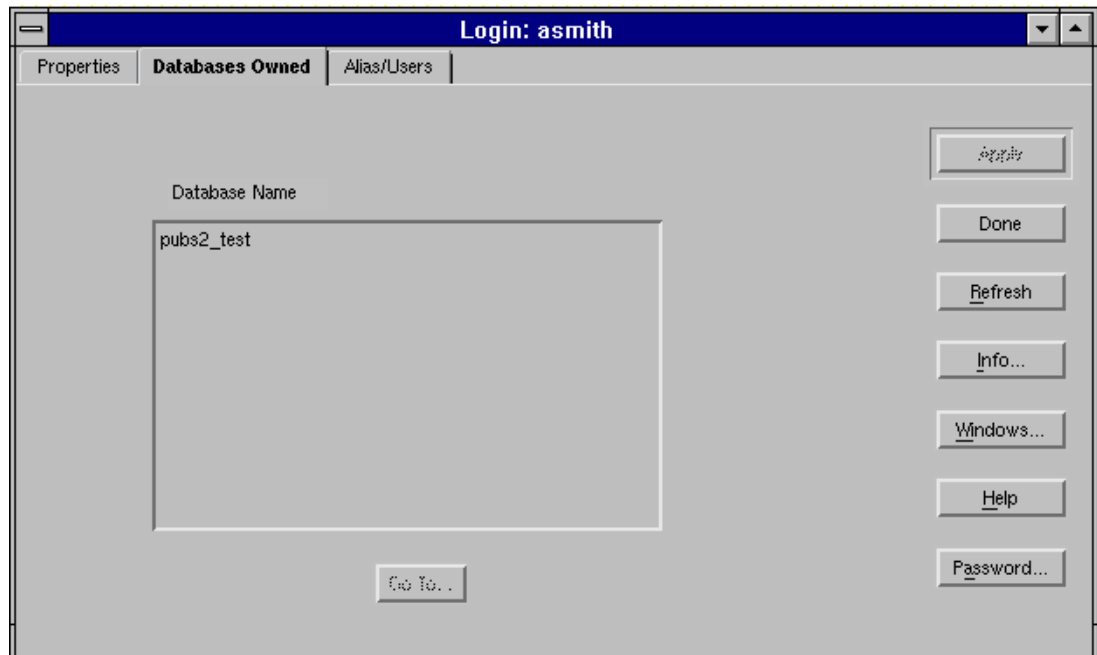


Figure 8-5: Login Databases Owned tab

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.

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 8-12.
2. Select Alias/Users to display the Alias/Users tab.

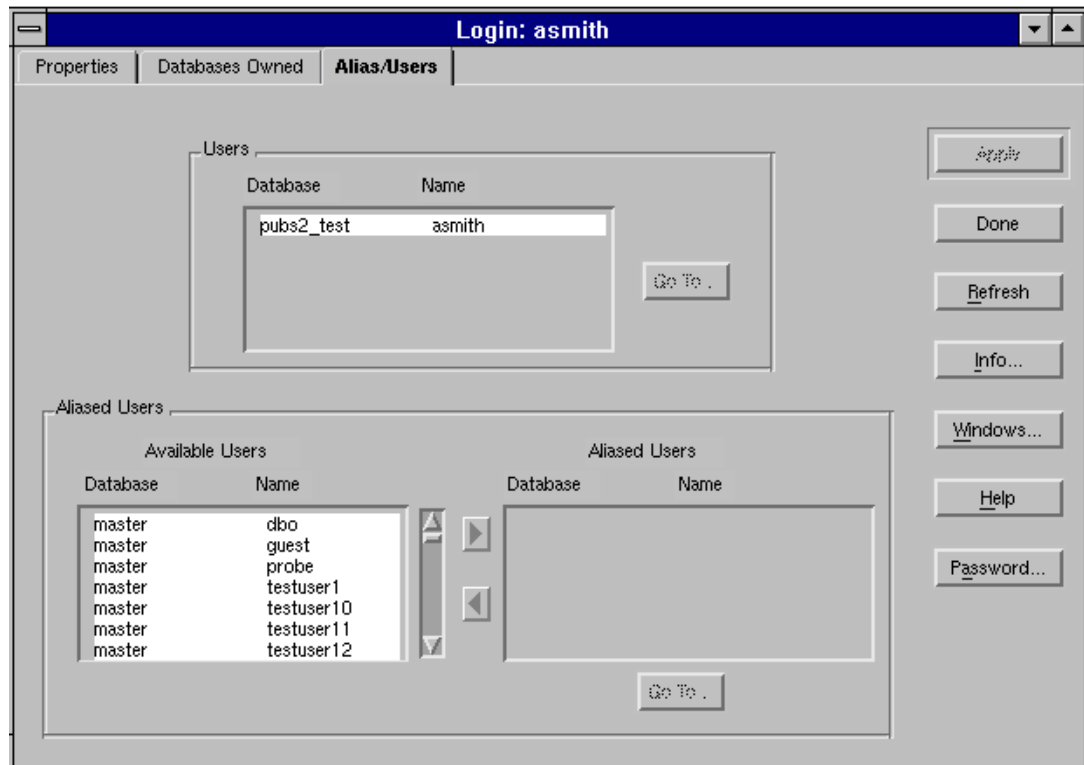





Figure 8-6: Login Alias/Users tab

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 8-12.
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

Navigating to Users Assigned at User Creation

1. Display the Login dialog box, as described in “Modifying Login Information” on page 8-12.
2. Select Alias/Users. The dialog box changes to the Alias/Users tab.
3. Select a user name in the Users list at the top of the Alias/Users tab.
4. Click the Go To button to display the Properties tab for the user.

Navigating to Aliased Users

1. Display the Login dialog box, as described in “Modifying Login Information” on page 8-12.
2. Select Alias/Users. The dialog box changes to the Alias/Users tab.
3. Select a user name in the Aliased Users scrolling list in the Aliased Users group box.
4. 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.

	TME	ESSM	SQL Server
Required roles	any	security	System Administrator or Database Owner

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.

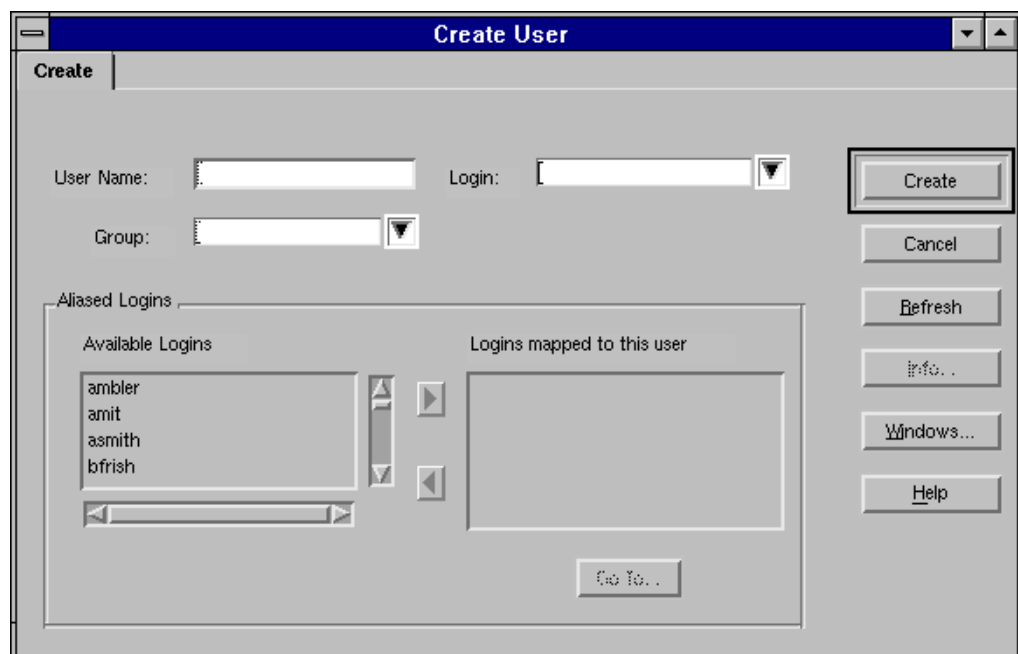


Figure 8-7: Create User dialog box

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



Select the Users container icon in the appropriate database, and select the Create toolbar button.



Drag the login icon that corresponds to the user onto the database icon.



Click the right mouse button over the Users container icon in the appropriate database, and choose Create from the shortcut menu.



Click the right mouse button over the Database icon, and choose Create from the shortcut menu.

Deleting a User

	TME	ESSM	SQL Server
Required roles	any	security	System Administrator or Database Owner

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 8-13. 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

Displaying user properties, objects owned, object permissions, and command permissions requires the following roles:

TME	ESSM	SQL Server
any	any	System Security Officer, System Administrator, or any valid database user

To display information about how the user is defined:

1. Select the icon of the user.
2. From the User menu, choose Properties.

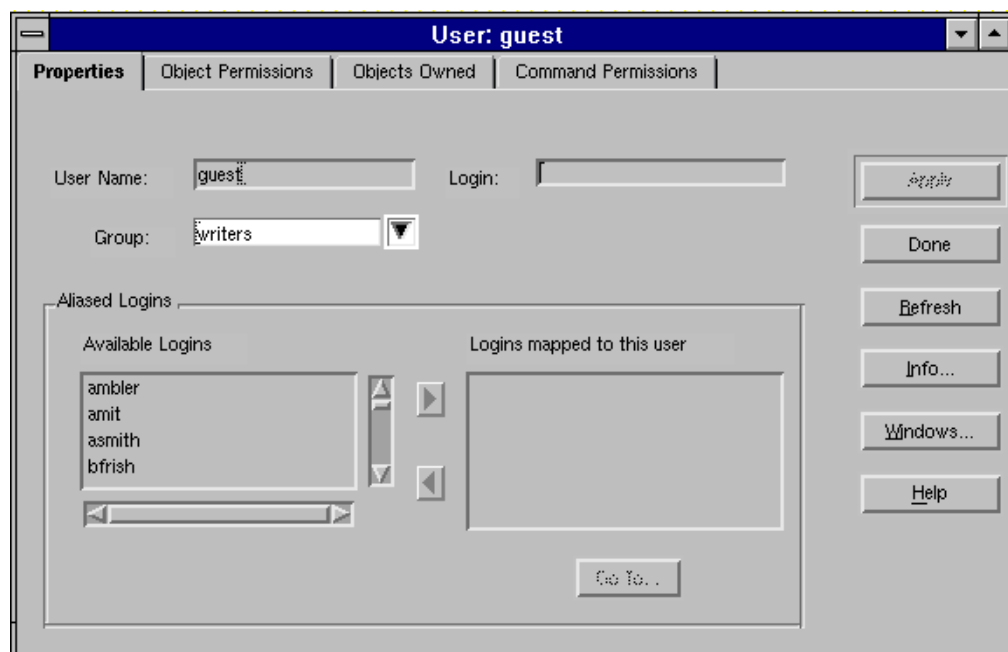


Figure 8-8: User Properties tab

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

	TME	ESSM	SQL Server
Required roles	any	security	System Administrator or Database Owner

The user properties that you can change are the user’s group membership and aliased logins.

To modify these properties:

1. Display the User dialog box, as described in “Displaying User Properties” on page 8-23.
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 8-23.
2. Select Objects Owned. The dialog box changes to the Objects Owned tab.

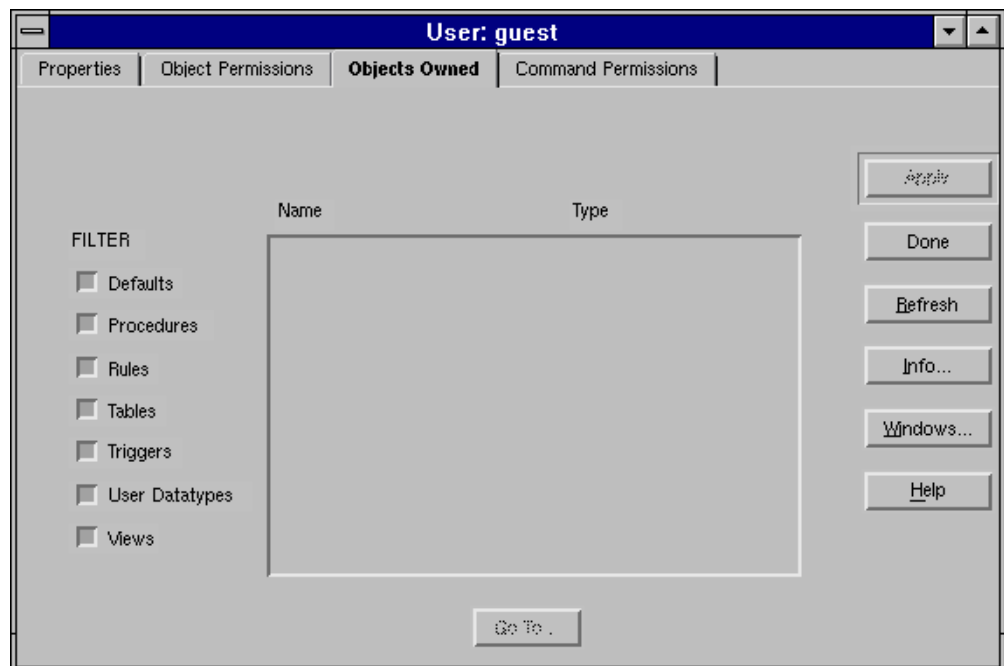


Figure 8-9: User Objects Owned tab

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 about assigning permissions by updating database objects, see the description of each object in Chapter 9, “Managing Database Objects.”

To grant or revoke object permissions:

1. Display the User dialog box, as described in “Displaying User Properties” on page 8-23.
2. Select Object Permissions. The dialog box changes to the Object Permissions tab.

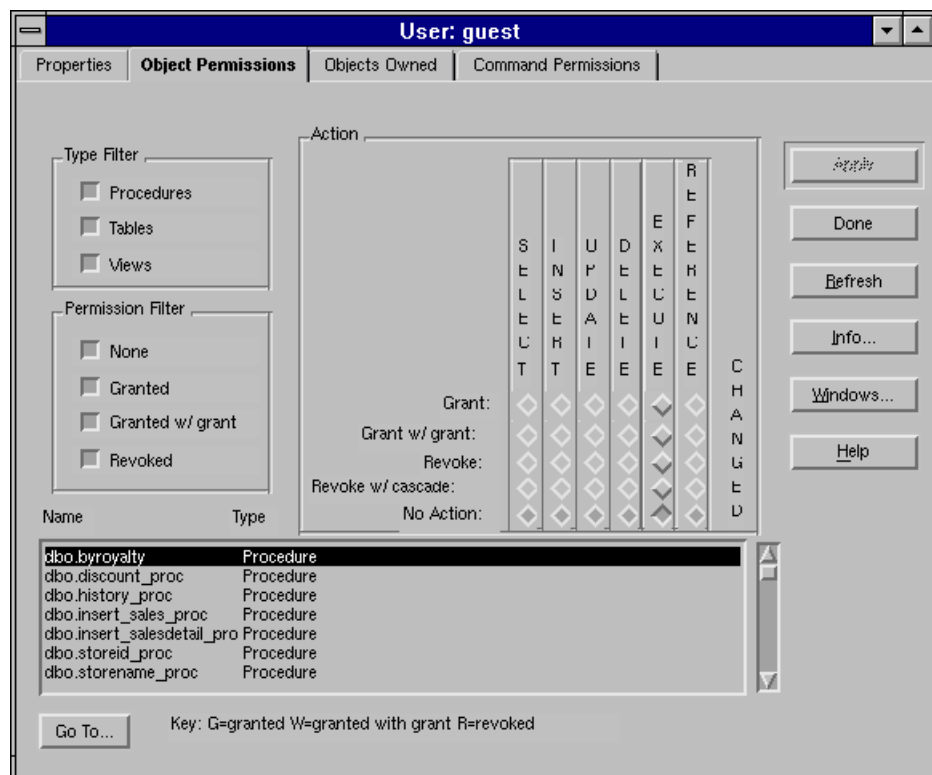


Figure 8-10: User Object Permissions tab

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.

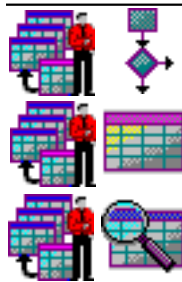
7. Click Apply to update the user's permissions in the database. Enterprise SQL Server Manager updates the list with one of the following codes in each column that corresponds to the type of permission you changed:
 - G—granted
 - W—granted with grant
 - R—revoked
 - Blank—no permission explicitly assigned

► **Note**

Enterprise 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.

► **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 8-23.
2. Select Command Permissions. The dialog box changes to the Command Permissions tab.

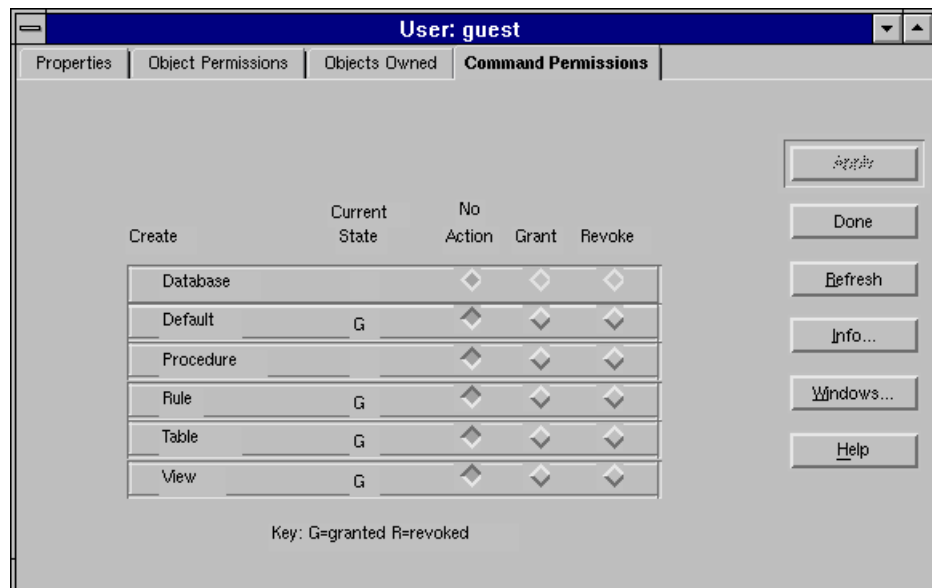


Figure 8-11: User Command Permissions tab

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. Enterprise 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**

Enterprise 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).

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

	TME	ESSM	SQL Server
Required roles	any	security	System Administrator or Database Owner

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.

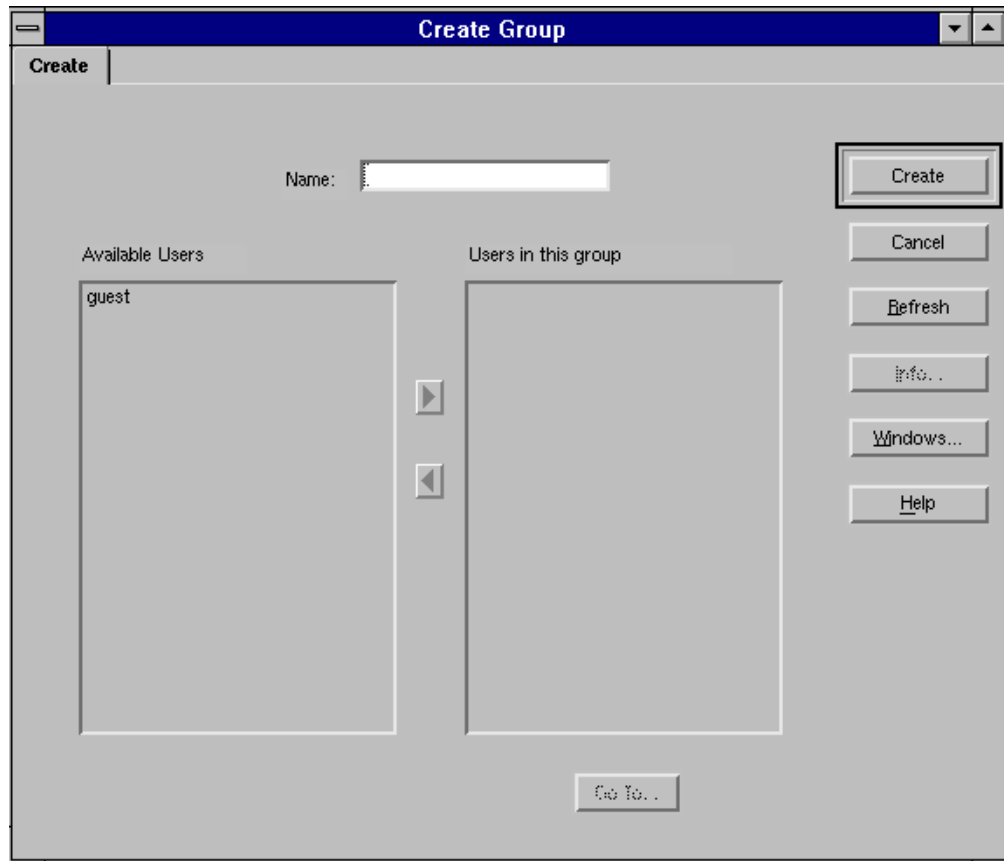


Figure 8-12: Create Group dialog box

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

	TME	ESSM	SQL Server
Required roles	any	security	System Administrator or Database Owner

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

To remove the users from a group:

1. Click the icon of the group to modify.
2. From the Group menu, choose Properties.

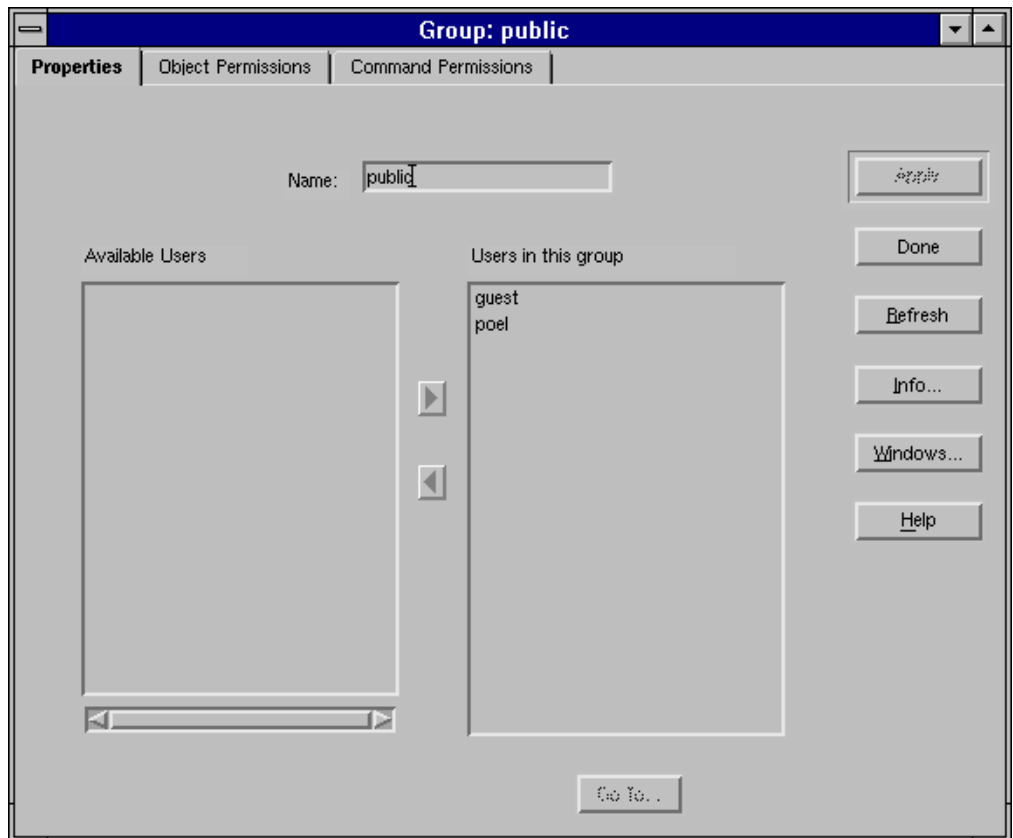


Figure 8-13: Group Properties tab

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

	TME	ESSM	SQL Server
Required roles	any	any	System Security Officer or valid database user

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.

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

	TME	ESSM	SQL Server
Required roles	any	security	System Administrator or Database Owner

To modify the membership of a group:

1. Display the Group dialog box as described in “Displaying Group Properties” on page 8-36.
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 about assigning permissions by updating database objects, see the description of each object in Chapter 9, “Managing Database Objects.”

To grant or revoke object permissions:

1. Display the Group dialog box, as described in “Displaying Group Properties” on page 8-36.
2. Select Object Permissions. The dialog box changes to the Object Permissions tab.

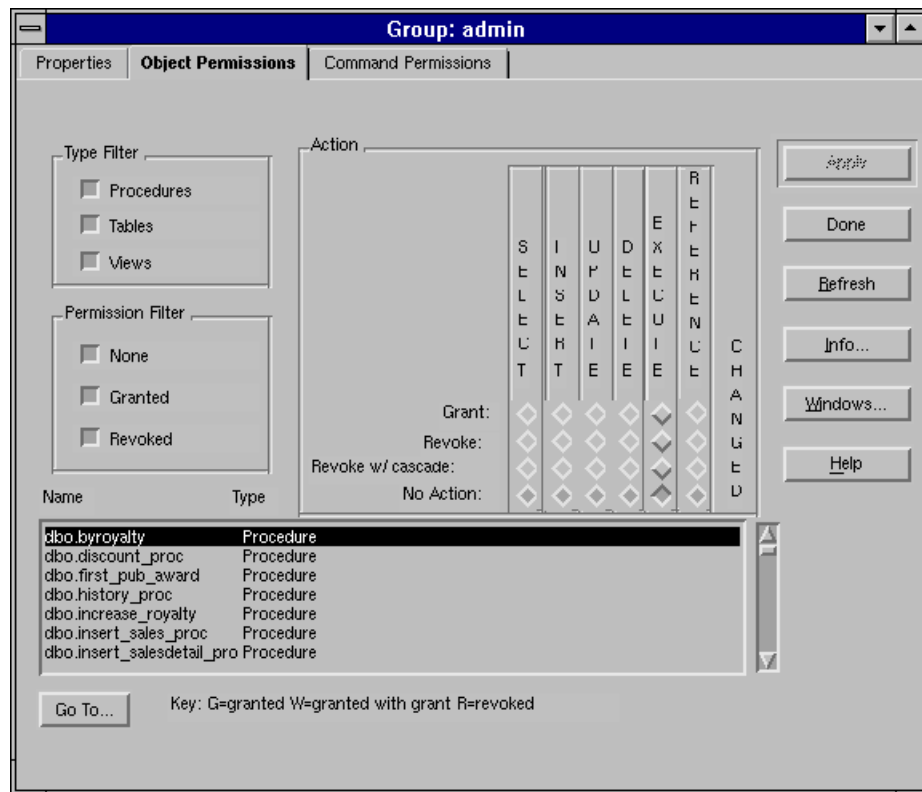


Figure 8-14: Group Object Permissions tab

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.

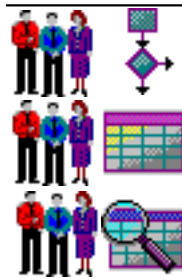
7. Click Apply to update the group's permissions in the database. Enterprise SQL Server Manager updates the list with one of the following codes in each column that corresponds to the type of permission you changed:
 - G—granted
 - R—revoked
 - Blank—no permission explicitly assigned

► **Note**

Enterprise 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 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 8-36.
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.

► **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 8-36.
2. Select Command Permissions. The dialog box changes to the Command Permissions tab.

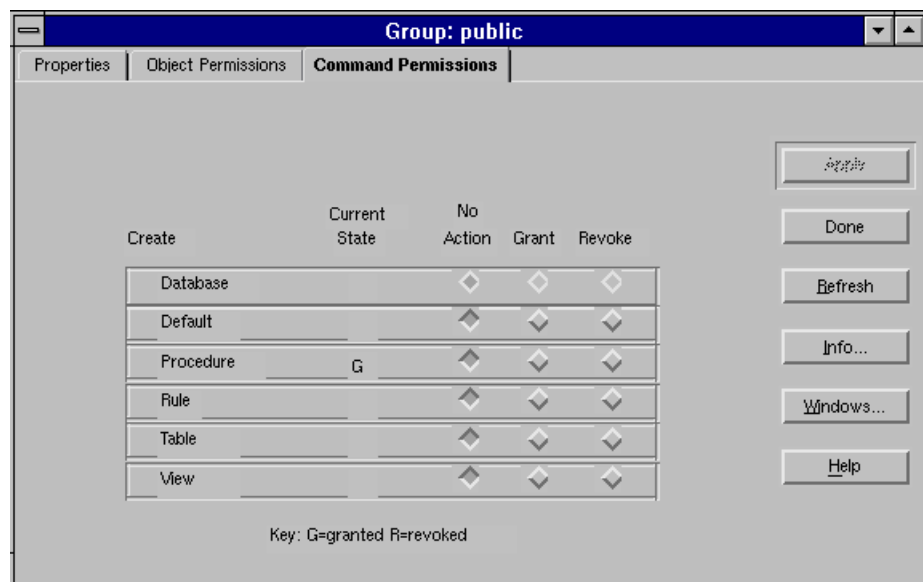


Figure 8-15: Group Command Permissions tab

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. Enterprise 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**

Enterprise 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 Enterprise SQL Server Manager protection system as implemented in Enterprise SQL Server Manager. The type of user listed as the default 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	x			x			x
Add columns to a table					x		x
Bind a default to a table column or user datatype					x		x
Bind a rule to a table column or user datatype					x		x
Change database device default status	x						x
Change database options	x			x			x
Change database ownership	x			x			x
Change database storage allocations	x					(1)	
Change group membership	x			x			x
Change login defaults or fullname	x						x
(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 login password (5)	x	x					x
Create a named cache	x					x	
Create a database	x					x	
Create a database device	x						x
Create a default				x		x	
Create a dump device	x						x
Create a group	x			x			x
Create an index					x		x
Create a login		x					x
Create a procedure				x		x	
Create a remote server	x						x
Create a rule				x		x	
Create a segment	x			x			x
Create a table		(2)		x		x (2)	
Create a trigger					x		x
Create a user	x			x			x
Create a view				x		x	
Define remote login mappings		x					x
Delete an alias	x			x			x
Delete a database or dump device	x						x
(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
Delete a database object					x(3)		x
Delete a group	x			x			x
Delete a login	x						x
Delete a remote server	x						x
Delete a segment	x						x
Delete a user	x			x			x
Delete a user datatype	x			x			x
Disable disk mirroring	x						x
Dump a database			x	x			x
Dump a transaction log			x	x			x
Enable disk mirroring	x						x
Execute dbcc commands				x			x
Execute a procedure					x(4)	x	
Extend a segment	x			x			x
Grant permission on a database object					x	x	
Grant permission to create a database object				x		x	
Grant roles to logins	x	x					x
(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
Insert a row in a table					x (3)	x	
Issue a database checkpoint				x			x
Issue an update statistics command					x		x
Kill a user process	x						x
Load a database from a backup			x	x			x
Load a transaction log from a backup			x	x			x
Lock a login	x	x					x
Move the transaction log to a different device	x		x				x
Place new table or view allocations on a segment	x			x	x		x
Reset SQL Server configuration options	x						x
Re-enable disk mirroring on an inactive mirror device	x						x
Revoke permission on an object					x		x
Revoke permission to create an object				x			x
Select rows in a table					x (3)	x	
(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
Shut down SQL Server	x						x
Unbind a default from a table column or user datatype					x		x
Unbind a rule from a table column or user datatype					x		x
Update rows in a table					x (3)	x	
(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

To create database objects, you must have the following roles:

Table 9-1: Role requirements for creating database objects

Object	TME	ESSM	SQL Server
Default	any	schema	Database Owner. If you are specifying bindings, you must be the owner of the object to which the default is bound.
Index	any	schema	Table owner
Procedure	any	schema	Database Owner
Rule	any	schema	Database Owner. If you are specifying bindings, you must be the owner of the object being bound to, or you must be a user or in a group with "create rule" permission.
Trigger	any	schema	Table owner
User datatype	any	schema	Database Owner
View	any	schema	Database Owner

To create an object:

1. Select the icon of the database in which 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.

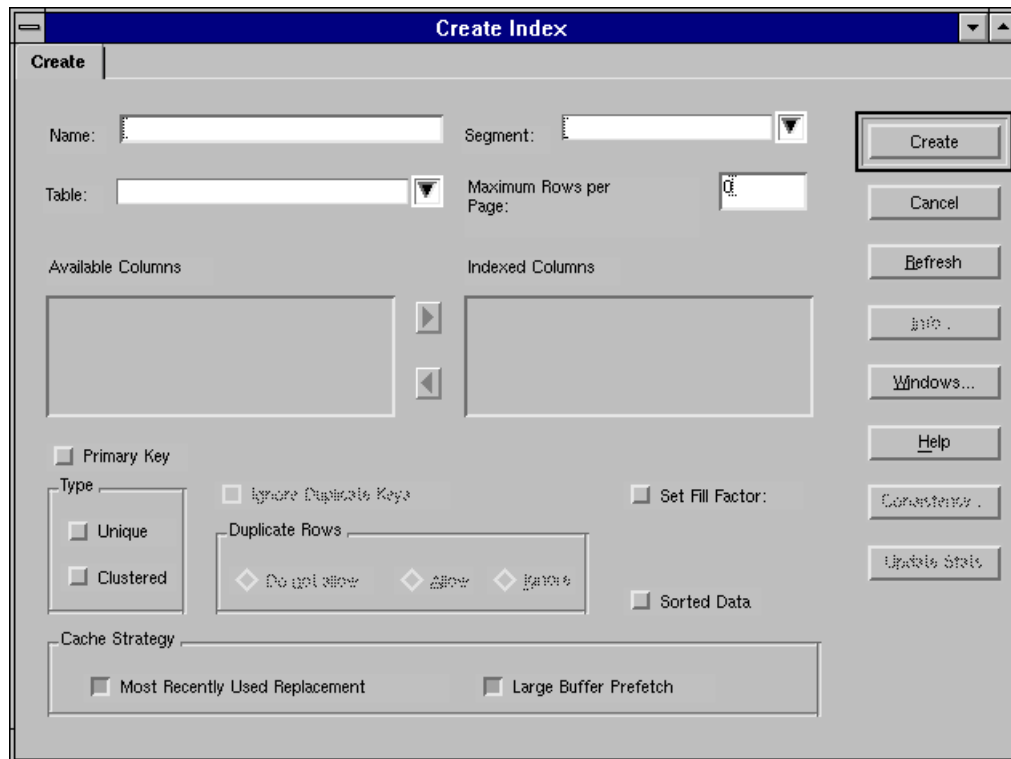


Figure 9-1: Create dialog box for an index

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



Select the container icon for the type of object you want to create. Select the Create toolbar button.



Click the right mouse button over the container icon for the object you want to create, and choose Create from the shortcut menu.



Click the right mouse button over the database icon. Choose Create from the shortcut menu; then, choose the command for the object you want.

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.

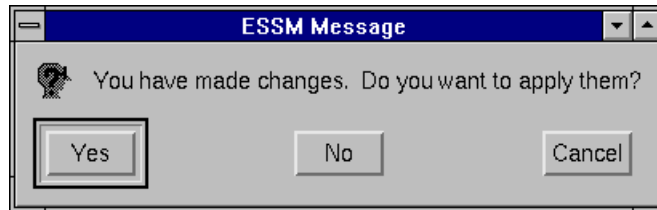


Figure 9-2: Confirmation dialog box

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

	TME	ESSM	SQL Server
Required roles	any	any, except for an index, which requires schema	any

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).

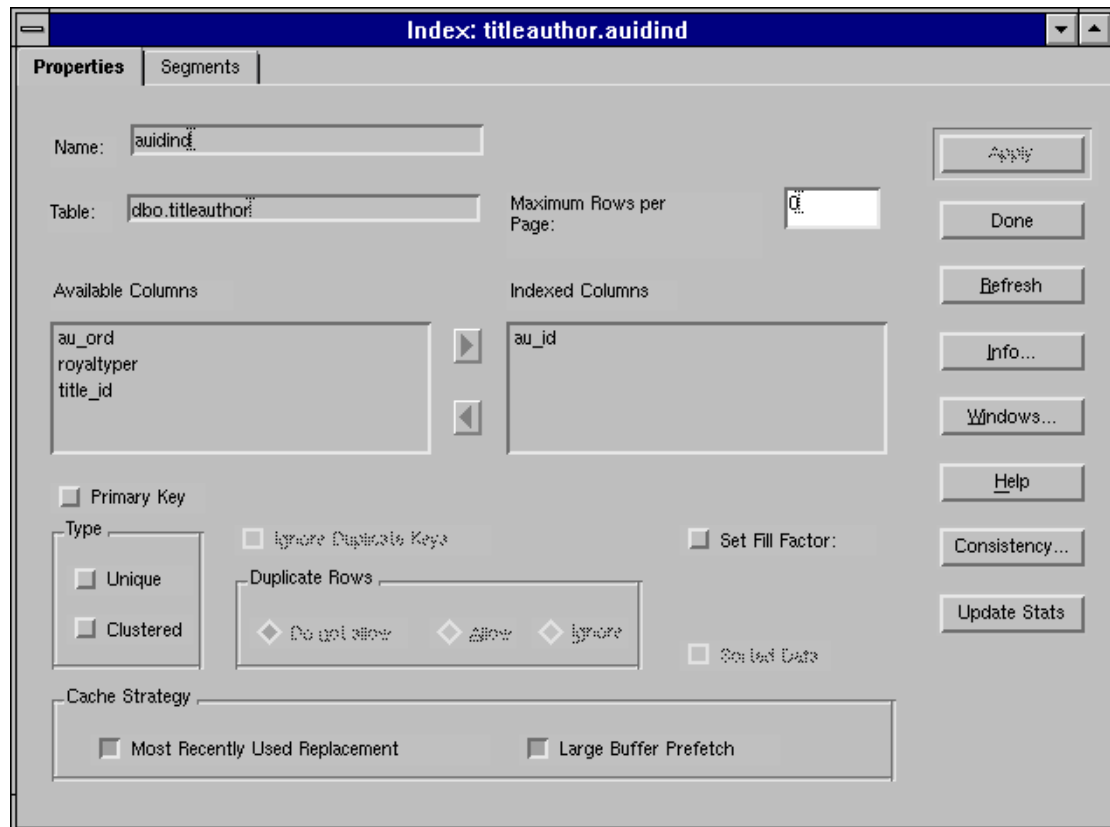


Figure 9-3: Properties tab for an index

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.

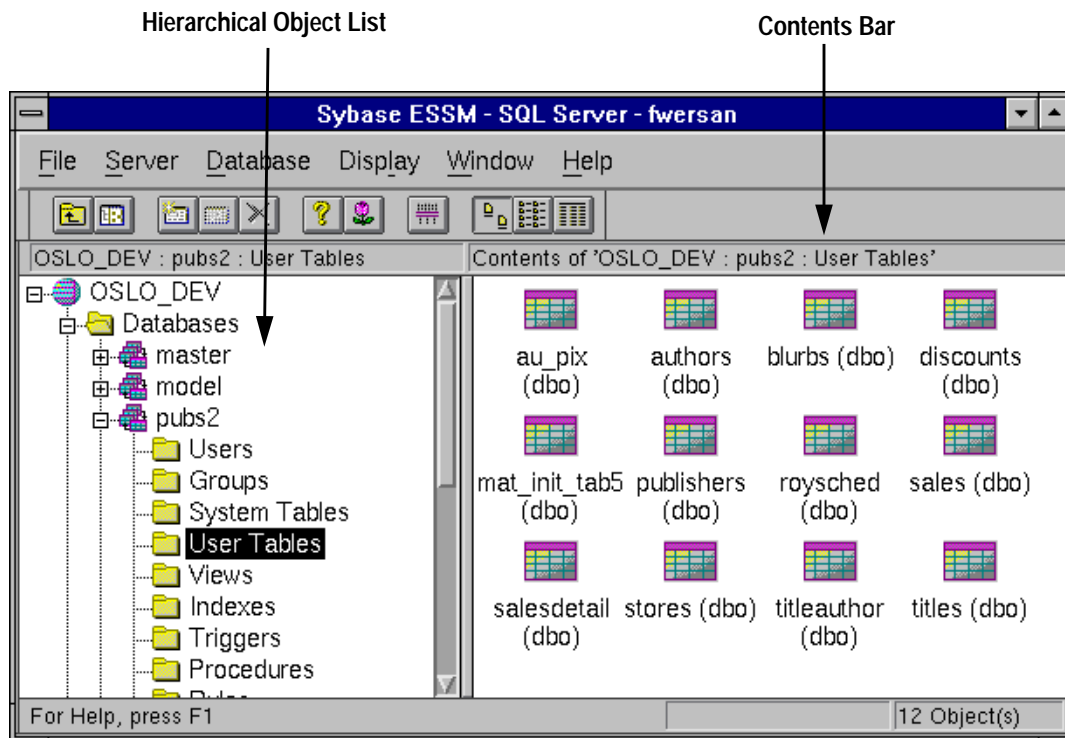


Figure 9-4: Voyager window showing object hierarchy

Displaying an Object Relative to Other Objects

To see information about 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 buttons for Dependencies and Permissions.

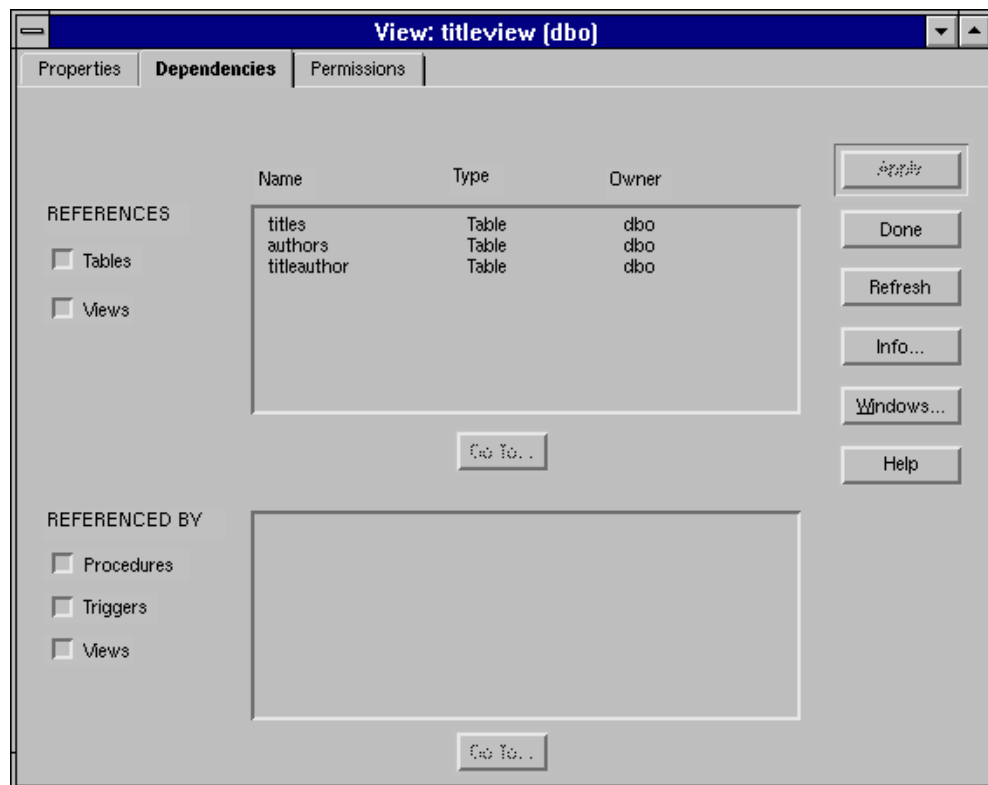


Figure 9-5: Dependencies tab for a view

Copying an Object

If you want to copy a database object, use profile management to distribute a profile of the object. To quickly start a profile distribution, you can drag the icon for a profile onto the subscriber to which you want to distribute it. For information about profile management, see Chapter 11, “Enterprise SQL Server Manager Profile Managers” and Chapter 12, “Managing ESSM Profiles.” You can also copy objects using Enterprise SQL Server Manager *scopy* commands. For information about these commands, see the *Enterprise SQL Server Manager Reference Manual*.

Deleting an Object

To delete database objects, you must have the following roles:

Table 9-2: Role requirements for deleting database objects

Object	TME	ESSM	SQL Server
Default	any	schema	Database Owner. If you are specifying bindings, you must be the owner of the object to which the default is bound.
Index	any	schema	Index owner
Procedure	any	schema	Stored procedure owner
Rule	any	schema	Rule owner
Trigger	any	schema	Trigger owner
User datatype	any	schema	Database Owner or datatype owner
View	any	schema	View owner

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.
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.

Comparing Objects

If you want to compare object properties, use profile management. You can compare the objects in a profile to the objects in a subscriber if you use the Preview Distribution feature. For more information about previewing distributions, see “Previewing a Distribution” in Chapter 12, “Managing ESSM Profiles.”

If you want to compare objects without using profile management, use Enterprise SQL Server Manager `scomp` commands. For information about these commands, see the *Enterprise SQL Server Manager Reference Manual*.

Managing Defaults

A **default** is a value you specify that SQL Server inserts for a table column when no value is provided. Enterprise 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 about 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 10-15.

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

	TME	ESSM	SQL Server
Required roles	any	schema	Database Owner. If you are specifying bindings, you must be the owner of the object being bound to.

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.

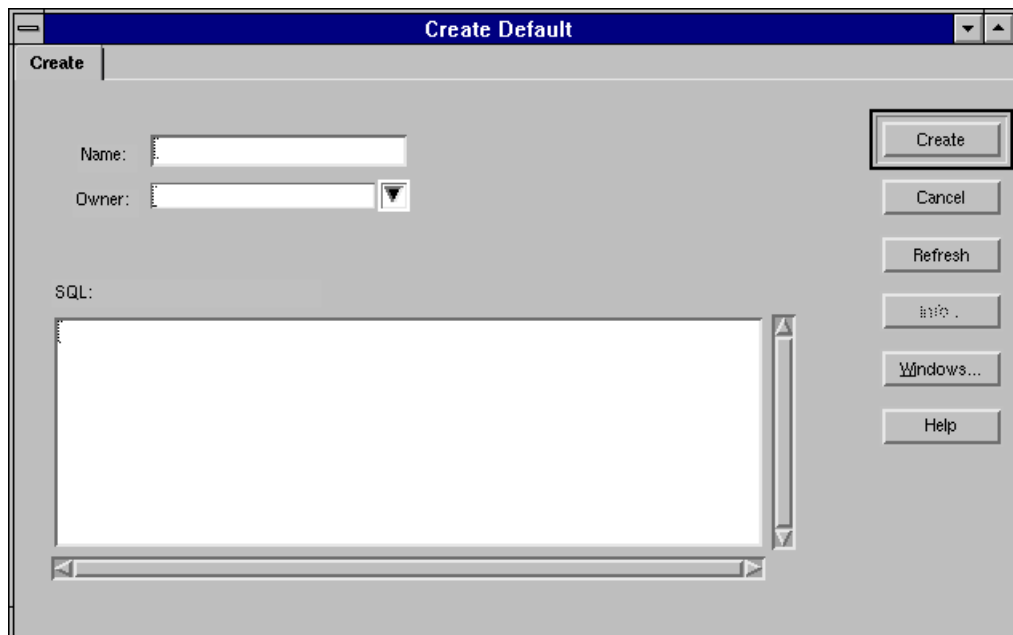


Figure 9-6: Create Default dialog box

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. Enterprise SQL Server Manager uses this expression to generate a create statement for the default in the correct Transact-SQL format.
6. Click Create.

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:

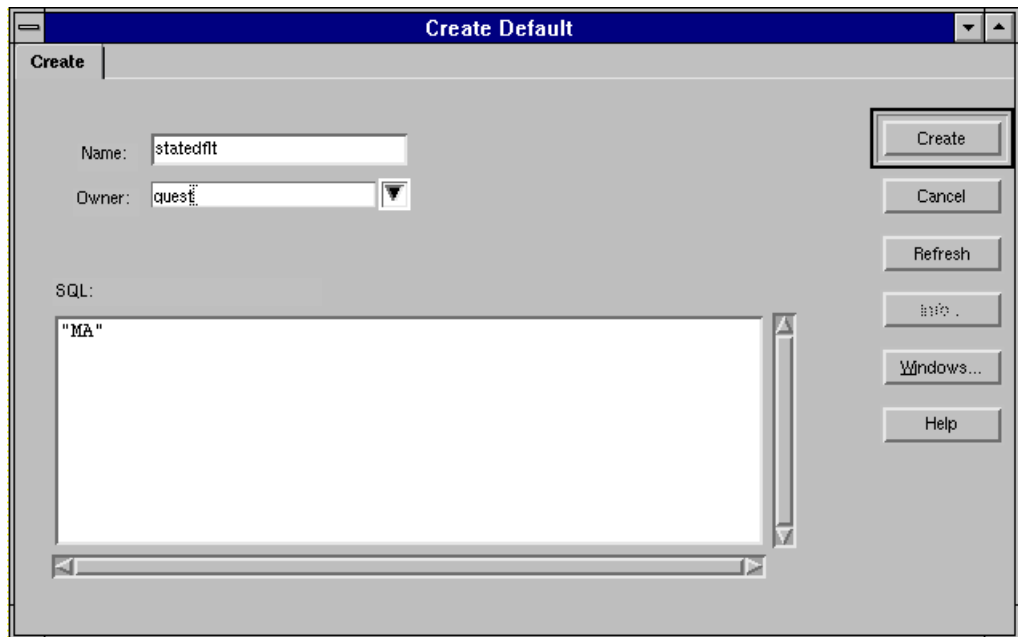


Figure 9-7: Creating the default *statedflt*

Enterprise SQL Server Manager constructs the appropriate SQL query from the data you provide and creates the default shown in the following figure.

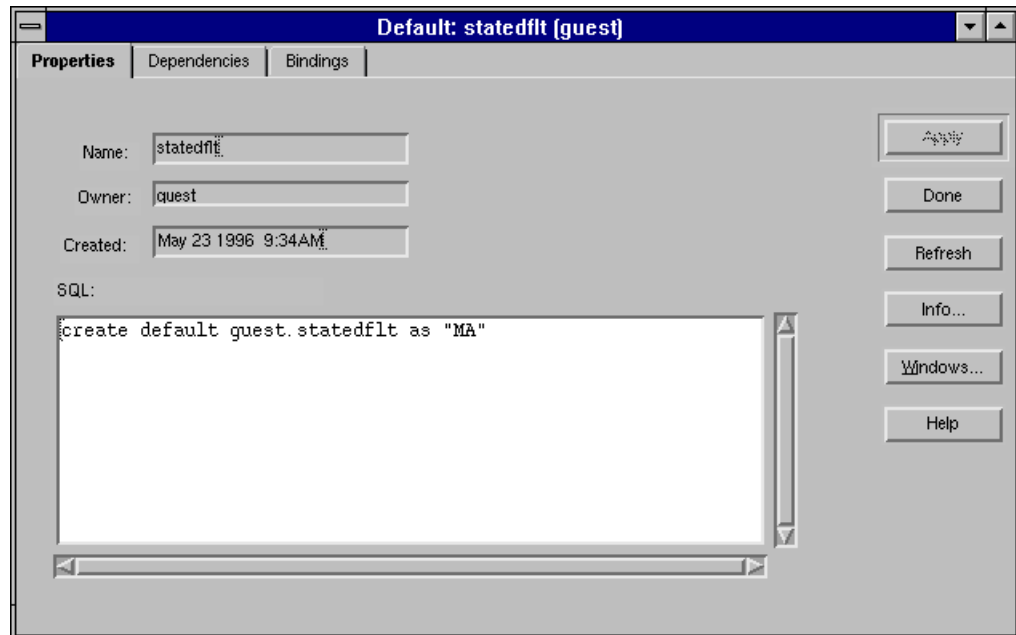


Figure 9-8: statedflt default as created by Enterprise SQL Server Manager

Displaying Default Properties

To display the Properties tab:

1. Select the icon of the default.
2. From the Default menu, choose Properties (Figure 9-8 on page 9-13).

Displaying Default Dependencies

To display dependencies for a default:

1. Display the default Properties tab as described in “Displaying Default Properties” on page 9-13.
2. Click Dependencies. The display changes to the Dependencies tab.

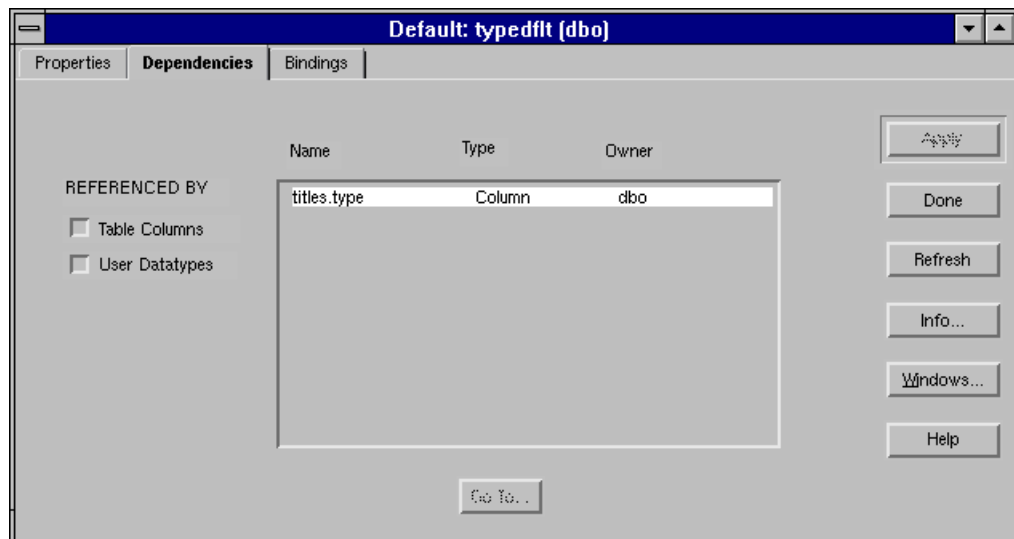


Figure 9-9: Default Dependencies tab

3. To display only columns that reference the default, clear (deselect) 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 9-13.
2. In the list box, select the object you want to view.
3. Click Go To. The Properties tab for the selected object is displayed.

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.

Columns that do not allow null values should 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.

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.

► *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 9-13.
2. Click Bindings. The display changes to the Bindings tab.

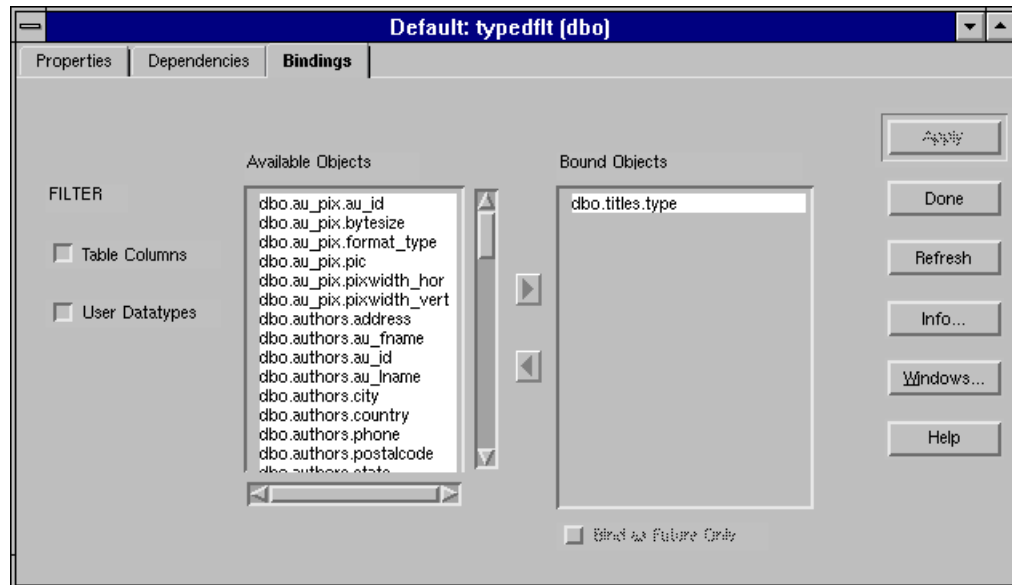


Figure 9-10: Default Bindings tab

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 9-13.
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.

Shortcuts



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 9-13.
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 Enterprise 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:

	TME	ESSM	SQL Server
Required roles	any	schema	table owner

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.

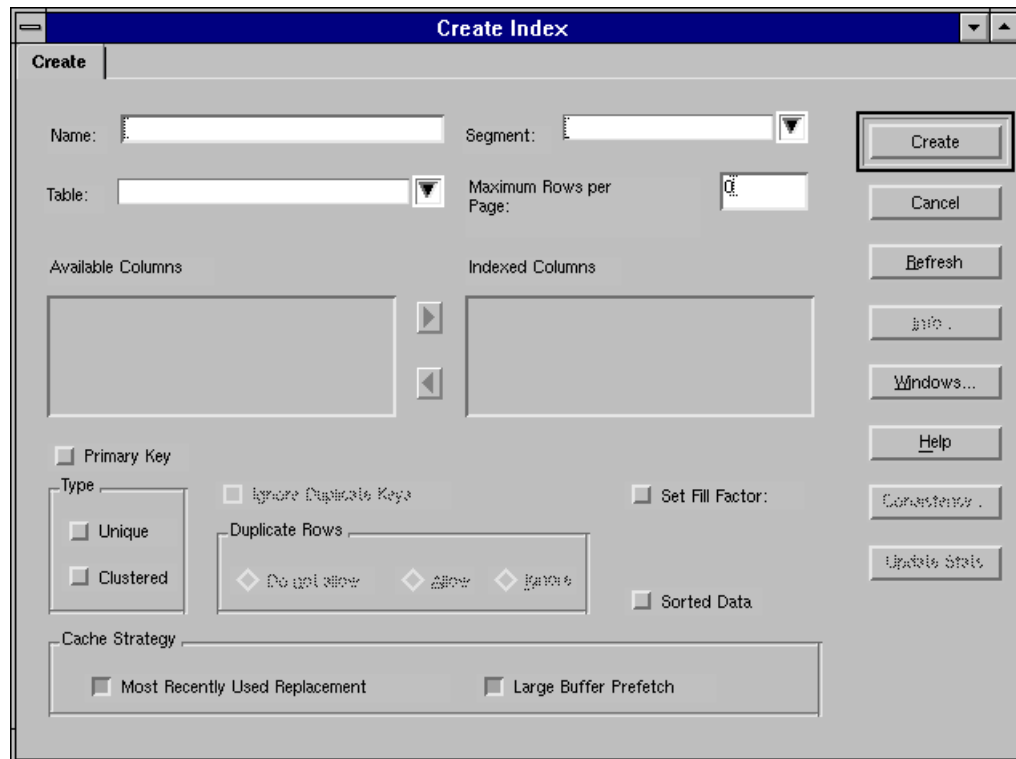



Figure 9-11: Create Index dialog box

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:
 - Display the Table Properties tab by selecting the table icon and choosing Properties from the Table menu.
 - 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 implement primary key constraints, select the Primary Key box. Enterprise SQL Server Manager automatically sets the controls that support primary key creation. Optionally, to specify that the index is clustered as well as unique, choose Clustered. Skip to step 12.

► *Note*

Do not confuse the unique and primary key integrity constraints with the information defined by the `sp_primarykey`, `sp_foreignkey`, and `sp_commonkey` system procedures. The unique and primary key constraints create indexes to define unique or primary key attributes of table columns. The system procedures define the logical relationship of keys for table columns that you enforce by creating indexes and triggers.

9. From the Type group box, choose 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 managing SQL Server release 11.0, 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.

Displaying Index Properties:

TME	ESSM	SQL Server
any	schema	table and index owner

To display index properties:

1. Select the icon of the index to display.
2. From the Index menu, choose Properties.

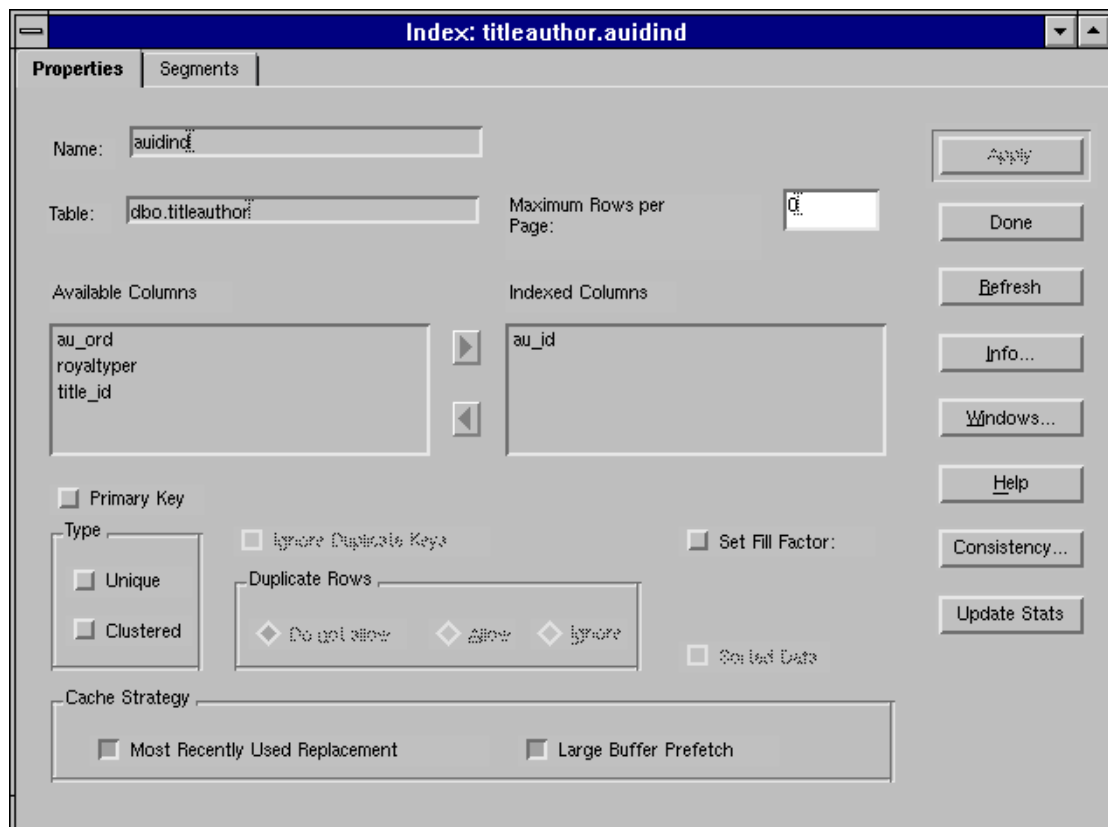


Figure 9-12: Index Properties tab

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 9-25.
2. Click Segments. The display changes to the Segments tab.

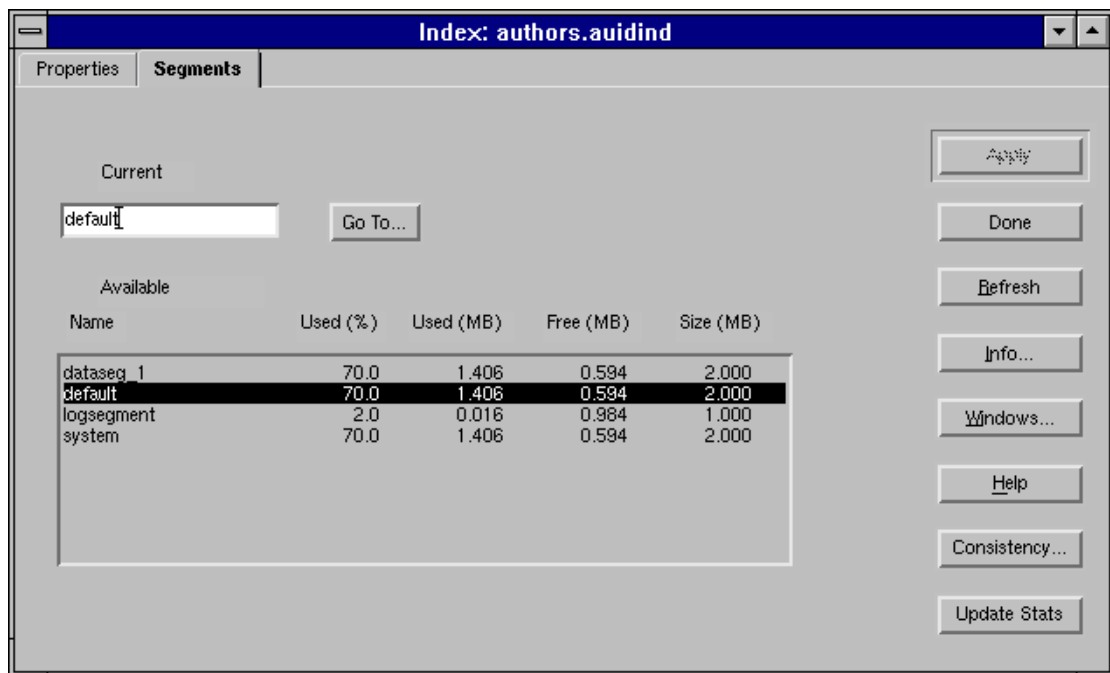


Figure 9-13: Index Segments tab

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.

	TME	ESSM	SQL Server
Required roles	any	schema	Database Owner

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.

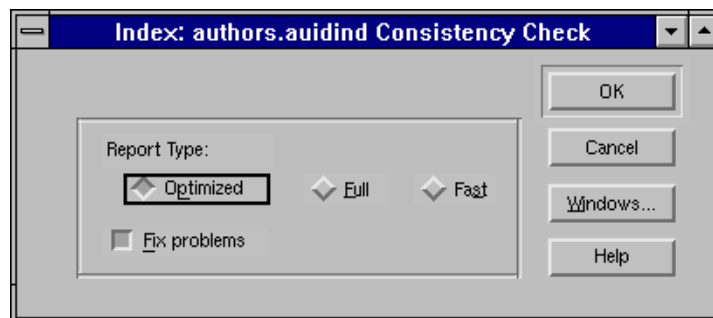


Figure 9-14: Index Consistency Check dialog box

3. The report types correspond to the `dbcc indexalloc` command. The following report types are available:
 - Select **Optimized** to report allocation pages listed in the Object Allocation Map pages for the index. (For information about the Object Allocation Map, see *System Administration Guide*.)
 - Select **Full** to report all types of allocation errors.
 - Select **Fast** to report pages that are referenced but not allocated in the extent.

By default, **dbcc indexalloc** fixes allocation errors as it executes. Enterprise 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.

4. Click **OK** to check index consistency. Enterprise 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.

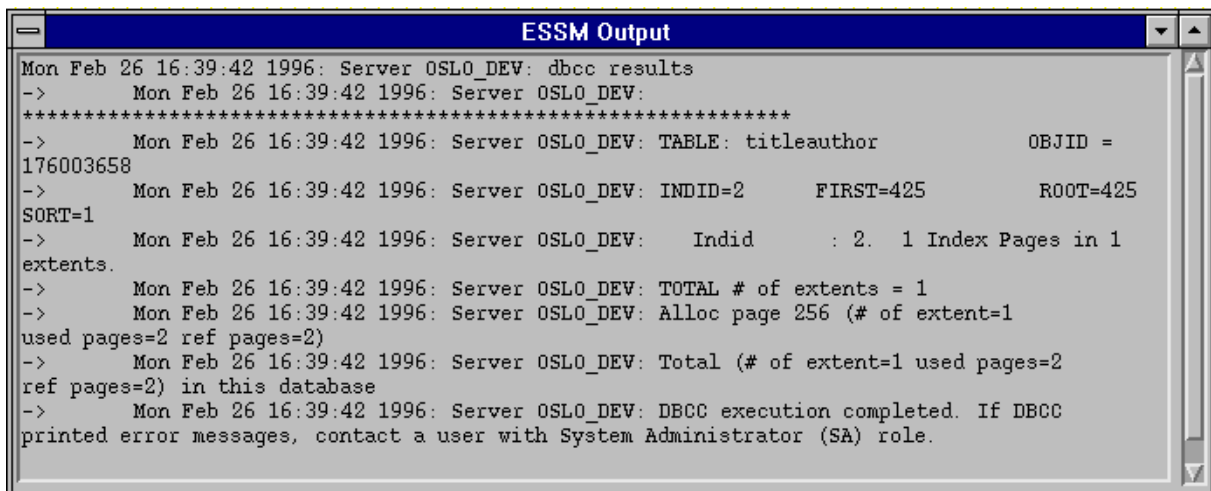




Figure 9-15: Output from Full option of index consistency check

Shortcuts

-
-  From the Index dialog box, click **Consistency**.
-
-  Click the right mouse button over the index icon; then, choose **Consistency** from the shortcut menu.
-

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`.

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.

	TME	ESSM	SQL Server
Required roles	any	schema	Database Owner

To 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.

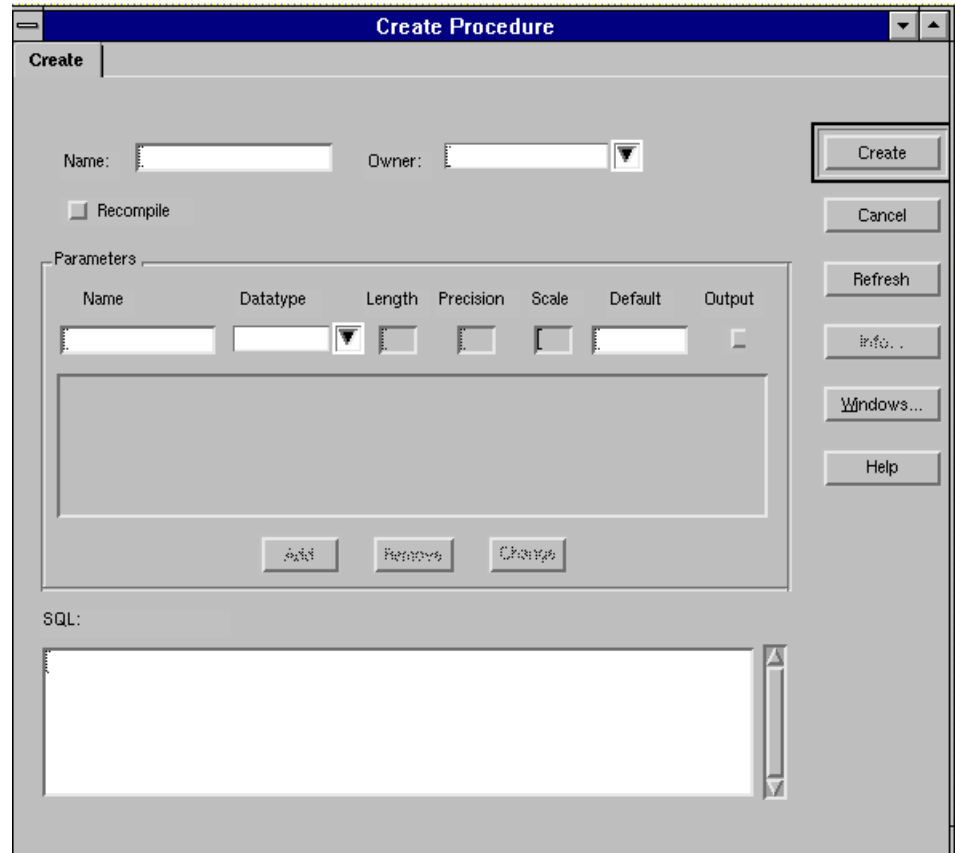


Figure 9-16: Create Procedure dialog box

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 @ (at) 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.

Displaying Procedure Properties

To display the Properties tab:

1. Select the procedure icon.
2. From the Procedure menu, choose Properties.

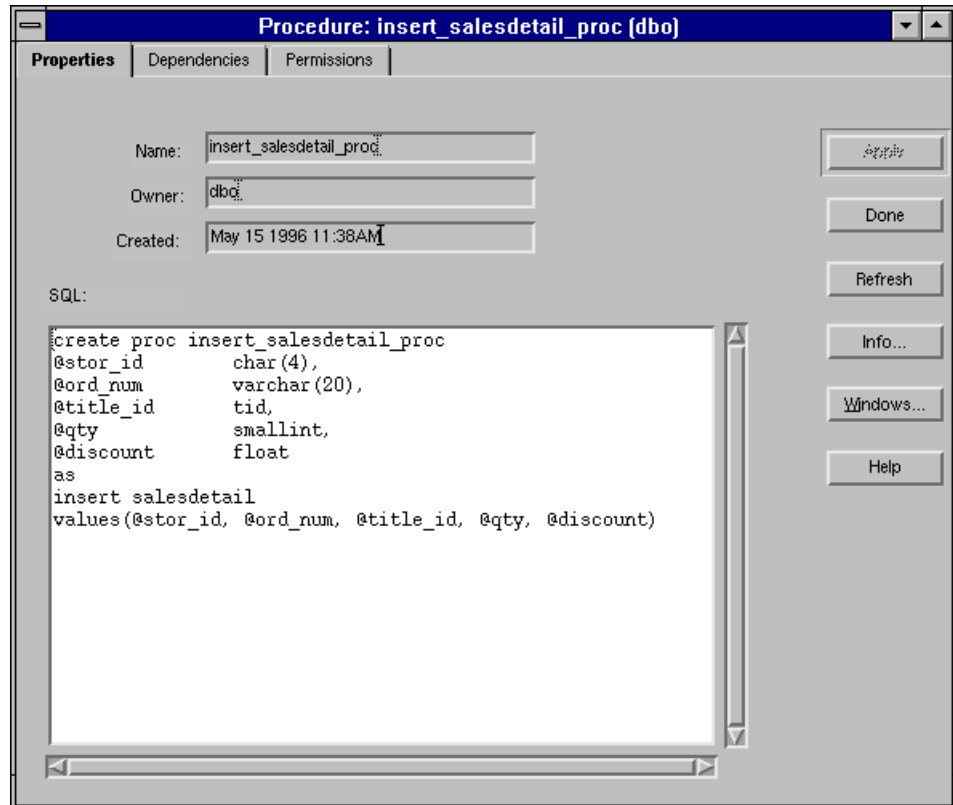


Figure 9-17: Procedure Properties tab

Displaying Procedure Dependencies

To display dependencies for a procedure:

1. Display the Procedure Properties tab as described in “Displaying Procedure Properties” on page 9-34.
2. Click Dependencies. The display changes to the Dependencies tab.

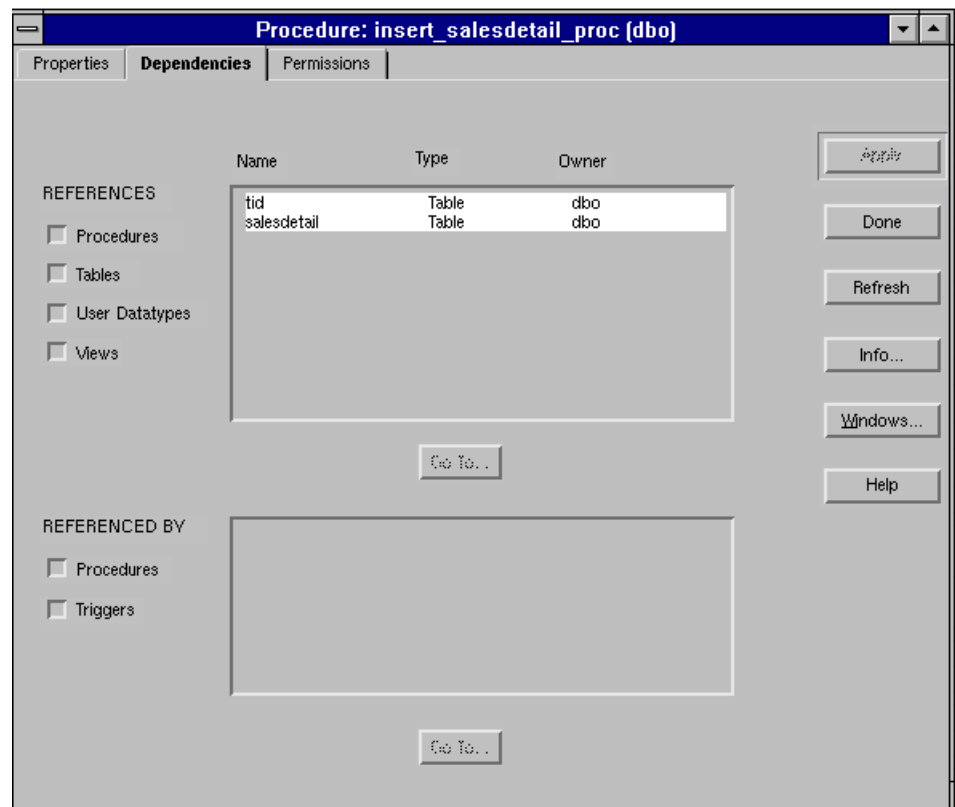


Figure 9-18: Procedure Dependencies tab

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 9-34.
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

Enterprise 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.

	TME	ESSM	SQL Server
Required roles	any	security	System Administrator or Database Owner

To update procedure permissions:

1. Display the Procedure Properties tab, as described in “Displaying Procedure Properties” on page 9-34.
2. Click Permissions. The display changes to the Permissions tab.

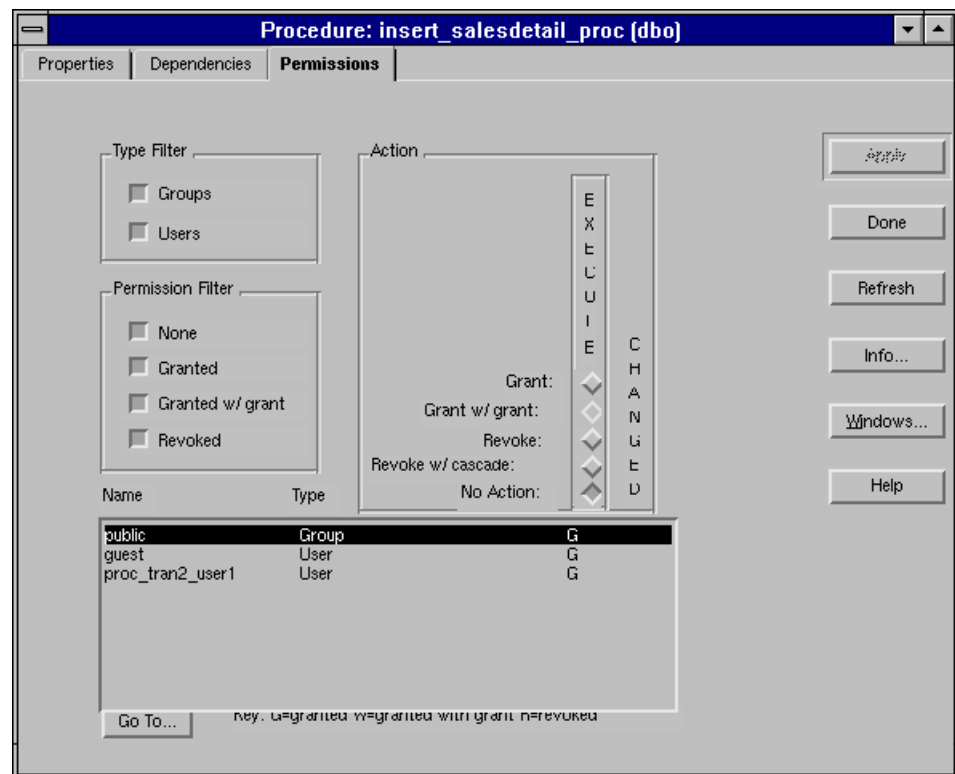


Figure 9-19: Procedure Permissions tab

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 9-34.
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.

Enterprise 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 character set**.
- 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 about 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 10-15.

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

	TME	ESSM	SQL Server
Required roles	any	schema	Database Owner. If you are specifying bindings, you must be the owner of the object being bound to, or you must be a user or in a group with "create rule" permission.

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.

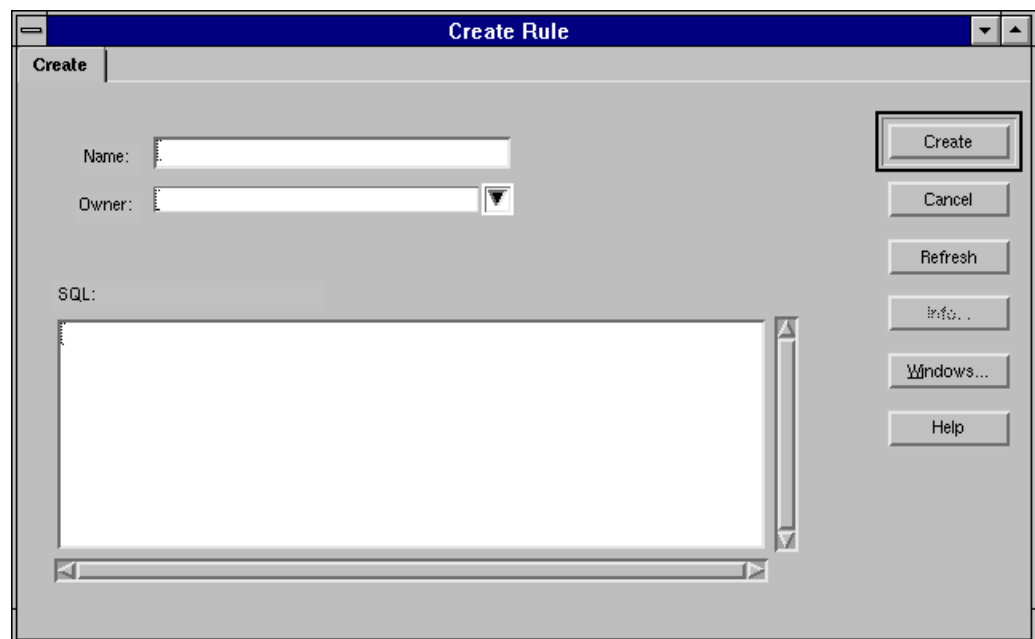


Figure 9-20: Create Rule dialog box

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; Enterprise 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.

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.

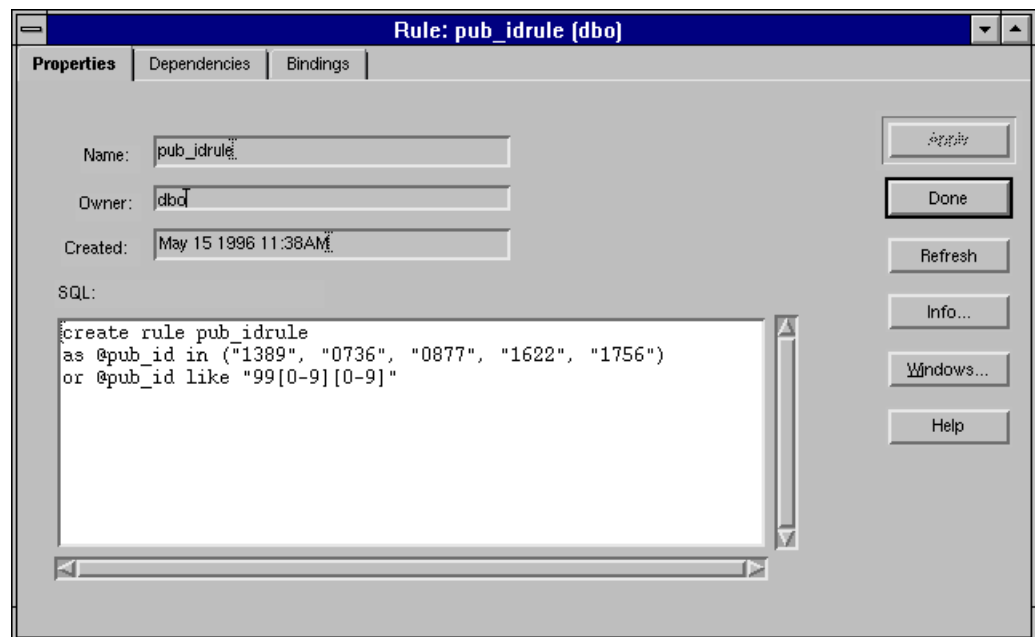


Figure 9-21: Rule Properties tab

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.

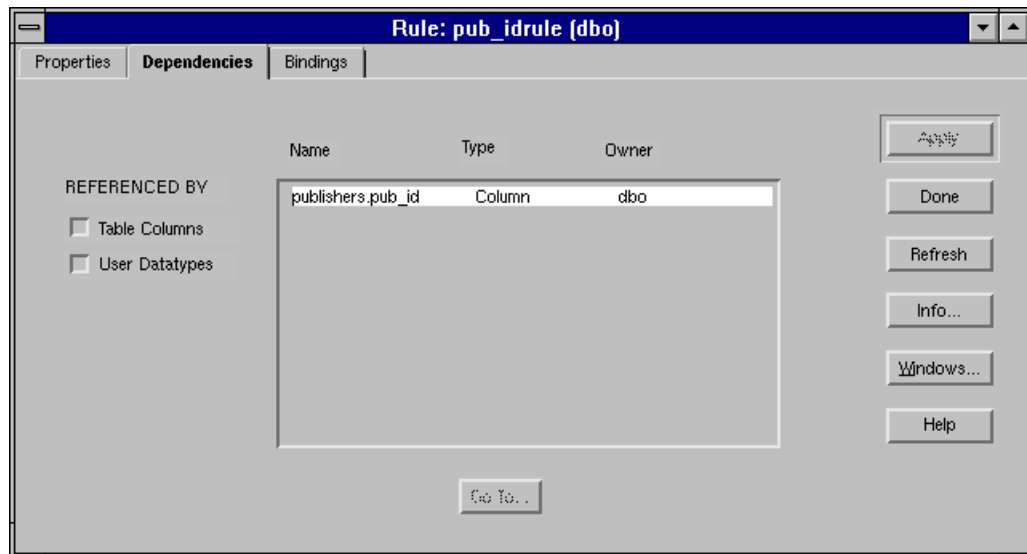


Figure 9-22: Rule Dependencies tab

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.

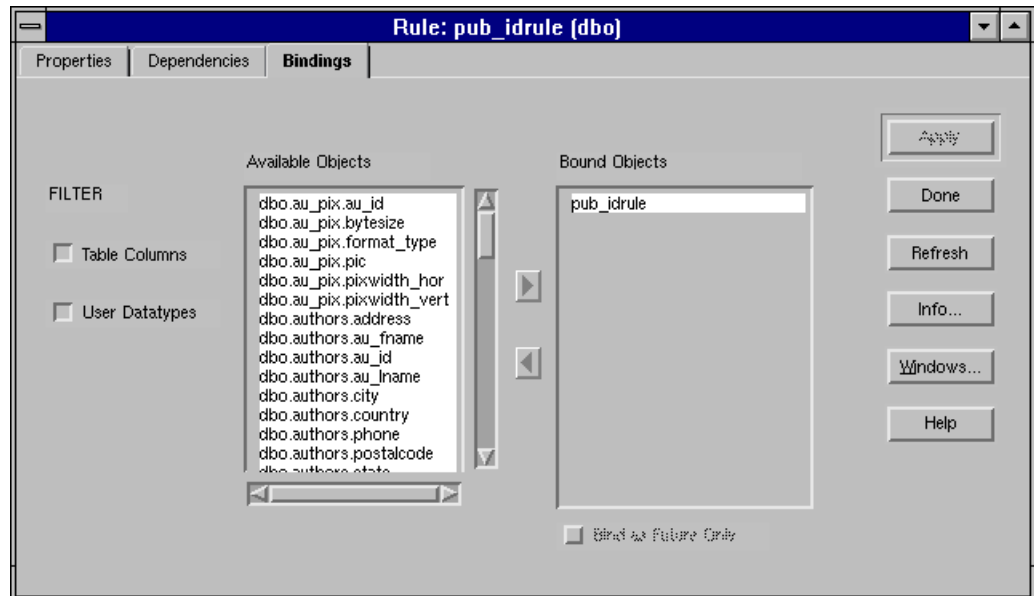


Figure 9-23: Rule Bindings tab

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



To display the Bindings tab, drag and drop the rule icon on the table or user datatype icon.

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.

	TME	ESSM	SQL Server
Required roles	any	schema	table owner

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.

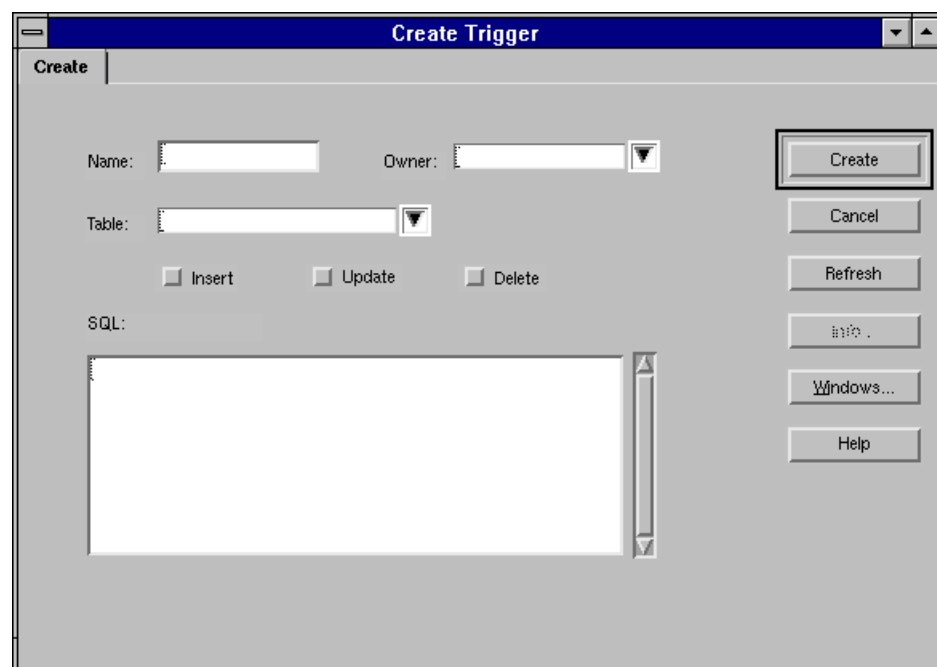


Figure 9-24: Create Trigger dialog box

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. (The default owner is dbo.)
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.

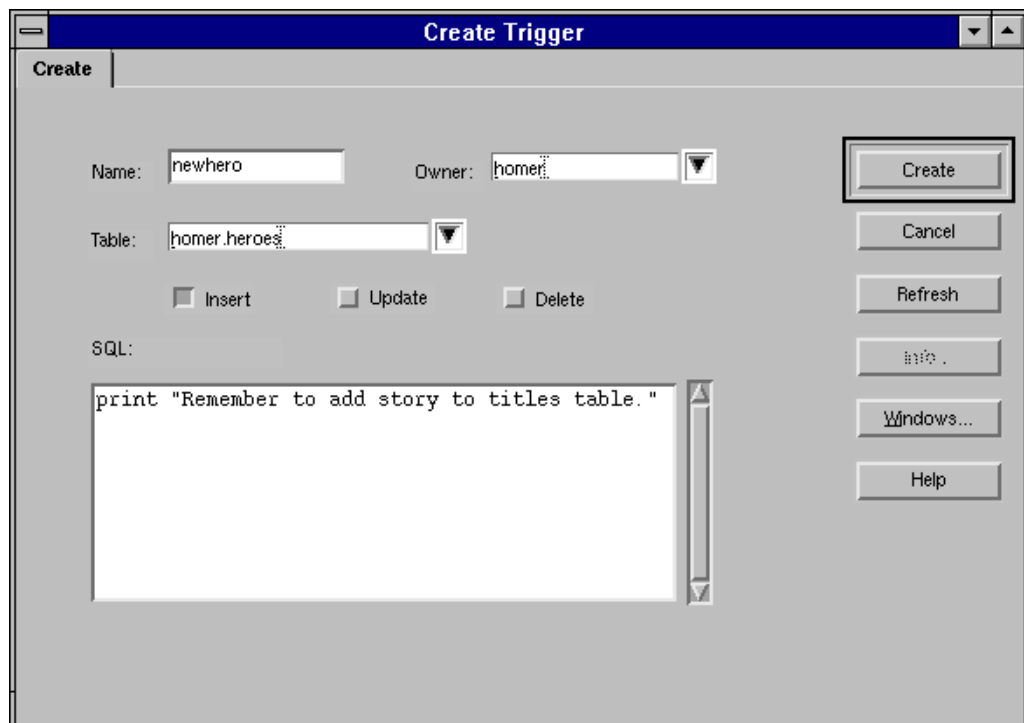


Figure 9-25: Creating the newhero trigger

Enterprise SQL Server Manager constructs the appropriate SQL query from the data you provide and creates the trigger shown in the following figure. The SQL text is editable.

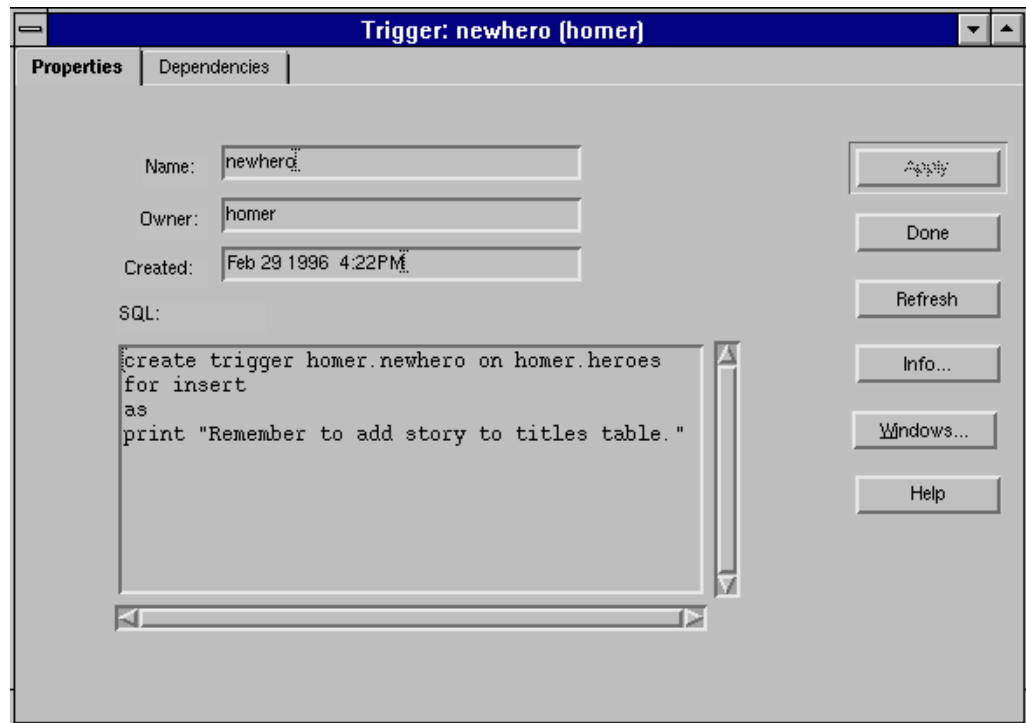


Figure 9-26: newhero trigger as created by Enterprise SQL Server Manager

Displaying Trigger Properties

To display the Properties tab:

1. Select the icon of the trigger.
2. From the Trigger menu, choose Properties.

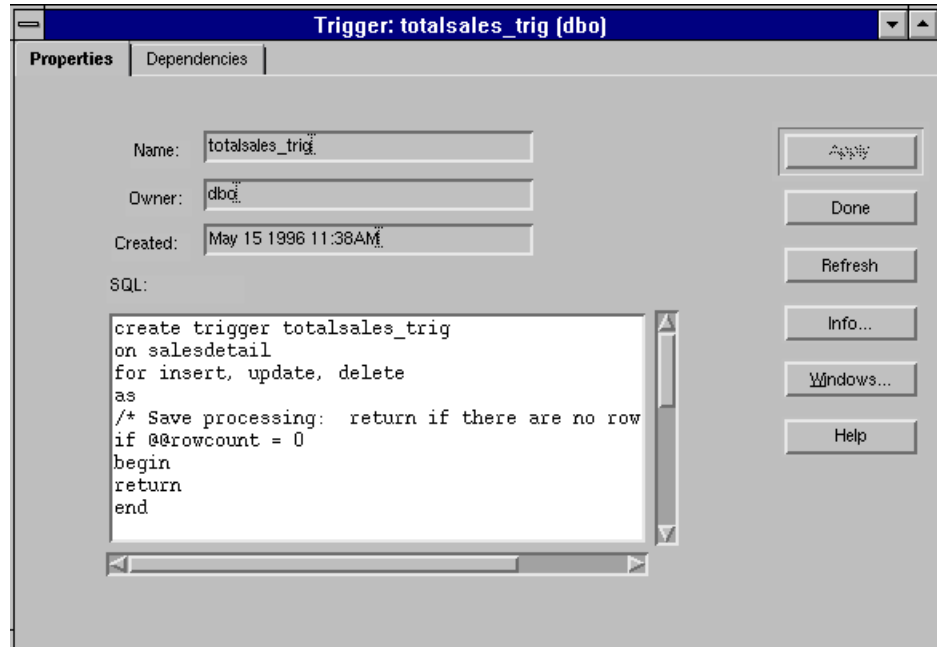


Figure 9-27: Trigger Properties tab

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.

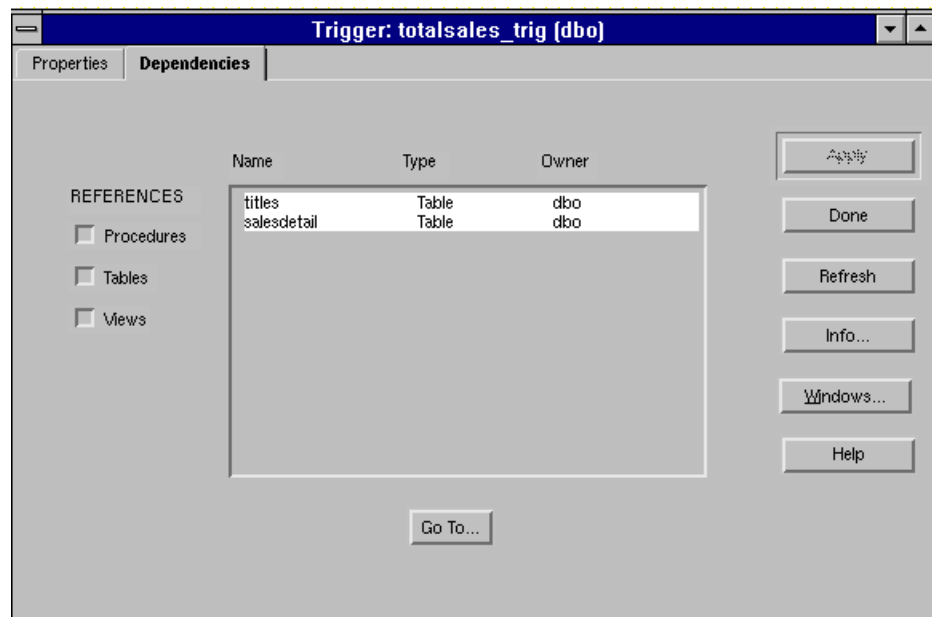


Figure 9-28: Trigger Dependencies tab

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.

	TME	ESSM	SQL Server
Required roles	any	schema	any

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.

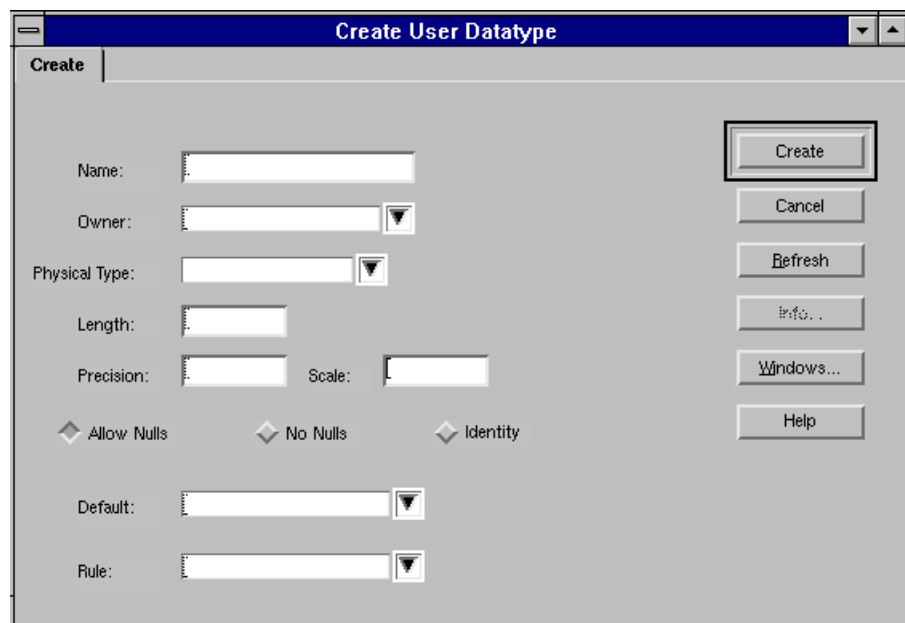


Figure 9-29: Create User Datatype dialog box

3. In the Name box, enter a name for the user datatype.
4. In the Owner box, enter the name of the owner.
5. From the Physical Type list, select a system datatype on which to base the user datatype.

6. If the datatype is *text*, *char*, *varchar*, *binary*, *varbinary*, *nchar*, or *nvarchar*, enter a maximum length for the datatype in the Length box.
7. 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.
8. 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.
9. 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.
10. 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.
11. To bind the user datatype to a rule, select a rule from the Rule list.
12. To bind the user datatype to a default, select a default from the Default list.
13. Click Create.

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.

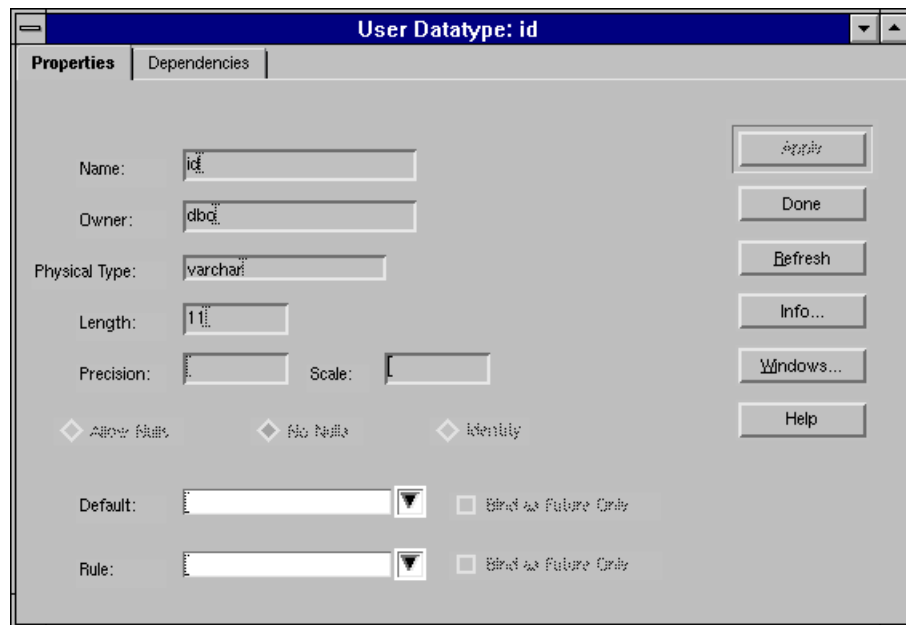


Figure 9-30: User Datatype Properties tab

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.

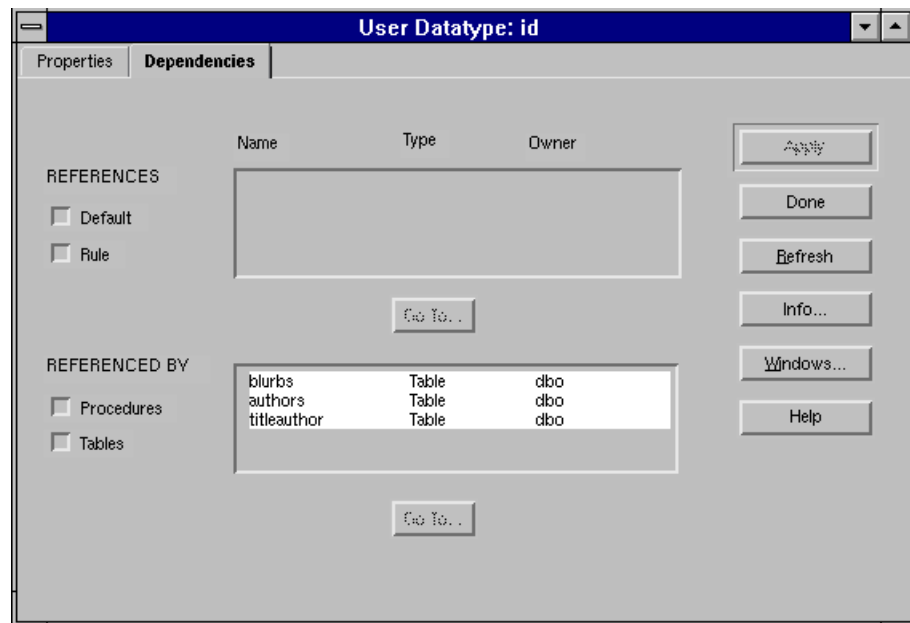


Figure 9-31: User Datatype Dependencies tab

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

	TME	ESSM	SQL Server
Required roles	any	schema	Database Owner

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.

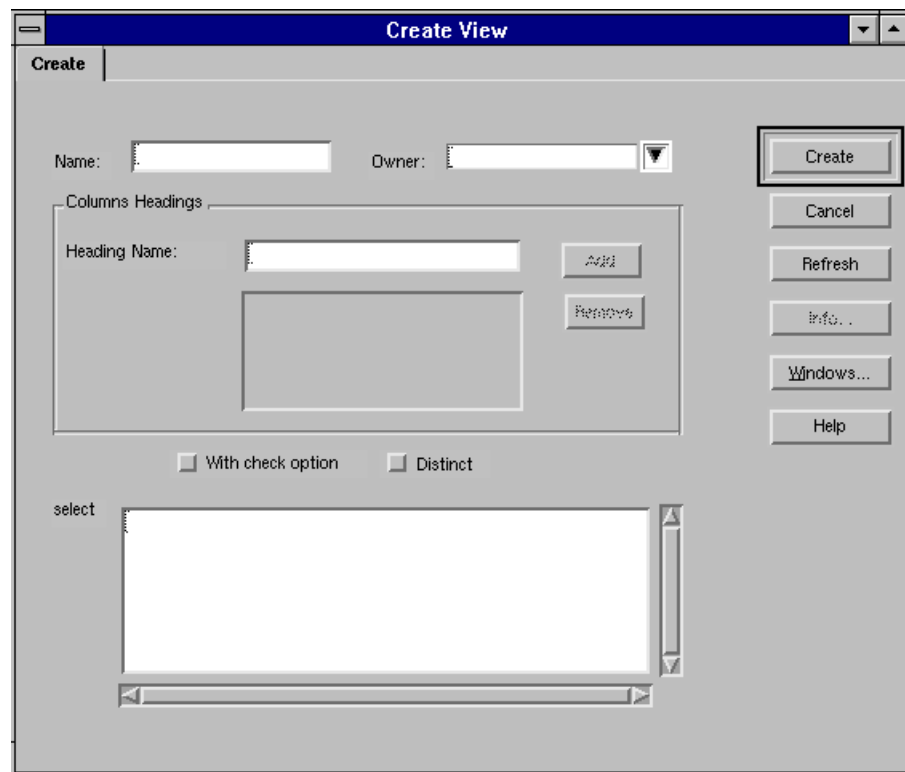


Figure 9-32: Create View dialog box

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 Enterprise 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*.

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:

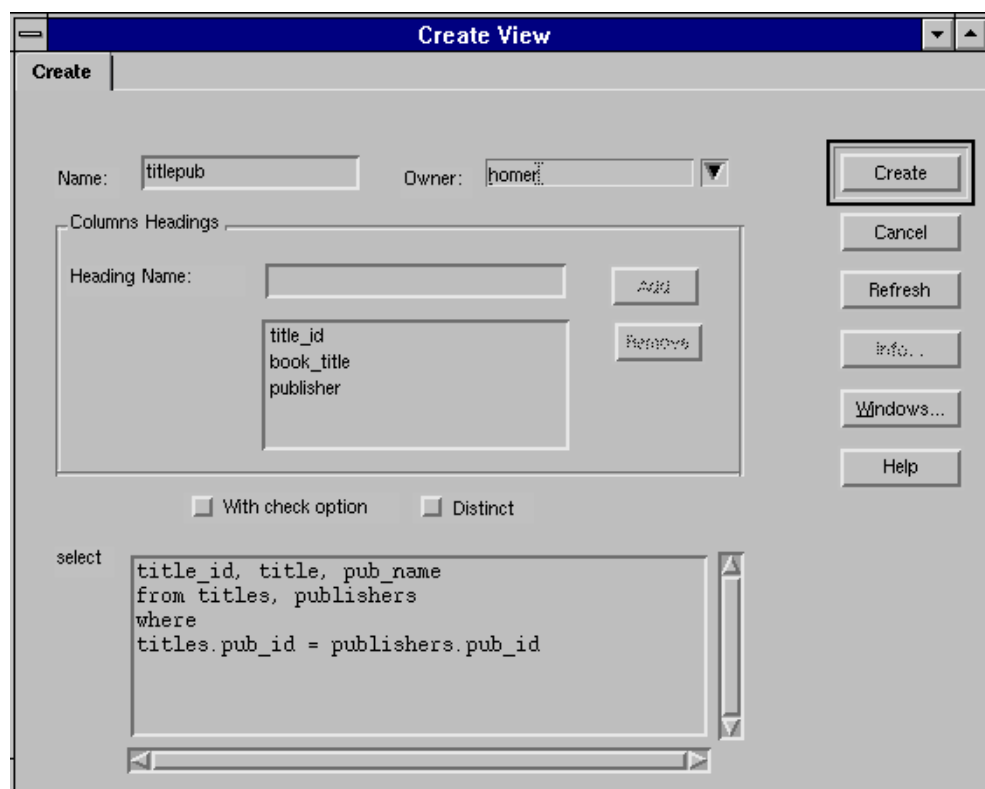


Figure 9-33: Creating the titlepub view

Enterprise SQL Server Manager constructs the appropriate SQL query from the data you provide and creates the view shown in the following figure. The SQL text is editable.

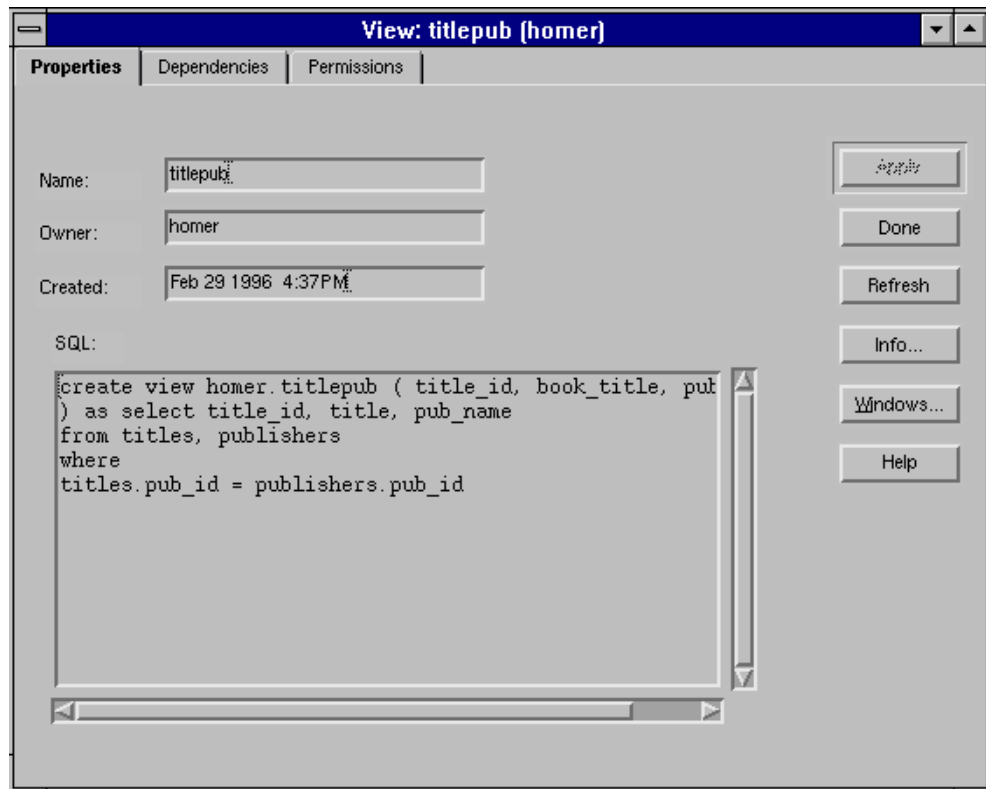


Figure 9-34: titlepub view as created by Enterprise SQL Server Manager

Note that Enterprise 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.

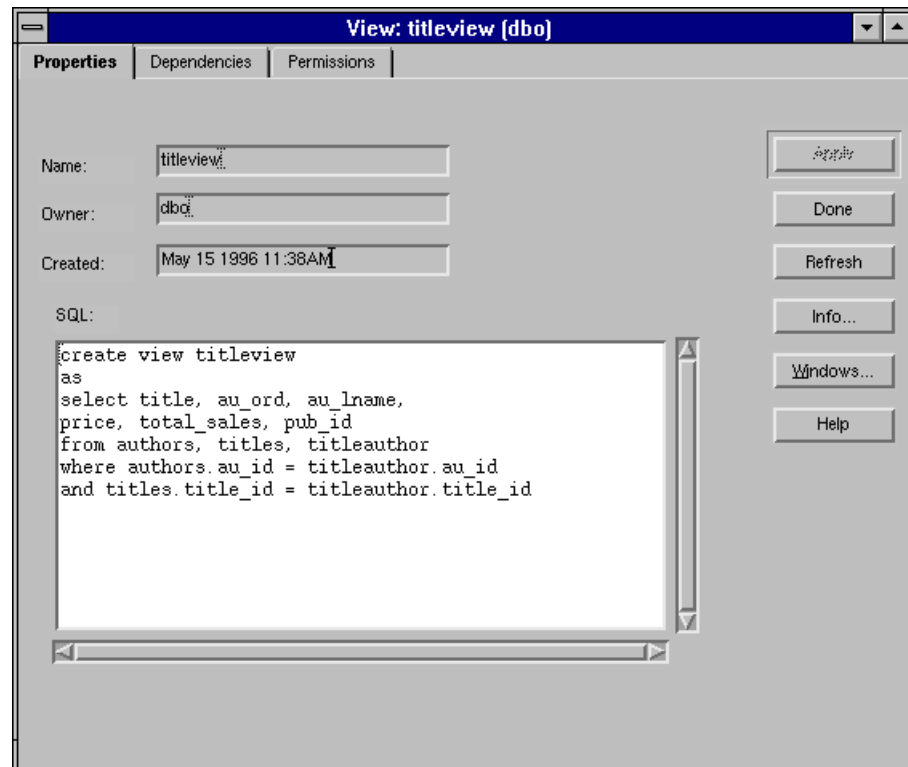


Figure 9-35: View Properties tab

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 9-65.
2. Click Dependencies. The display changes to the Dependencies tab.

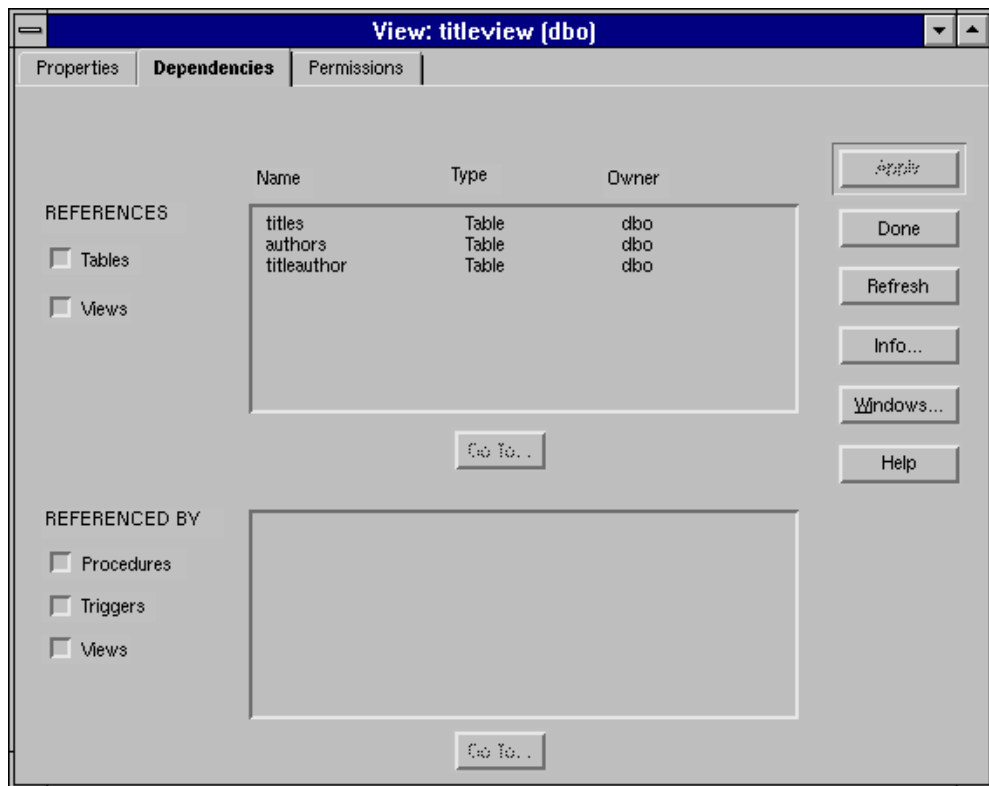


Figure 9-36: View Dependencies tab

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 9-65.
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).

	TME	ESSM	SQL Server
Required roles	any	security	System Administrator or Database Owner

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 9-65.
2. Select Permissions. The display changes to the Permissions tab.

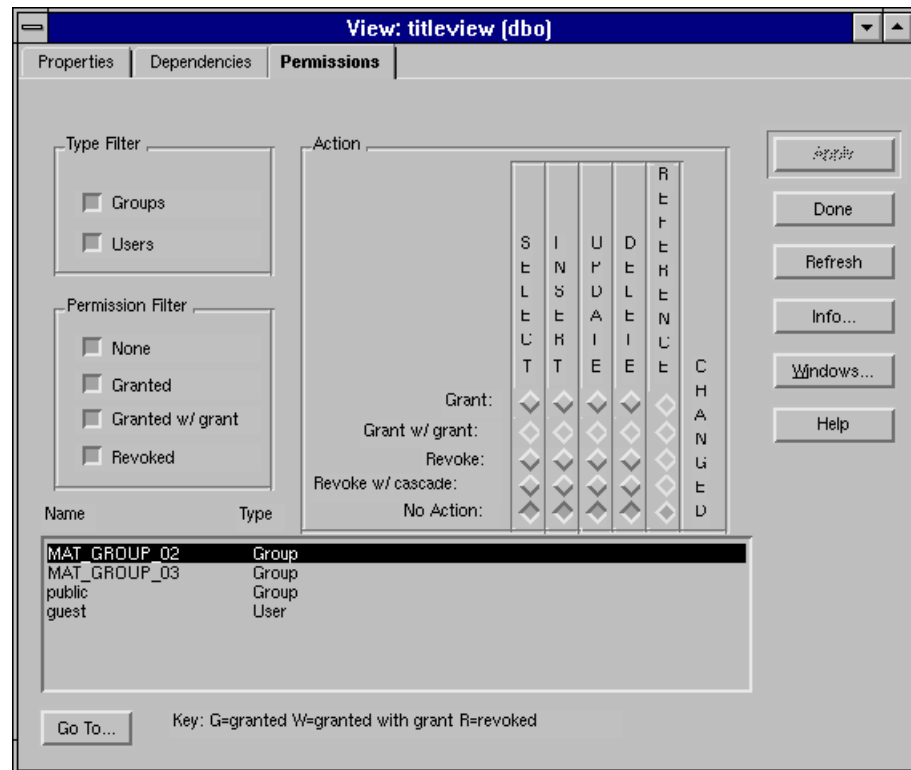


Figure 9-37: View Permissions tab

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 change a permission for a user or group, a check mark is added to the Changed column. 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. Enterprise 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**

Enterprise 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.

Shortcuts



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 9-65.
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 Enterprise SQL Server Manager to create and update user table definitions.

The chapter describes the following tasks:

- Creating and updating a table
- Copying table data
- 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
- Partitioning a table
- Creating referential constraints on a table
- Recompiling procedures and triggers
- Checking table consistency
- Updating statistics

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. Enterprise 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**.

Enterprise 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 about creating a default as a database object, see “Creating a Default” on page 9-11.

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 Constraints 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 about creating a table check constraint, see “Creating Check Constraints on a Table” on page 10-30. For information about creating a rule and binding it to a table column, see “Creating a Rule” on page 9-41 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 10-35.

Constraints Summary

The following table summarizes the types of constraints you can place on table data and lists how to implement them by using Enterprise 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

Procedure for Creating a Table

	TME	ESSM	SQL Server
Required roles	any	schema	Database Owner or a user in the database

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.

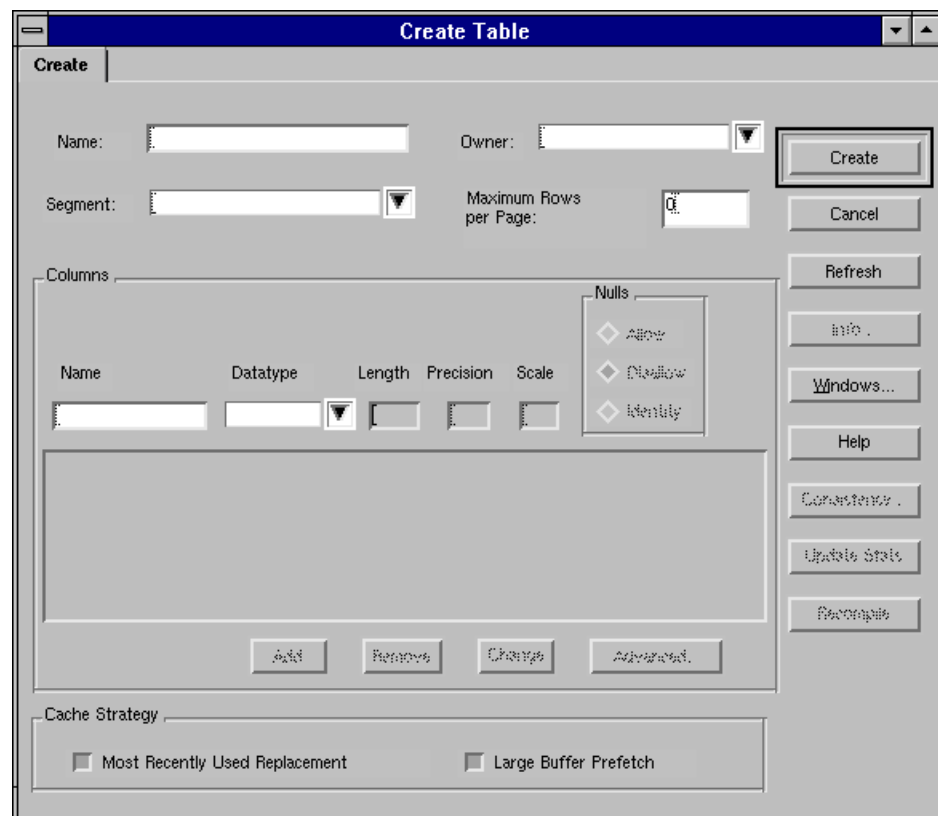




Figure 10-1: Create Table dialog box

3. In the Name box, enter a name for the table.
4. From the Owner list, choose an owner for the table. The default is “dbo”.
5. From the Segment list, choose a segment on which to store the table. The default is the default segment.

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. Values between 0 and 256 are valid.
7. Specify column definitions, as described in “Specifying Column Definitions” on page 10-8.
8. Optionally, create column constraints, as described in “Specifying Column Constraints” on page 10-9.

► *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 10-15.

9.  If you are connected to SQL Server release 11.0, 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 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 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



Select the Table container icon in the appropriate database, and select the Create toolbar button.



Click the right mouse button over the Table 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 Table.

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, Enterprise 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

Specify constraints on a column after you define it and add it to the Columns list, but before you create the table.

► 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 10-15.

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.

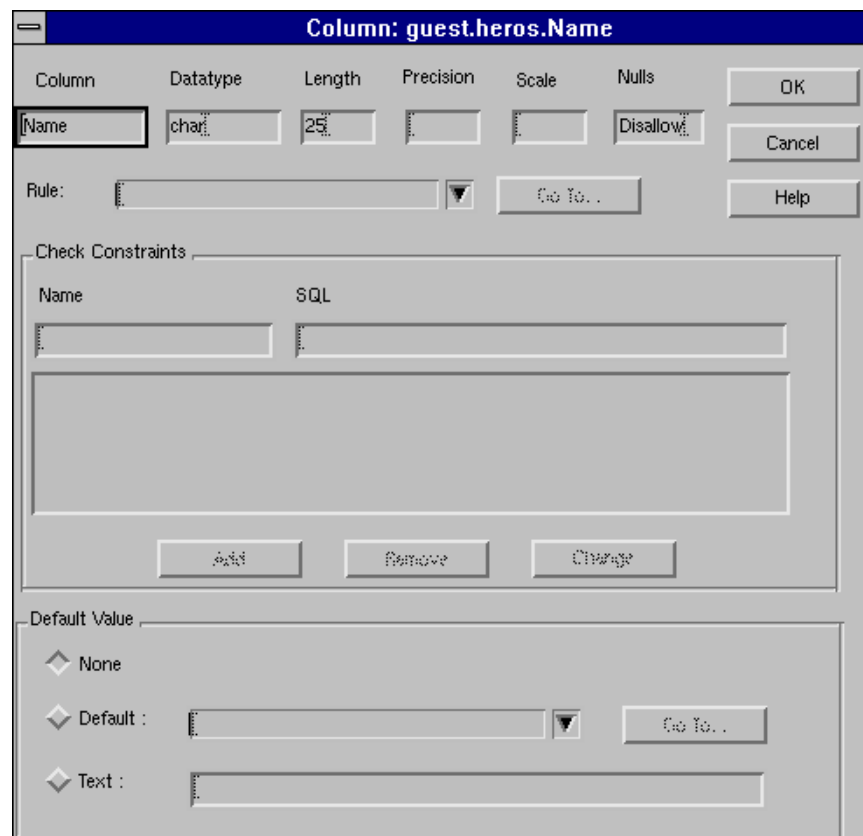


Figure 10-2: Table Column Advanced dialog box

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” on page 10-11.
 - 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.
7. In the Create Table dialog box, 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]")
```

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. To delete tables, you must have the following roles:

	TME	ESSM	SQL Server
Required roles	any	schema	table owner

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

	TME	ESSM	SQL Server
Required roles	any	any	any

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 SQL Server release 11.0, 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.

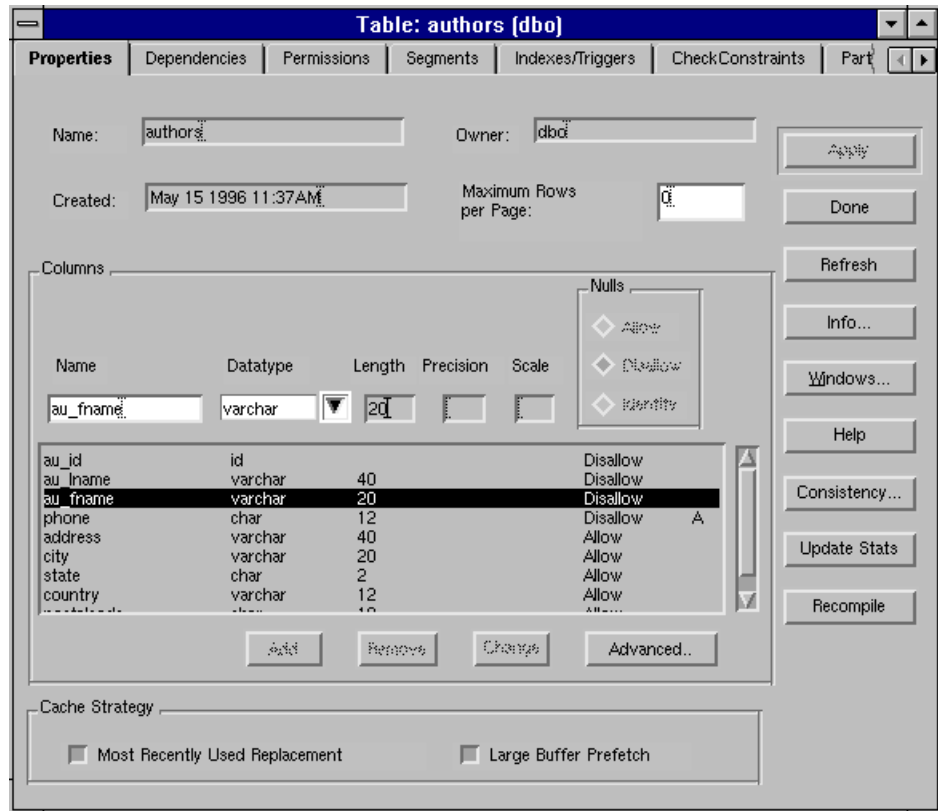


Figure 10-3: Table Properties tab

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 10-13.
2. For each column you add, follow the instructions in “Specifying Column Definitions” on page 10-16.
3. To create constraints on new columns, follow the instructions in “Specifying Column Constraints” on page 10-17.

► **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, Enterprise 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

Specify constraints on a column after you define it and add it to the Columns list, but before you apply the change. If a column has constraints, or a bound rule or default, an A is listed in the right-most column of the list of columns in the Table Properties tab. The A indicates that advanced column information is defined.

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:

The screenshot shows the 'Table Column Advanced' dialog box for the column 'Name' in the table 'guest.heros'. The dialog is titled 'Column: guest.heros.Name'. It features a table with columns for 'Column', 'Datatype', 'Length', 'Precision', 'Scale', and 'Nulls'. The 'Name' column is selected, and its properties are displayed: 'Name', 'char', '25', and 'Disallow'. Below this table are buttons for 'OK', 'Cancel', and 'Help'. A 'Rule:' dropdown menu is present, with a 'Go To...' button next to it. The 'Check Constraints' section contains a table with 'Name' and 'SQL' columns, and buttons for 'Add', 'Remove', and 'Change'. The 'Default Value' section has radio buttons for 'None', 'Default', and 'Text', with a 'Go To...' button next to the 'Default' radio button.

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” on page 10-19.
 - 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.
7. In the Properties tab, 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]")
```

Modifying a Column's Datatype

After you create a column, you can change the datatype through the command line. Use the `ssetcolumn` command.



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.



Setting Cache Strategy

If you are managing SQL Server release 11.0, 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 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 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.

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 10-13.
2. Click Dependencies. The display changes to the Dependencies tab.

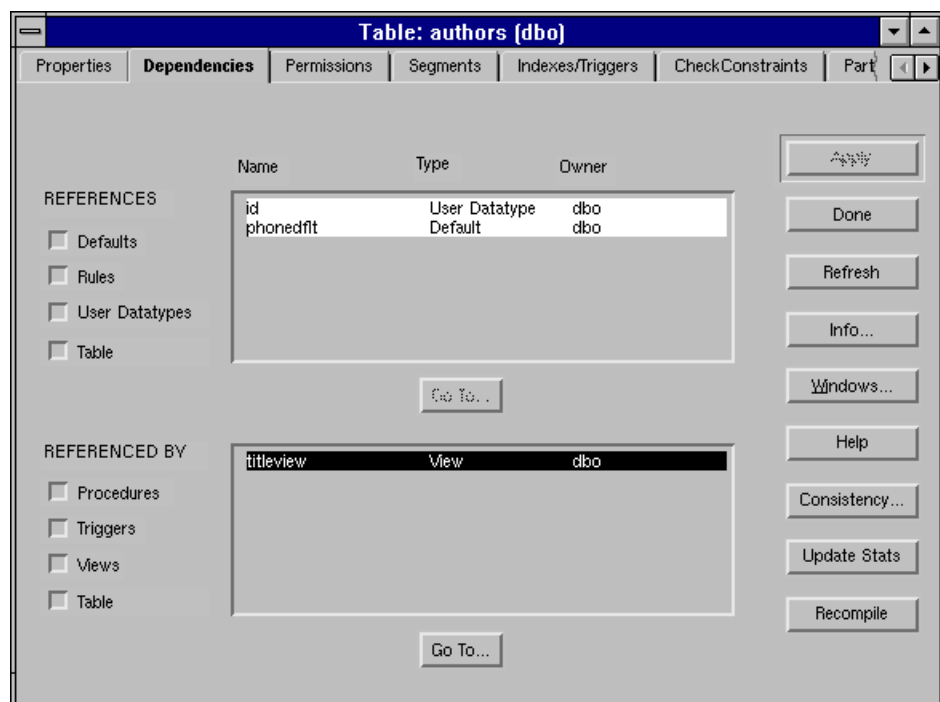


Figure 10-5: Table Dependencies tab

3. To filter the display of object types that the table references, select the Defaults, Rules, User Datatypes, or Tables buttons in the References list.
4. To filter the display of object types referenced by the table, select the Procedures, Views, Triggers, or Tables buttons in the Referenced By list.

To navigate to objects that reference a table or that the table references, 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).

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 10-13.
2. Click Permissions. The display changes to the Permissions tab.

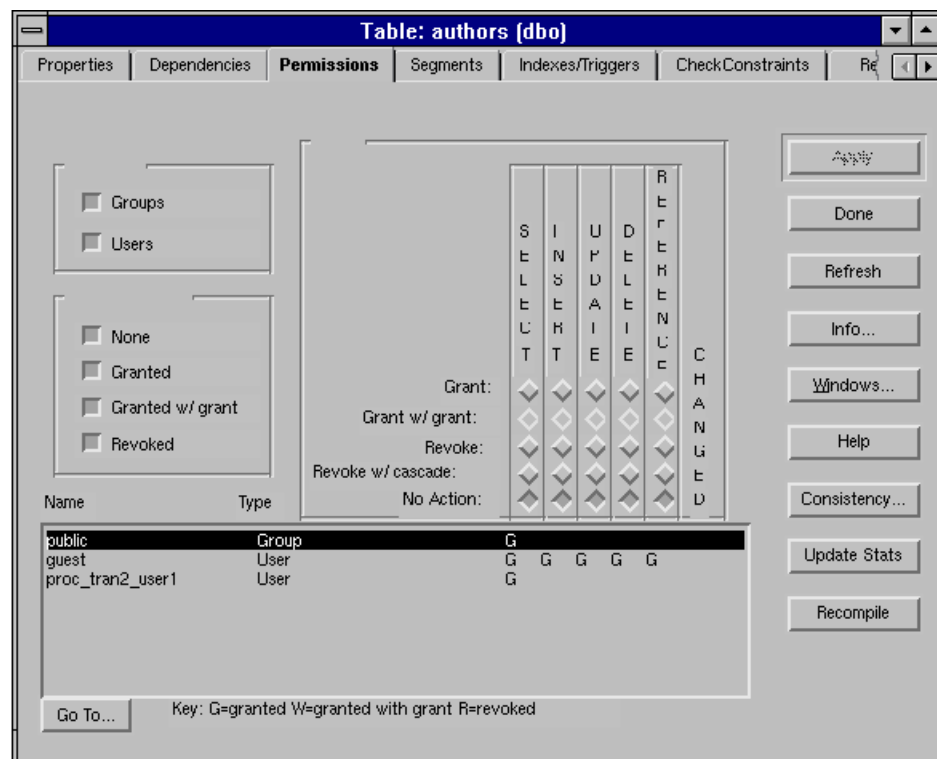


Figure 10-6: Table Permissions tab

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. Enterprise 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**

Enterprise 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 10-13.
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 10-13.
2. Click Segments. The display changes to the Segments tab.

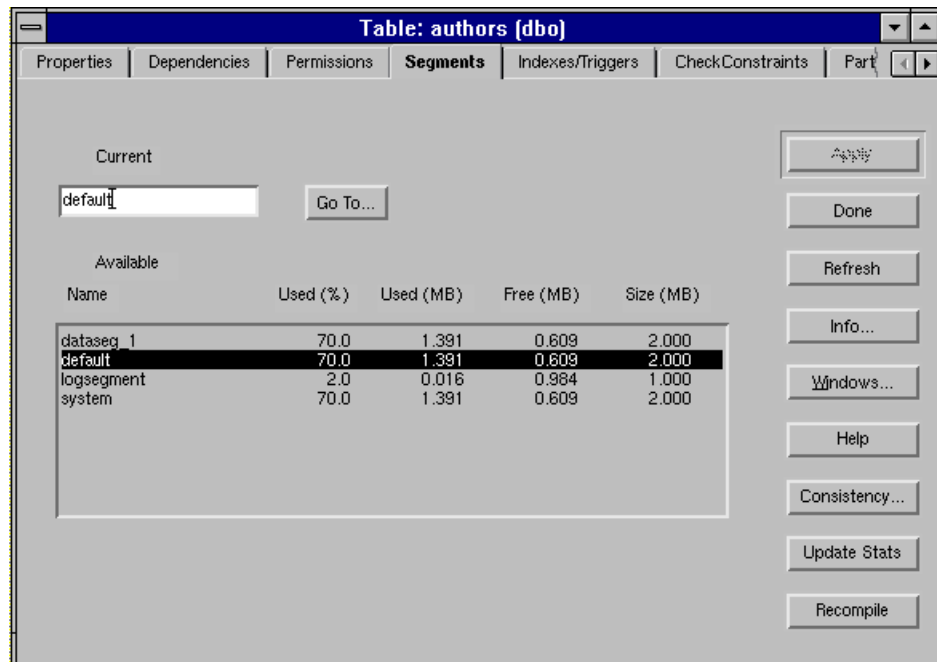
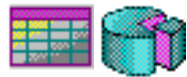


Figure 10-7: Table Segments tab

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 10-13.
2. Click Segments. The display changes to the Segments tab.
3. Select a segment in the Available list.
4. Click Go To. The Properties tab for the segment you selected is displayed.

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 10-13.
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 the Triggers button. To display only triggers associated with the table, clear the Indexes button.

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 10-13.
2. Select Indexes/Triggers. The display changes to the Indexes/Triggers tab.
3. To display only indexes associated with the table, clear the Triggers button. To display only triggers associated with the table, clear the Indexes button.
4. To display the Properties tab for an index or trigger, select an object in the list box; 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 10-13.
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 9-20 or “Creating a Trigger” on page 9-49.

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 about 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 10-13.
2. Click Check Constraints. The display changes to the Check Constraints tab:

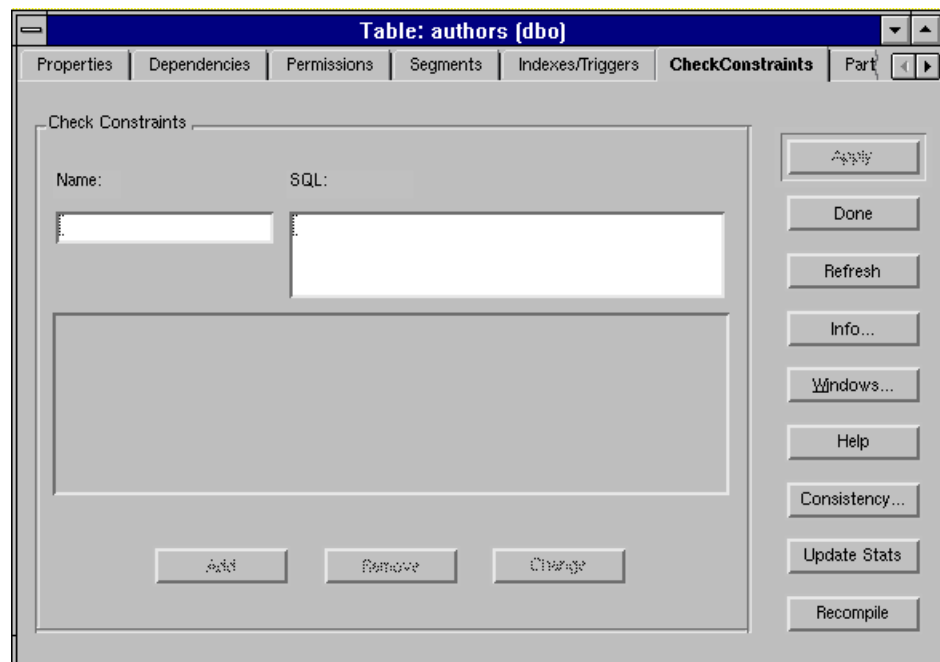


Figure 10-9: Table Check Constraints tab

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 SQL Server release 11.0, 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 10-13.
2. Click Partition. The display changes to the Partition tab:

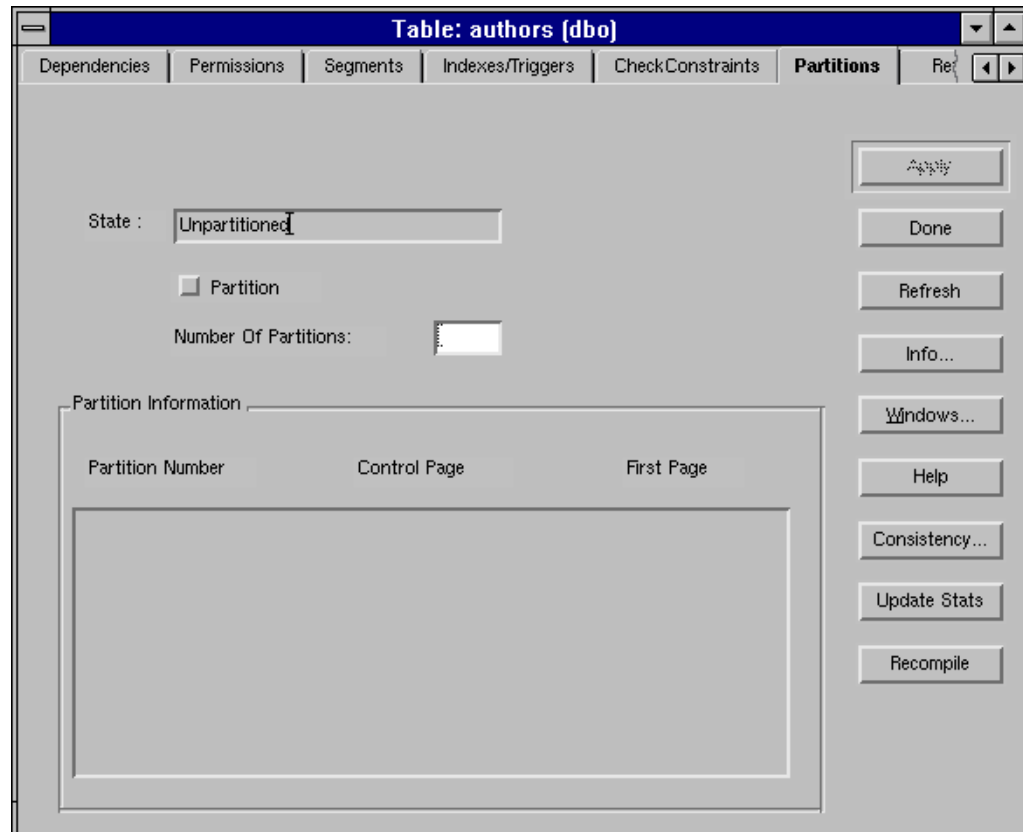


Figure 10-10: Table Partition tab

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 10-13.
2. Click Partition. The display changes to the Partitions tab.

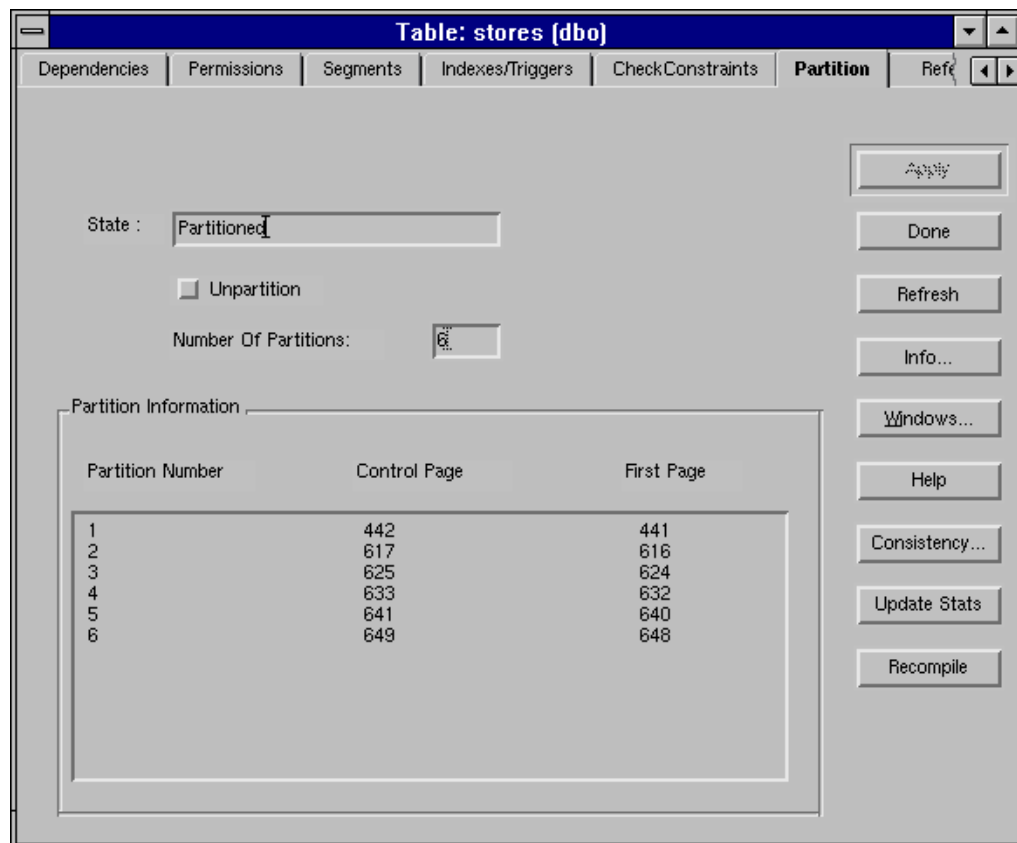


Figure 10-11: Table Partition tab

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.

To create a referential constraint:

1. Display the Table Properties tab, as described in “Displaying Table Properties” on page 10-13.
2. Click Referential Constraints. The display changes to the Referential Constraints tab.

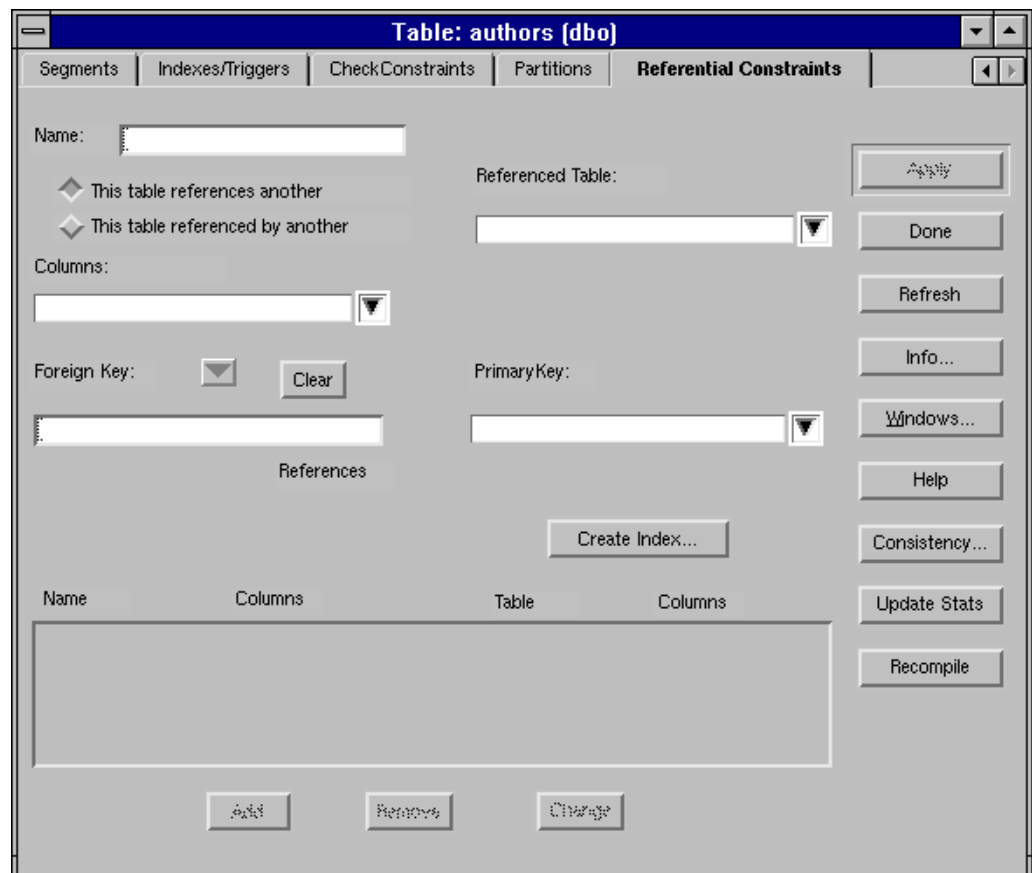


Figure 10-12: Table Referential Constraints tab

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 10-37. 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 10-38.

This Table References Another

When you select **This Table References Another**, the **Referential Constraints** tab has the controls shown in Figure 10-12 on page 10-36.

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, select it, then 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:

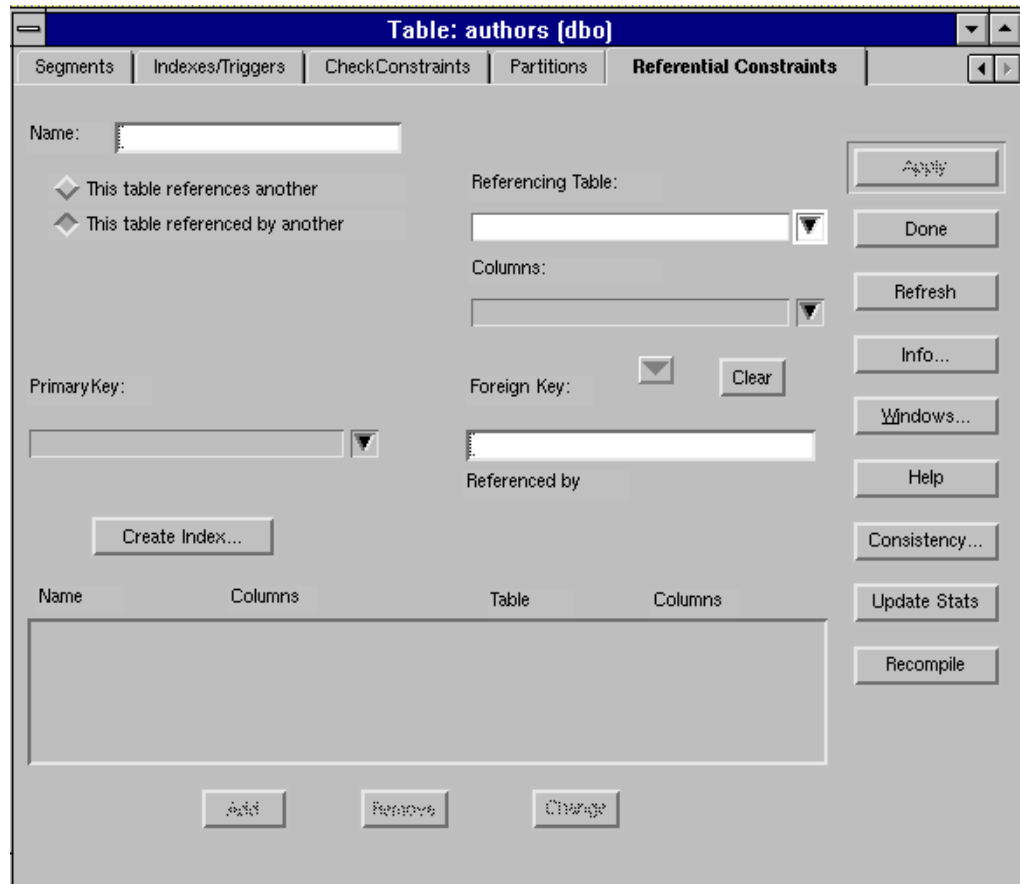


Figure 10-13:Referential Constraints tab for referenced table

To specify a constraint:

1. From the Referencing Table list, select the table for which SQL Server validates inserts against the current table.
2. From the Columns list, select the columns in the referencing table that make up the foreign key:

Select a column.

Click the down arrow to add it to the Foreign Key box.

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, select it, then 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.
3. Click OK.

Shortcut



From any Table dialog tab, select the Recompile button.

Checking Table 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 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.

	TME	ESSM	SQL Server
Required roles	any	schema	Database Owner; For the Fix text and Reindex options, you must also be the table owner .

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.

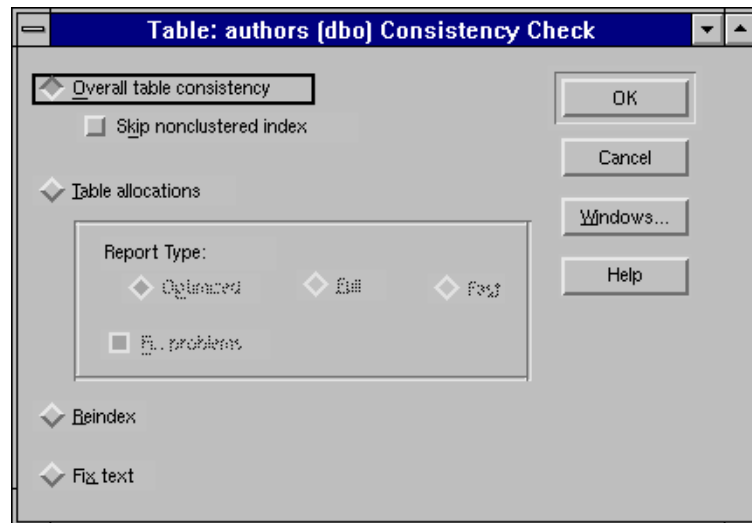


Figure 10-14: Table Consistency dialog box

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 10-42.
4. Click OK. SQL Server executes the `dbcc` command. When it completes, Enterprise 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

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.

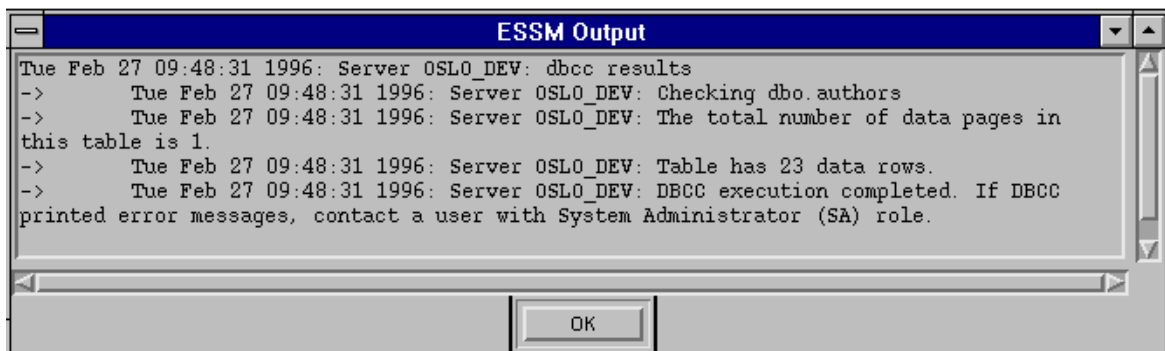


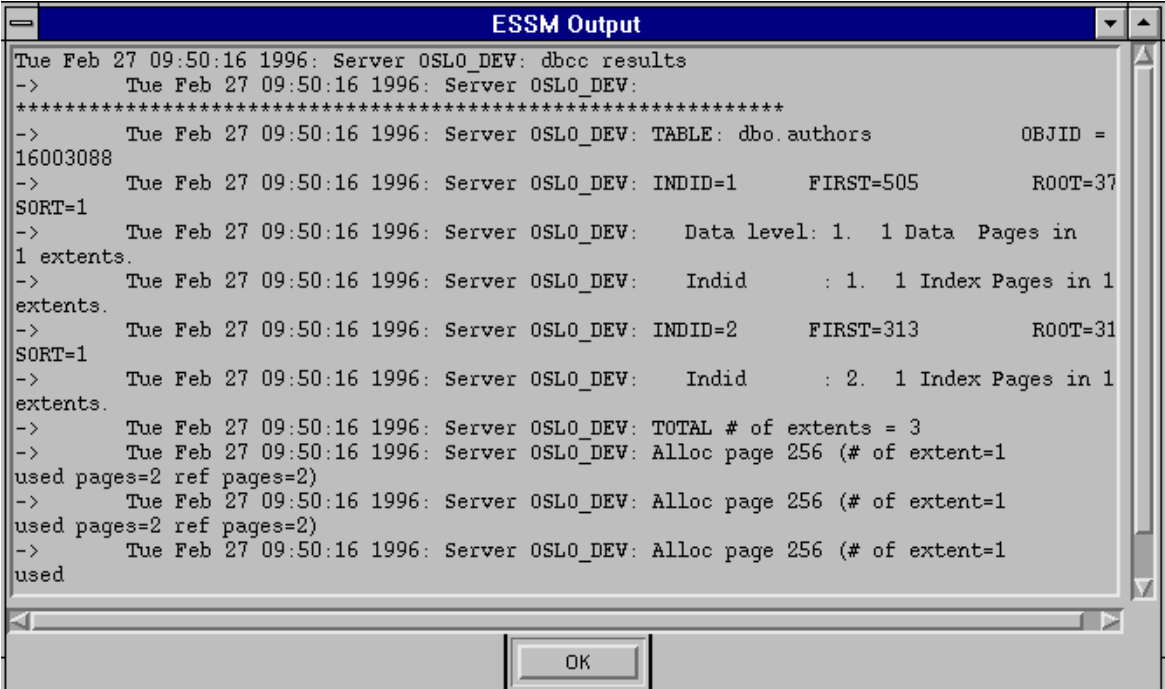
Figure 10-15: Table consistency check output—overall consistency option

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. Enterprise 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.



```

ESSM Output
Tue Feb 27 09:50:16 1996: Server OSLO_DEV: dbcc results
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV:
*****
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV: TABLE: dbo.authors          OBJID =
16003088
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV: INDID=1          FIRST=505          ROOT=37
SORT=1
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV: Data level: 1.  1 Data Pages in
1 extents.
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV: Indid           : 1.  1 Index Pages in 1
extents.
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV: INDID=2          FIRST=313          ROOT=31
SORT=1
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV: Indid           : 2.  1 Index Pages in 1
extents.
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV: TOTAL # of extents = 3
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV: Alloc page 256 (# of extent=1
used pages=2 ref pages=2)
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV: Alloc page 256 (# of extent=1
used pages=2 ref pages=2)
-> Tue Feb 27 09:50:16 1996: Server OSLO_DEV: Alloc page 256 (# of extent=1
used
  
```

Figure 10-16: 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.

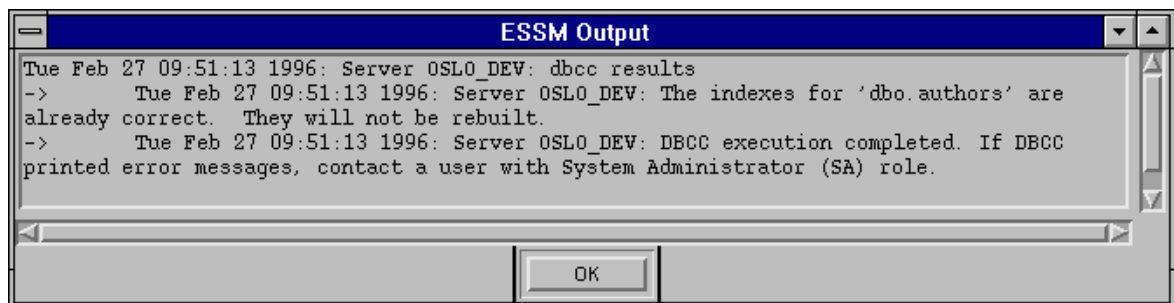


Figure 10-17: Table consistency check output—reindex option

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.

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.

Copying Table Data

You can distribute (copy) the data in a table as part of a profile distribution. If table data copy is enabled, the data in all tables in the profile is distributed. For more information, see “Distributing Table Data” in Chapter 12, “Managing ESSM Profiles”.

11

Enterprise SQL Server Manager Profile Managers

What's in This Chapter

The previous chapters in this manual focused on how to use the SQL Server window to manage objects in a single SQL Server installation. This chapter and the following chapters discuss the enterprise part of Enterprise SQL Server Manager, that is, managing many SQL Server installations at once.

This chapter describes:

- Profile management concepts
- Managing profile managers
- Using a profile manager window

Profile Management Concepts

Profile management is a powerful tool that allows you to distribute SQL Server and database objects to many SQL Servers across the enterprise in a single step.

The ability to create and distribute profiles is the key to managing your enterprise. Without profile distribution, you can manage only one SQL Server installation at a time. With profile distribution, you can manage hundreds of SQL Server installations at a time.

A **profile** is a description or model of an object or group of objects of the same type in SQL Server that you want to distribute (copy) to other SQL Server installations. Subscribers are the SQL Server installations, databases, and other profile managers to which you want to distribute the objects. You use profile managers to create profiles and set up relationships with subscribers.

When you distribute a profile to a subscriber, it is implemented at the target (recipient) SQL Server or database. This allows you to create an object on one SQL Server and then distribute the object definition to as many other SQL Server installations or databases as you are managing with Enterprise SQL Server Manager.

You also use profile managers to manage default and validation policy, a method of enforcing rules when objects are created or changed.

Figure 11-1 illustrates profile management as described in this chapter. The SQL Server Koko is **associated** with the profile manager Koko_ProfMgr. The database *pubs2* on Koko populates a database profile. That profile is distributed to subscribing SQL Servers Eng1, Eng2, and to a subscribing SQL Server Profile Manager, Agra_ProfMgr. Also, because SQL Server Agra is associated with Agra_ProfMgr, it automatically becomes a subscriber to Koko_ProfMgr and *pubs2* is distributed to it. Now, the database *pubs2* on the subscribers is identical to the database *pubs2* on Koko.

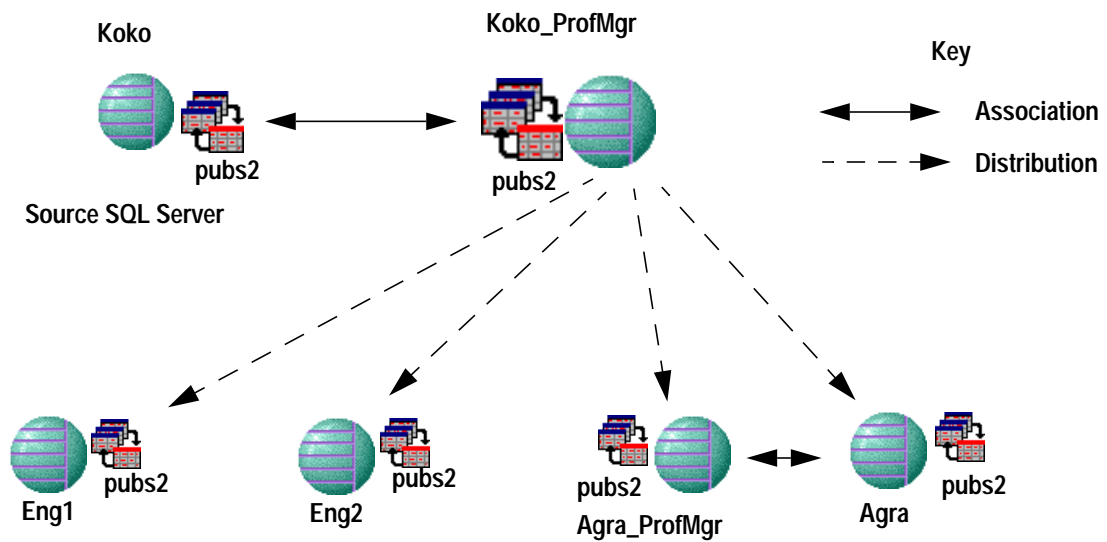


Figure 11-1: Copies of a profile are distributed to subscribers

► **Note**

Some aspects of the Enterprise SQL Server Manager implementation of profile management differ from the generic description in Tivoli documentation. Unless otherwise specified, all discussion of profiles and profile management in this chapter applies only to profiles as implemented by Enterprise SQL Server Manager and managed from a SQL Server Profile Manager window or a Database Profile Manager window.

Enterprise SQL Server Manager Profile Managers

An Enterprise SQL Server Manager profile manager lets you create and manage SQL Server profiles, database profiles, and subscribers to those profiles. You can also create, change, and delete SQL Server objects from within profiles. Managing profile managers involves:

- Registering profile managers as managed resources
- Creating profile managers
- Deleting profile managers
- Moving profile managers

Enterprise SQL Server Manager profiles are managed by two types of profile managers: SQL Server Profile Managers and Database Profile Managers.



SQL Server Profile Manager



Database Profile Manager


Figure 11-2: Profile Manager icons

SQL Server Profile Managers

Each SQL Server Profile Manager is associated with a managed SQL Server. The profiles in the profile manager refer to the SQL Server objects in the associated SQL Server. SQL Server Profile Managers allow you to distribute the profiles to subscribing SQL Servers and SQL Server Profile Managers. Through the distribution process, the profiles are copied to the subscribers.

SQL Server Profile Managers support the following profile types:

Table 11-1: SQL Server Profile Manager profiles

Profile Type	Profile operations
SQLServerProfile	Modify SQL Server configuration (you cannot create or delete SQL Servers using profile management)
SQLDumpDeviceProfile	Create, delete dump devices
SQLDbDeviceProfile	Create, modify, and delete database devices
SQLDatabaseProfile	Create databases, modify database options, delete databases
SQLLoginProfile	Create, modify, and delete logins
SQLRemoteServerProfile	Create, modify, and delete remote servers and remote logins
 SQLCacheProfile	Create, modify, and delete named caches

Database Profile Managers

Each Database Profile Manager is associated with a database in a managed SQL Server. The profiles in the profile manager refer to the database objects in the associated database. Database Profile Managers allow you to distribute the profiles to subscribing databases and Database Profile Managers.

Database Profile Managers support the following profile types:

Table 11-2: Database Profile manager profile types

Profile type	Operations performed
SQLDatabaseProfile	Modify database options
SQLSegmentProfile	Create, modify, and delete segments
SQLGroupProfile	Create, modify, and delete group names
SQLUserProfile	Create, modify, and delete database users
SQLDefaultProfile	Create, modify, and delete defaults
SQLRuleProfile	Create, modify, and delete rules
SQLDataTypeProfile	Create, modify, and delete user datatypes
SQLTableProfile	Create, modify, and delete tables
SQLIndexProfile	Create, modify, and delete indexes
SQLViewProfile	Create, modify, and delete views
SQLProcedureProfile	Create, modify, and delete procedures
SQLTriggerProfile	Create, modify, and delete triggers

► **Note**

Successful distribution of a SQL Server or database profile can depend on the successful distribution of other profiles. See Appendix C, “SQL Server Profile Operations,” for information about distribution dependencies.

Creating Profile Managers

Profile managers are Tivoli managed resources. Before you can create one, you must add the resource type to the list of current resources in the policy region in which you want to create the profile manager. You may have already added the resource types `SQLServerProfileMgr` and `SQLDatabaseProfileMgr` to the list of managed resources in your policy regions when you installed Enterprise SQL Server Manager. If you did not do so, see “Configuring Policy Regions for Managed SQL Server Resources” in Chapter 2, “Setting Up the Tivoli Management Environment for ESSM.” Follow the instructions for adding resource types to a policy region. After you add the resource types for Enterprise SQL Server Manager profile managers, you can create profile managers.

Creating a SQL Server Profile Manager

	TME	ESSM	SQL Server
Required roles	senior	any	none

Create a SQL Server Profile Manager in a policy region window.

1. From the Create menu of a policy region window, choose SQLServerProfileMgr. The Create SQL Server Profile Manager dialog box opens.

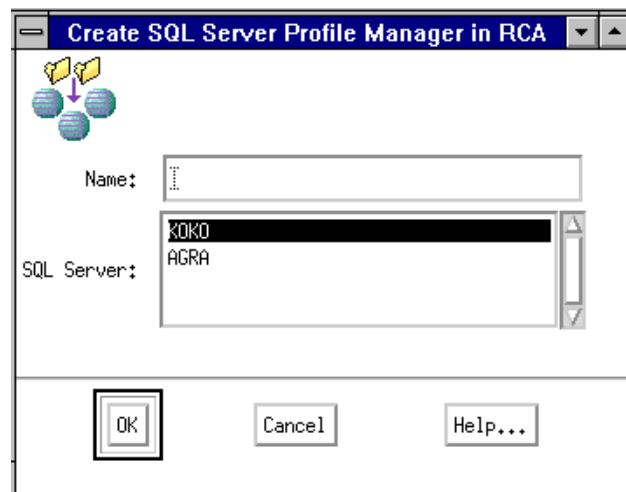


Figure 11-3: Create SQL Server Profile Manager dialog box

2. In the Name box, type a name for the profile manager.

► **Note**

Once you create the profile manager, you cannot change its name, so be sure you enter the name correctly before you complete this procedure.

3. In the SQL Server list box select a name from the list.
4. Click OK.

Enterprise SQL Server Manager creates the SQL Server Profile Manager, and its icon appears in the policy region window.

► Note

Only one SQL Server Profile Manager can be associated with a given SQL Server. (Multiple Database Profile Managers can be associated with a SQL Server; each Database Profile Manager is associated with a different database in the SQL Server.)

Creating a Database Profile Manager

	TME	ESSM	SQL Server
Required roles	senior	any	none

You create a Database Profile Manager in a policy region window.

1. From the Create menu of a policy region window, choose SQLDatabaseProfileMgr. The Create Database Profile Manager dialog box opens.

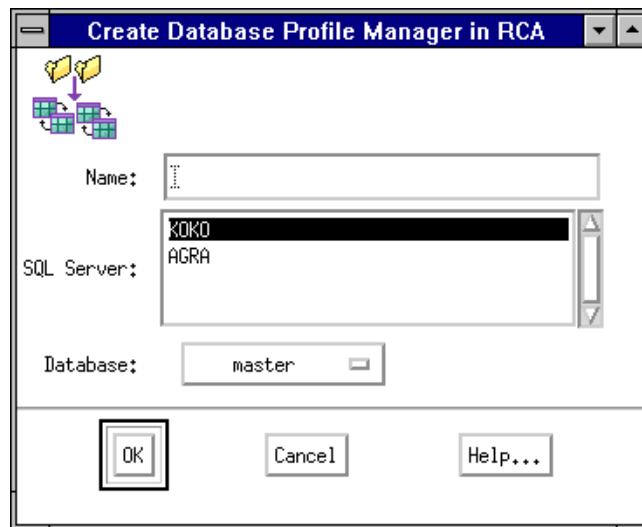


Figure 11-4: Create Database Profile Manager dialog box

2. In the Name box, type a name for the profile manager.
3. In the SQL Server box select a SQL Server from the list.

► **Note**

Once you create the profile manager, you cannot change its name, so be sure you entered the name correctly before you complete this procedure.

4. Select a name from the Database drop-down list.
5. Click OK.

Enterprise SQL Server Manager creates the database profile manager, and its icon appears in the policy region window.

► **Note**

Only one Database Profile Manager can be associated with a given database.

Moving Profile Managers

	TME	ESSM	SQL Server
Required roles	admin, senior, or super	none	none

You can move a profile manager from one policy region to another. The profile manager you are moving must be a valid managed resource type in the destination policy region. To move a profile manager to a new policy region:

1. In the policy region window, select the profile manager you want to move.
2. Hold down the Shift key and drag the icon onto a policy region icon or open policy region window. Enterprise SQL Server Manager moves the profile manager to the new policy region.

You can move more than one profile manager at a time by making a group selection.

Deleting Profile Managers

Before you can delete a profile manager, you must remove all subscribers to that profile manager. For information about removing subscribers, see “Removing Subscribers from a SQL Server Profile Manager” or “Removing Subscribers from a Database Profile Manager” in Chapter 12, “Managing ESSM Profiles.”

You must also delete all profiles that you created in this profile manager. You do not need to delete copies of profiles, that is, profiles that were distributed to this profile manager from another profile manager. For information about deleting profiles, see “Deleting a Profile” in Chapter 12, “Managing ESSM Profiles.”

	TME	ESSM	SQL Server
Required roles	admin, senior, or super	none	none

To delete a profile manager from a policy region:

1. In a policy region window, select the profile manager you want to delete.
2. From the Edit menu, choose Delete. A confirmation dialog box is displayed, asking if you want to proceed.



Figure 11-5: Delete profile manager confirmation dialog box

3. Confirm the deletion.

Profile Manager Window

Profile manager windows show all profiles in and subscribers to that profile manager. Profile manager windows use the Voyager format. Most of the components of a Profile Manager window are the same as those described in Chapter 3, “Enterprise SQL Server Manager Windows and Dialog Boxes.” See that chapter for information about how to use the window and common menus and other elements. This section describes aspects of profile manager windows that differ from the SQL Server window. Unless specifically stated, SQL Server Profile Manager windows and Database Profile Manager windows have the same menus and dialog boxes.

Opening a Profile Manager

To open a profile manager, in a policy region window, choose Open from the popup menu of a profile manager icon.

Figure 11-7 illustrates a SQL Server Profile Manager window.

Figure 11-8 illustrates a Database Profile Manager window.

Shortcuts

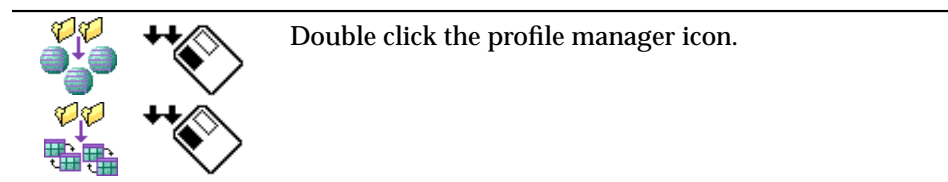


Figure 11-6: Shortcuts for opening a profile manager

► **Note**

When you first create a SQL Server Profile Manager or Database Profile Manager the Profiles container has a plus (+) sign next to it, but there are no profiles defined yet, so you cannot expand the hierarchy. Your first task in a profile manager would be to create profiles or add subscribers.

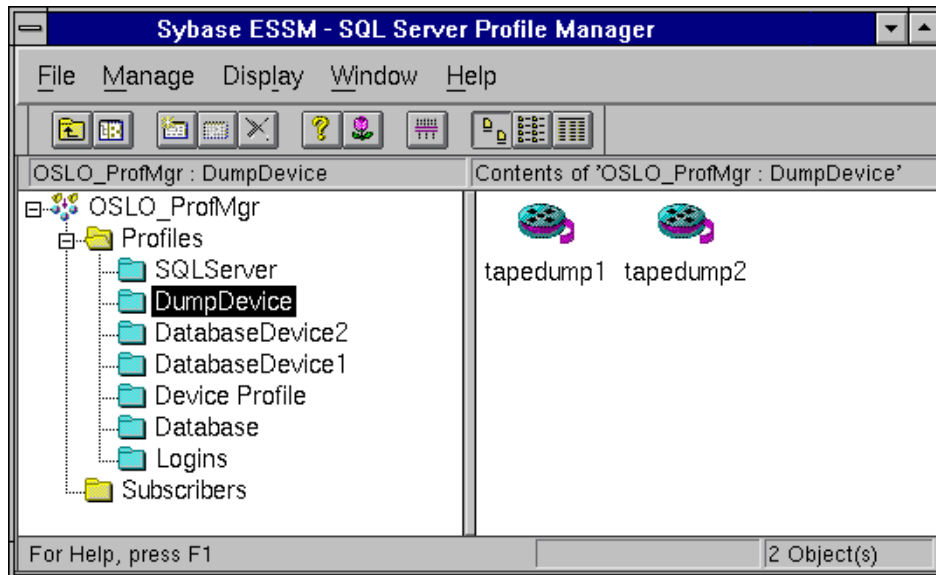


Figure 11-7: SQL Server Profile Manager window

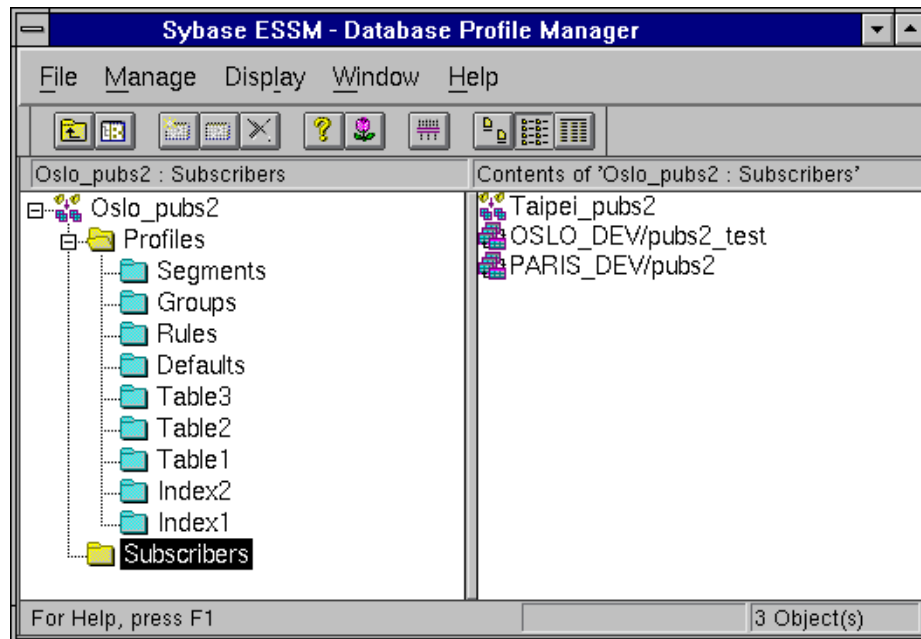


Figure 11-8: Database Profile Manager window

Left Pane of a Profile Manager Window

The left pane of a profile manager window has profiles container and a subscriber container. When you add a profile to the profile manager, a container for that type of profile is placed within the main profiles container. Profiles are listed by profile type. If you add more than one profile of the same type, the profiles are listed in alphabetical order within that type.

The first time you open a profile manager window, the Profiles container is empty. You must create profiles to populate it. For information about how to create a profile, see “Creating a Profile” in Chapter 12, “Managing ESSM Profiles.”

The Right Pane of the Profile Manager Window

The right pane of the Voyager window always displays the contents of the container that is selected in the left pane.

There are two containers that do not display contents in the right pane. The `SQLServerProfile` in a SQL Server Profile Manager and the `SQLDatabaseProfile` in a Database Profile Manager cannot be populated, so they do not display contents. If you select one of these profiles in the left pane, a single icon is displayed in the right pane.

If you double-click a profile icon in the right pane, the right pane displays the contents of that profile.

You can open the source SQL Server for a subscriber by choosing Open on the popup menu of a subscriber.

Subscribers Icon

If you select the Subscribers container in the left pane, the subscribers in the profile manager appear in the right pane (Figure 11-9).

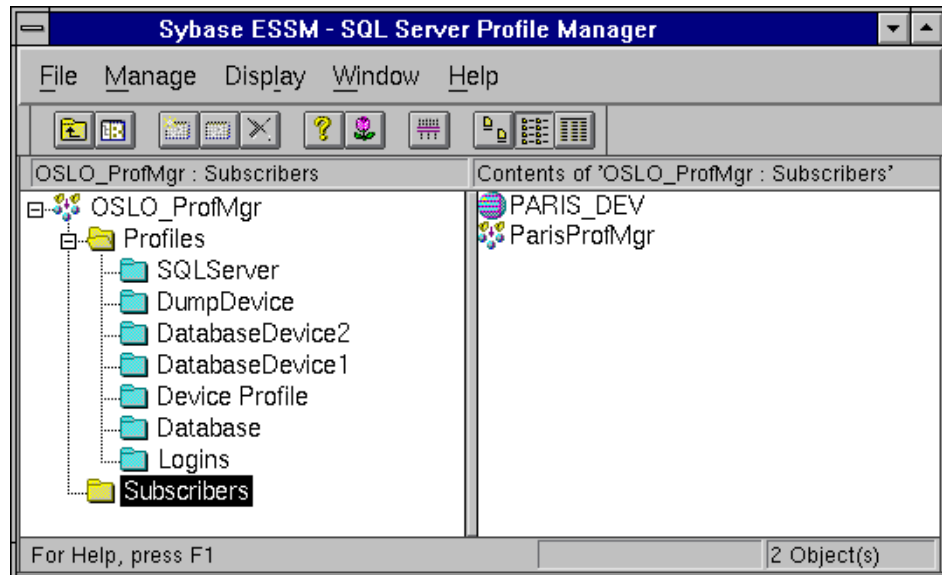


Figure 11-9: SQL Server Profile Manager with Subscribers container selected

Details View

Profile manager windows use the three view options described in “Customizing the Display” in Chapter 3, “Enterprise SQL Server Manager Windows and Dialog Boxes.” The Details command on the Display menu shows detailed information about each object in the right pane.

Profile Manager Menus

A Profile Manager window has a set of standard menus and context-sensitive menus.

Standard Profile Manager Menus

The standard menus in a profile manager are:

- File
- Manage
- Display
- Window
- Help

The Manage menu is described in this section. The File, Display, Window, and Help menus have the same commands as the SQL Server window. For information about these menus, see Chapter 3, “Enterprise SQL Server Manager Windows and Dialog Boxes.”

The Manage Menu

The Manage menu has the same commands in both types of profile managers, except for the first menu item. In a SQL Server Profile Manager window, the first command on the Manage menu is Open Source SQL Server. In a Database Profile Manager window, the first command on the Manage menu is Open Source Database.

Table 11-3 describes the commands on the Manage menu.

Table 11-3: Manage menu in a profile manager window

Menu Command	Function
Open Source SQL Server (SQL Server Profile Manager)	Opens a SQL Server window for the source SQL Server.
Open Source Database (Database Profile Manager)	Opens a SQL Server window with the source database selected in the left pane and the contents of that database appearing in the right pane.
Profiles–	
Create Profiles	Opens the Create Profile dialog box.
Populate	Opens the Populate Profile dialog box.
Synchronize Profile	Synchronizes a profile with SQL Server.
Distribute	Opens the Distribute Profile dialog box.
Default Policy	Opens the Default Policy dialog box.
Validation Policy	Opens the Validation Policy dialog box.
Validate Profile	Checks attributes of profile members against validation policies.
Delete	Deletes the selected profiles.
Subscribers–	
Add/Remove	Opens the Subscribers dialog.
Open	Opens the SQL Server window or profile manager window for the selected subscriber.

Context-Specific Menus

The context-specific menus are displayed based on the type of container and profile selected in the window. The menu commands on most context-specific menus are Properties and Delete. In a SQL Server Profile Manager, the `SQLServerProfile` does not have a context-specific menu. In a Database Profile Manager, the `SQLDatabaseProfile` does not have a context-specific menu. However, these two profiles have popup menus.

Some tasks that you can do in a SQL Server window are not available in a profile manager.

Shortcut Menus

As in the SQL Server window, containers and icons in profile manager windows have shortcut menus.

Table 11-4: Shortcut menu commands for profile manager windows

Context	Commands
Profiles container	<ul style="list-style-type: none"> • Create Profiles
Subscribers container	<ul style="list-style-type: none"> • Add/Remove
Container icons in left pane (except <code>SQLServerProfile</code> in a SQL Server Profile Manager and <code>SQLDatabaseProfile</code> in a Database Profile Manager)	<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 • Populate • Synchronize Profile • Distribute • Default Policy • Validation Policy • Validation • Validate Profile • Delete • New Window <p>The Table container has an additional menu item – Copy Data on Distribute</p>

Table 11-4: Shortcut menu commands for profile manager windows (continued)

Context	Commands
SQLServerProfile container	<ul style="list-style-type: none"> • Configuration • Synchronize Profile • Distribute • Delete • New Window
SQLDatabaseProfile container in a Database Profile Manager	<ul style="list-style-type: none"> • Options • Synchronize Profile • Distribute • Delete • New Window
Container icons in right pane	Create object
SQLServerProfile icon in right pane	Configuration
All object icons in the right pane (except SQLServerProfile in SQL Server Profile Manager and SQLDatabaseProfile in Database Profile Manager)	Same commands as the pull-down menu for the object (Properties and Delete)
Right pane (when clicked in an area containing no icons)	<ul style="list-style-type: none"> • Large Icon • List • Details • Create <i>object_type</i>, where <i>object_type</i> is the object in the pane—for example, Create Login

Opening a SQL Server Window from a Profile Manager

To open a SQL Server window from a profile manager, use one of the following methods:

- From the Manage menu in a SQL Server Profile Manager, choose Open Source SQL Server.
- From the Manage menu in a Database Profile Manager, choose Open Source Database.
- From the Subscribers menu, choose Open.
- From a subscriber's pop-up menu, choose Open.

12

Managing ESSM Profiles

What's in this Chapter

Chapter 11, “Enterprise SQL Server Manager Profile Managers,” explained how to work with profile managers. This chapter describes how Enterprise SQL Server Manager uses profiles to allow you to manage large numbers of SQL Servers from a single location. This chapter describes:

- Managing profiles
- Creating objects in profile managers
- Managing subscribers
- Distributing profiles

► *Note*

Unless otherwise specified, all discussion of profiles and profile management in this chapter applies only to profiles as implemented by Enterprise SQL Server Manager and managed from a SQL Server Profile Manager window or a Database Profile Manager window. This discussion does not apply to Tivoli profile managers.

Managing Profiles

You manage profiles in a SQL Server Profile Manager window or a Database Profile Manager window.

Managing profiles involves the following tasks:

- Creating profiles
- Populating profiles
- Unpopulating profiles
- Adding subscribers to profiles
- Removing subscribers from profiles
- Distributing profiles
- Deleting profiles

Creating a Profile

	TME	ESSM	SQL Server
Required roles	senior, or super	none	none

To create a profile:

1. From the Manage menu in a profile manager window, choose Profiles→ Create Profiles. The Create Profiles dialog box opens.

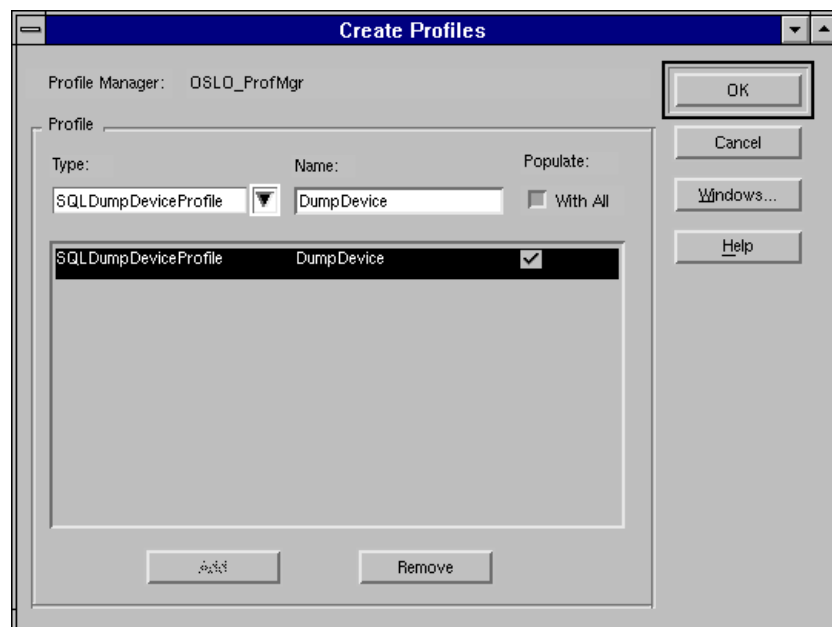


Figure 12-1: Create Profiles dialog box

2. In the Type box, select the type of profile you want to create from the drop-down list.
3. In the Name box, type a name for the profile.
4. Optionally, select the Populate: With All check box. This command works as follows:
 - If you check the With All check box, the profile is populated with all the objects of the selected type in the SQL Server or database.
 - If you have previously added a profile of the specified type, any object that is not already a member of a profile is added to this profile.

- If a SQL Server profile or database profile is selected, the check box is disabled. These profile types cannot be populated.

For more information about populating profiles, see “Populating a Profile” on page 12-5.

5. Profiles are listed in the display area as tabular rows.
 - Click the Add button to move the profile into the table as a new row.
 - Select a row and click the Remove button to remove a profile from the table.

► **Note**

The Add and Remove buttons are disabled if the option is not available. Add is available when a valid type and name are entered. Remove is available if there is an entry selected in the list.

6. Click OK to create the profile. Enterprise SQL Server Manager adds an icon in the profile manager window for each new profile.

Shortcut



Click the right mouse button over the Profiles container.
Choose Create Profiles from the popup menu.

Figure 12-2: Shortcuts for creating a profile

Deleting a Profile

	TME	ESSM	SQL Server
Required roles	admin, senior, or super	none	none

To delete profiles:

1. In a profile manager window, select the profile you want to delete.
2. From the Manage menu, choose Profiles→ Delete. Enterprise SQL Server Manager displays a confirmation dialog box, asking if you want to proceed with the delete operation.



Figure 12-3: Delete profile confirmation dialog box

3. Confirm the deletion.

Shortcut



In the left pane of the profile manager window, click the right mouse button over the container icon. Select Delete from the popup menu.

Figure 12-4: Shortcuts for deleting a profile

Populating a Profile

A profile contains objects that you want to distribute. For example, in a login profile, you list the logins that you want to distribute. These objects are called the population of the profile.

There are two profile types that cannot be populated: the SQL Server profile in a SQL Server Profile Manager and the database profile in a Database Profile Manager. These profiles are not true containers; they contain configuration information, not objects.

► **Note**

You can populate a database profile in a SQL Server Profile Manager.

There are two ways to populate a profile: with the **Populate With All** choice in the **Create Profile** dialog box and with the **Populate Profile** dialog box.

	TME	ESSM	SQL Server
Required roles	senior or super	server; for database objects, you need schema	System Security Officer

Populating a Profile with All Objects in One Step

When you create a profile, you can populate it with one of each object of that type in the SQL Server or database. For example, if you create a login profile and choose **Populate With All**, all the logins in the SQL Server associated with the profile manager are added to the profile.

If you select the **Populate: With All** command on the **Create Profile** dialog box, the profile is populated at the same time that it is created. If you do not select this option, or if it is not available, use the **Populate Profile** dialog box to add objects to a profile.

► **Note**

If you have previously added a profile of the specified type, **Populate with All** adds any object that is not already a member of a profile to this profile.

Populating a Profile with the Populate Profile Dialog Box

To populate a profile after it has been created:

1. From the Manage menu in a profile manager window, choose Profile → Populate. The Populate Profile dialog box opens.

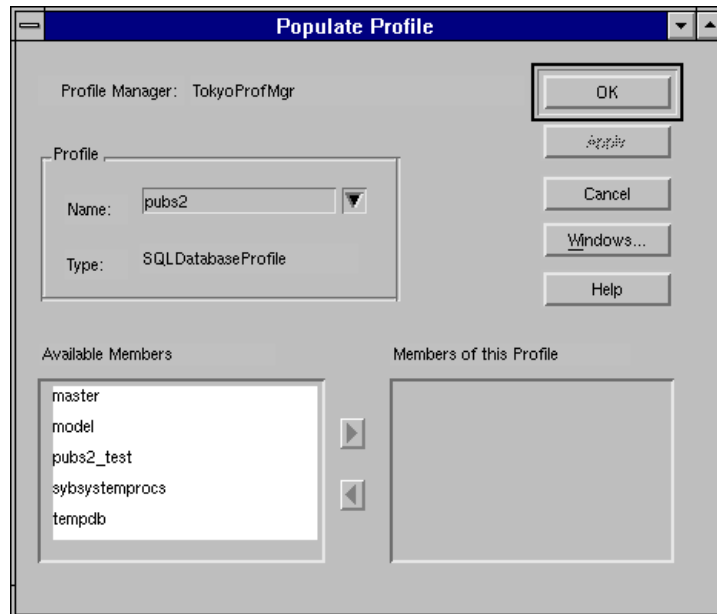


Figure 12-5: Populate Profile dialog box

2. In the Name box, type the name of the profile you want to populate or select a profile from the list. The type label lists the profile type.
3. In the Available Members box, select the objects that you want to add to the profile.
4. Click the right-pointing arrow button to move the selected objects to the Members of This Profile box.
5. Click OK. Enterprise SQL Server Manager adds icons for the objects in the profile manager window.

When a database object is added to a profile, it is named in the format *owner.object_name*. Indexes are named *owner.table_name.index_name*. Icons are labeled in the format *object_name(owner)*.

► Note

When you populate a profile, it is populated with the database or SQL Server object. It is not populated with DDL. You cannot distribute DDL.

Shortcut

In the left pane of the profile manager window, click the right mouse button over the profile container icon. Select Populate from the popup menu.

Unpopulating a Profile

	TME	ESSM	SQL Server
Required roles	senior or super	none	System Security Officer

To unpopulate a profile:

1. From the Manage menu in a profile manager window, choose Profile → Populate. The Populate Profile dialog box opens (Figure 12-5).
2. In the Name box, type the name of the profile you want to unpopulate or select a profile from the list. The type label lists the profile type.
3. In the Members of This Profile box, select the object names that you want to remove from the profile.
4. Click the left-pointing arrow button to move them to the Available Members box.
5. Click OK. Enterprise SQL Server Manager removes the icons for the objects from the profile manager window.

Shortcut

In the left pane of the profile manager window, click the right mouse button over the profile container icon. Select Populate from the popup menu.

Synchronizing a Profile

Profile synchronization lets you update a profile to capture changes made to the source SQL Server outside of Enterprise SQL Server Manager. When you change an object in a profile manager, the object is changed in the associated SQL Server. However, a change made to an object in the associated SQL Server from outside of Enterprise SQL Server Manager, by a local administrator, for example, does not change the profile. Therefore, you must synchronize the profile to make sure it is an accurate copy of the object in SQL Server.

	TME	ESSM	SQL Server
Required roles	senior or super	none	System Security Officer

To synchronize a profile:

1. In the left or right pane of a profile manager, select the profile that you want to synchronize.
2. From the profile's popup menu, choose Synchronize Profile.

When Enterprise SQL Server Manager finishes synchronizing the profile, a completion message is displayed.

Validating a Profile

If you have validation policies enabled, you can check to see if the members of a profile conform to the validation policies. To validate a profile:

1. Select the profile you want to validate.
2. From the Manage menu, choose Profiles→ Validate Profile. Enterprise SQL Server Manager checks the attributes of each member of the profile against the validation policies that are enabled. The Validation Results window displays a report that lists members and attributes that do not conform to the enabled policies. If the display is empty, all members conform to the validation policies.

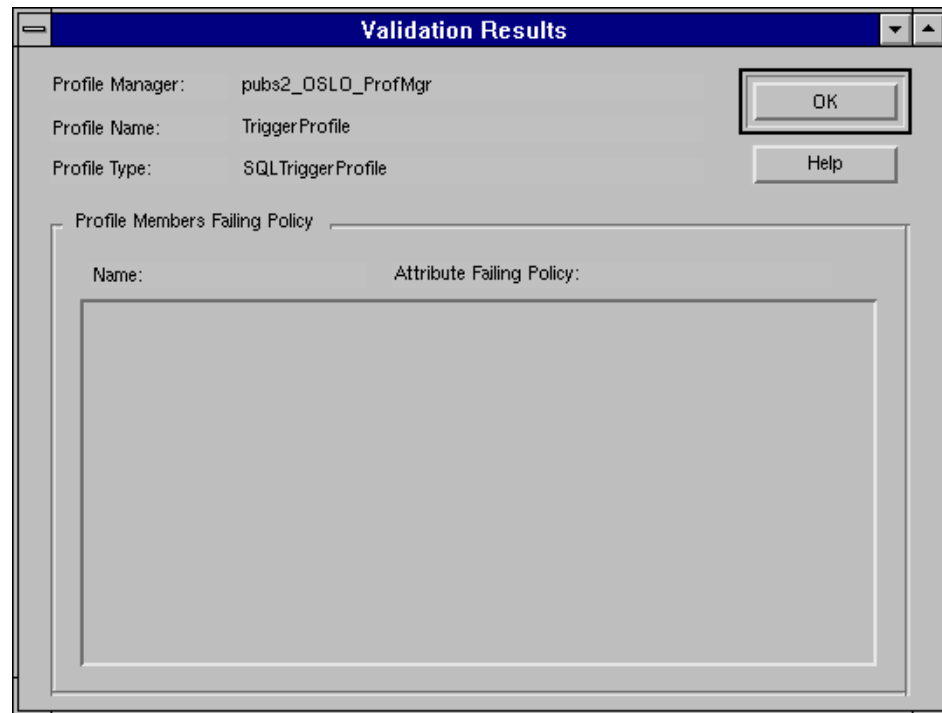


Figure 12-6: Validation Results window

Shortcut



In the left pane of the profile manager window, click the right mouse button over the profile container icon. Select Validate Profile from the popup menu.

Creating and Deleting Objects in Profile Managers

TME/ESSM/SQL Server

Required roles	To manipulate objects in a profile manager, you need the same roles as to manipulate them in a SQL Server window, plus the server role.
-----------------------	--

A profile manager gives direct access to the objects included in a profile. Using context-specific menu commands, you can create, modify, and delete SQL Server objects in a profile manager window. Objects that you create become part of SQL Server just as if you created them in the SQL Server window or by using an Enterprise SQL Server Manager command. If you create an object in a profile, the object is automatically added to the profile.

Changes made to objects in a profile manager window are made to the actual objects in the source SQL Server. These changes can be distributed to subscribers of the profile.

If you delete an object in a profile, it is deleted from the source SQL Server. In the profile, it is marked as deleted, and the next time you distribute the profile, the object is deleted from the subscribing SQL Servers.

◆ **WARNING!**

Deleting an object is not the same as removing it as a member of a profile (unpopulating the profile). If you remove an object from a profile, it is still an object in SQL Server, but is no longer distributed as part of the profile. If you delete an object, it is no longer present in SQL Server.

Creating Objects: Getting Default Values

When you create an object in a profile, default policy applies. However, the default values are not entered into the text boxes of the dialog box automatically. You have two options for generating the default values:

- Click **Get Defaults**. The default values are entered in the text boxes of the dialog box. If input values for a script have not been entered in the dialog box, you receive a message asking you to enter them. Enter the required information, then click **Get Defaults** again.
- Enter required information in the create dialog box. Default values are generated when you click **Create** and are visible in the **Properties** tab of the new object's dialog box.

To override a default value, enter a different value in the appropriate text box.

► **Note**

The **Get Defaults** button is available only when you create an object in a profile manager.

Managing Subscribers

Managing subscribers involves the following tasks:

- Adding subscribers to profile managers
- Removing subscribers from profile managers

A subscriber is a SQL Server, database, or profile manager that receives the profiles distributed by a profile manager. A profile manager can contain one or more subscribers. Table 12-1 lists the types of subscribers allowed in the two kinds of profile managers.

Table 12-1: Subscriber types allowed in profile managers

Profile Manager	Subscribers
SQL Server Profile Manager	Managed SQL Servers SQL Server Profile Managers
Database Profile Manager	Databases Database Profile Managers

Profile Managers as Subscribers

Enterprise SQL Server Manager profile managers can subscribe to other Enterprise SQL Server Manager profile managers of the same type. This feature lets you establish hierarchical distributions of profiles for scalability or security.

If a profile manager subscribes to another profile manager, Enterprise SQL Server Manager automatically adds the SQL Server or database associated with the subscribing profile manager as a subscriber, too. (Every profile manager has a SQL Server or a database associated with it. The associated SQL Server or database is the source of the profiles in the profile manager.) The relationship is implicit; the SQL Server or database is not listed as a subscriber to the profile manager.

Figure 12-7 illustrates a SQL Server profile manager named `Koko_ProfMgr` that is associated with SQL Server `Koko`. If `Koko_ProfMgr` subscribes to SQL Server profile manager `Agra_ProfMgr`, the SQL Server `Koko` also becomes a subscriber to `Agra_ProfMgr`.

► **Note**

If you distribute to a subscribing profile manager, the profile data does not get distributed to that subscribing profile manager's subscribers (Figure 12-8.) If you want the subscribing profile manager's subscribers to get the data, you must explicitly distribute the objects to the subscribing profile manager's subscribers.

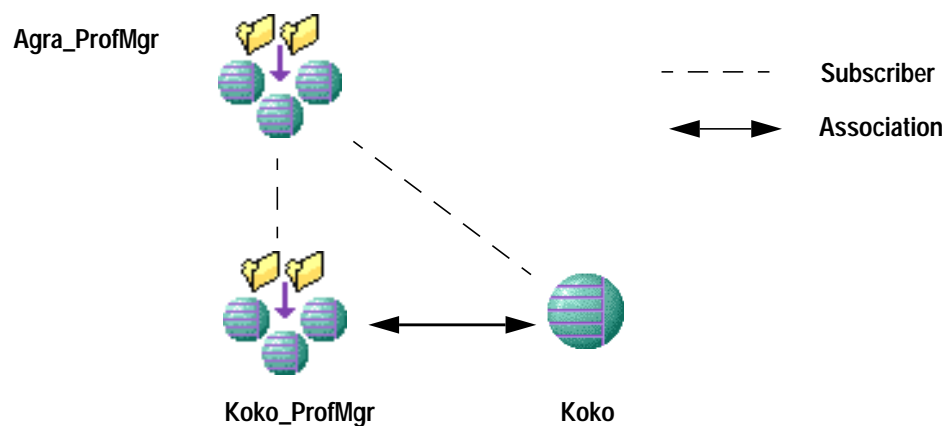


Figure 12-7: A profile manager as a subscriber

As illustrated in Figure 12-8, when profiles are distributed from an Enterprise SQL Server Manager profile manager, they get distributed to both the subscribing profile manager and its associated SQL Server, which has the effect of keeping the two synchronized. You can use hierarchical profile managers to organize and manage core resources in one location (a profile manager) and distribute those resources to subscribers and other profile managers, which can augment those resources as needed.

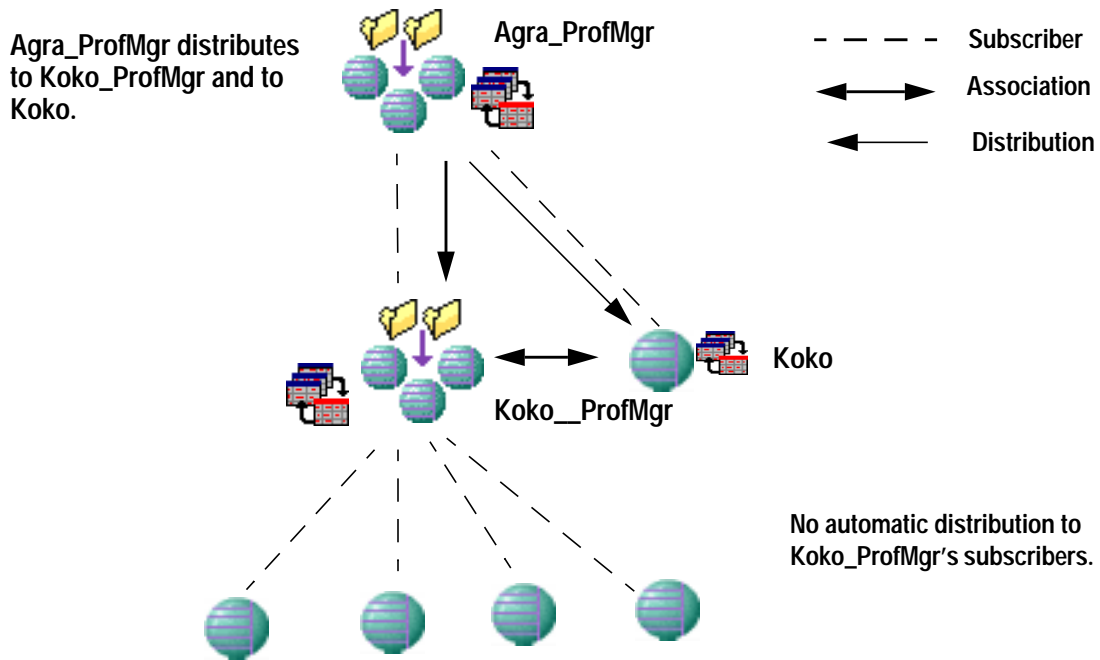


Figure 12-8: Distribution to a subscribing profile manager

Adding Subscribers to a SQL Server Profile Manager

	TME	ESSM	SQL Server
Required roles	admin, senior or super	none	none

To add subscribers to a SQL Server Profile Manager:

1. From the Manage menu in a Profile Manager window, choose Subscribers → Add/Remove. The Subscribers dialog box opens. You can add managed SQL Servers or SQL Server Profile Managers as subscribers.

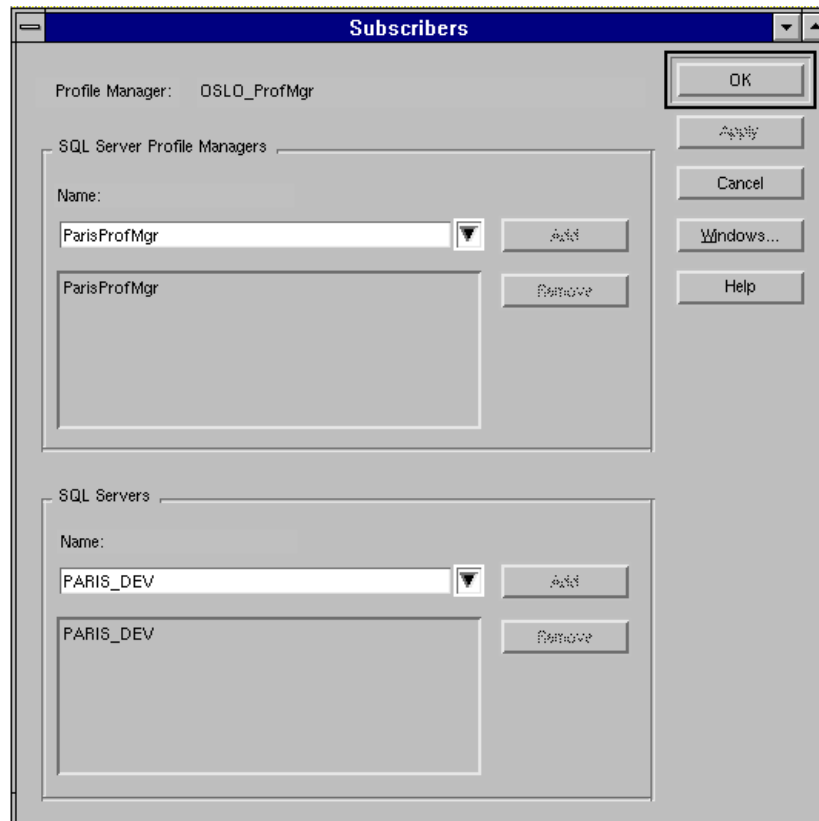


Figure 12-9: Subscribers dialog box for a SQL Server profile manager

2. To add a SQL Server Profile Manager as a subscriber, in the Name text box of the SQL Server Profile Managers group box, type the name of a SQL Server Profile Manager that you want to subscribe to this profile manager, or select a name from the drop-down list.

3. Click Add to add the SQL Server Profile Manager to the list of subscribers.
4. To add a SQL Server as a subscriber, in the Name text box of the SQL Servers group box, type the name of a SQL Server that you want to subscribe to this profile manager, or select a name from the drop-down list.
5. Click Add to add SQL Server to the list of subscribers.
6. Click OK to apply changes and exit the window. Click Apply to apply the change and keep the window open for further changes.



Release Compatibility

SQL Server release 10.x can subscribe to a SQL Server Profile Manager associated with a SQL Server release 11.0 and SQL Server release 11.0 can subscribe to a SQL Server Profile Manager associated with SQL Server release 10.x. The following restrictions apply to these relationships:

- You cannot distribute named cache profiles to SQL Server release 10.x.
- Do not distribute a `SQLServerProfile` to a SQL Server at a different release than the SQL Server Profile Manager. The configuration options of the SQL Server associated with the SQL Server Profile Manager must match those of the subscriber.

Shortcut



Click the right mouse button over the Subscribers container icon. From the popup menu, choose Add/Remove.

Adding Subscribers to a Database Profile Manager

	TME	ESSM	SQL Server
Required roles	admin, senior, or super	none	none

To add or display subscribers to a Database Profile Manager:

1. From the Manage menu in the Database Profile Manager window, choose Subscribers → Add/Remove. The Subscribers dialog box opens.

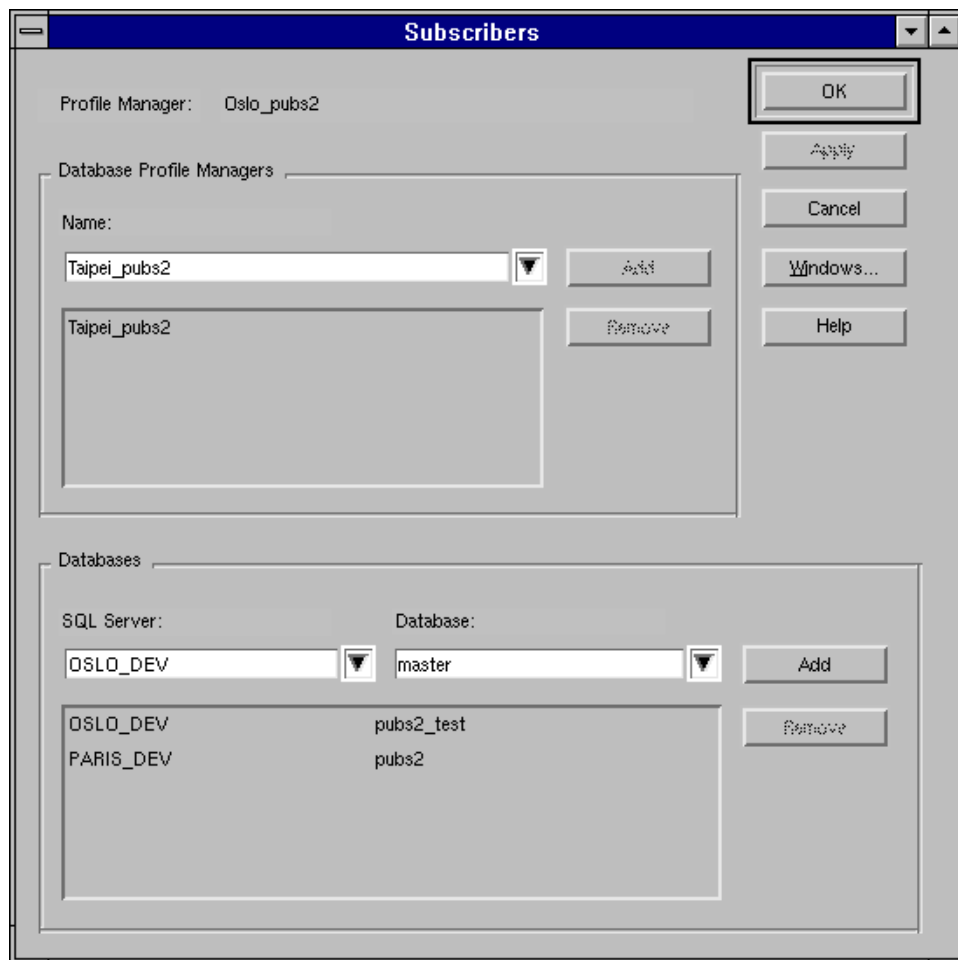


Figure 12-10:Subscribers dialog box for a database profile manager

2. To subscribe a Database Profile Manager to this profile manager, type the name of a database profile manager in the Name box or select one from the drop-down list.

3. Click Add to add the Database Profile Manager listed in the Name box to the list.
4. To subscribe databases to this profile manager, type the name of a SQL Server in the SQL Server box or select a SQL Server from the drop-down list. When you select a SQL Server, the databases in that SQL Server become available in the Database drop-down list and any databases in the SQL Server that are already subscribers are listed in the list box.
5. In the Database box, select a database that you want to subscribe to this profile manager.
6. Click Add to move it to the list of subscribers.
7. Click OK to apply the change and exit the window.
Click Apply to apply the change and keep the window open for further changes.

Shortcut



Click the right mouse button over the Subscribers container icon. From the popup menu, choose Add/Remove.

Removing Subscribers from a SQL Server Profile Manager

	TME	ESSM	SQL Server
Required roles	senior or super	none	none

To remove a subscriber from a SQL Server Profile Manager:

1. From the Manage menu in a profile manager window, choose Subscribers → Add/Remove. The Subscribers dialog box opens.
2. To unsubscribe a SQL Server Profile Manager, select the SQL Server Profile Manager in the SQL Server Profile Managers list box.
3. Click Remove.
4. To unsubscribe a SQL Server, select the SQL Server in the SQL Servers list box.
5. Click Remove.
6. Click OK.

Shortcut



Click the right mouse button over the Subscribers container icon. From the popup menu, choose Add/Remove.

Removing Subscribers from a Database Profile Manager

	TME	ESSM	SQL Server
Required roles	senior or super	none	none

To remove subscribers from a Database Profile Manager:

1. From the Manage menu in a Database Profile Manager window, choose Subscribers→ Add/Remove. The Subscribers dialog box opens.
2. To unsubscribe a Database Profile Manager, select the Database Profile Manager in the Database Profile Managers list box.
3. Click Remove.
4. To unsubscribe a database, in the SQL Server box, select the SQL Server that contains the database you want to unsubscribe.
5. In the list box, select the database you want to unsubscribe.
6. Click Remove.
7. Click OK.

Shortcut



Click the right mouse button over the Subscribers container icon. From the popup menu, choose Add/Remove.

Distributing Profiles

From a profile manager, you can distribute one or more profiles to one or more subscribers.

Because there are many SQL Server object dependencies (for example, logins require that the selected default database exists before the login can be created), it is important to distribute profiles in the correct sequence. When you select multiple profiles to distribute, Enterprise SQL Server Manager distributes the profiles in the correct sequence. However, if you choose to distribute one profile at a time, remember that unresolved dependencies can result in an unsuccessful distribution. (For information about dependencies, see Appendix C, “SQL Server Profile Operations.”)

You cannot distribute a SQL Server profile to a SQL Server with a different release. In other words, you cannot distribute a SQL Server profile of SQL Server release 10.x to SQL Server release 11.0 and vice versa.

	TME	ESSM	SQL Server
Required roles	admin, senior, or super	see Table 12-2	see Table 12-2

Table 12-2: ESSM and SQL Server roles required for distributing ESSM profiles

Profile Type	ESSM Role	SQL Server Role
SQL Server	server	System Administrator
Database, Database Device, Dump Device, Segment	space	System Administrator
Login, Remote Login	security	System Administrator
Remote Server, Group, User	security	System Security Officer
Cache, Default, Rule, Table, Trigger, User - datatype, Index, View, Procedure	schema	System Administrator

► **Note**

To distribute a remote server that contains remote logins, you must have the System Administrator instead of the System Security Officer.

Distributing All Profiles to All Subscribers

To distribute all profiles in a profile manager to all subscribers:

1. From the profile manager pop-up menu in a policy region, choose **Distribute**. The **Distribute Profiles** dialog box opens.

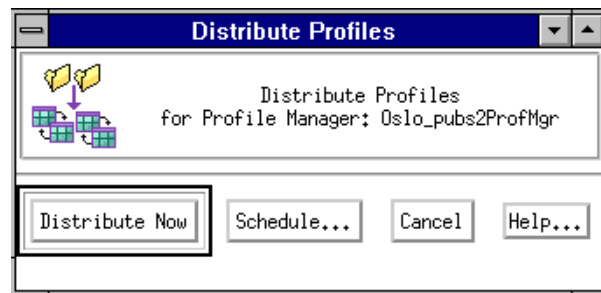


Figure 12-11: Distribute Profiles dialog box for policy region

2. To distribute the profiles immediately, select **Distribute Now**. To schedule the distribution, select **Schedule**. For information about how to use the Scheduler, see Chapter 9, "The Scheduler" in the *Tivoli Management Platform Guide*.

Distributing Profiles Selectively

To distribute profiles selectively:

1. From a profile manager window, select Profile → Distribute. The Distribute Profile dialog box opens.

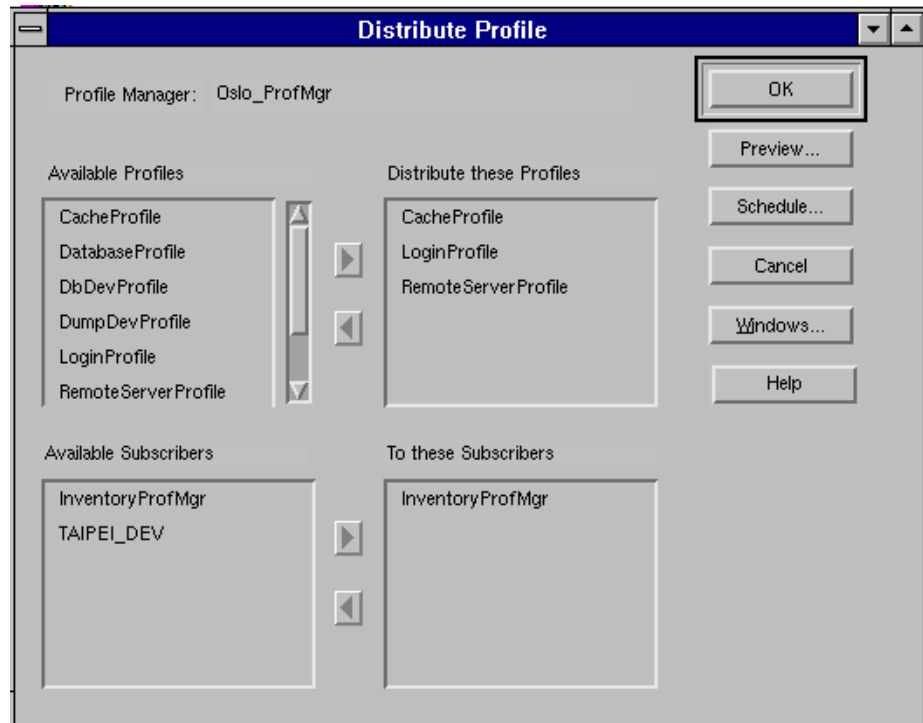


Figure 12-12:Distribute Profile dialog box

2. In the Available Profiles list box, select the profiles that you want to distribute.
3. Click the right-pointing arrow to move the selected profiles to the Distribute These Profiles list.
4. In the Available Subscribers list box, select the subscribers to which you want to distribute the profiles.
5. Click the right-pointing arrow to move the selected subscribers to the To These Subscribers list. The OK, Preview, and Schedule buttons become enabled.

6. Click one of the command buttons, as follows:
 - Click OK to distribute the profiles. The dialog box closes and a message dialog box appears stating that the Distribute operation is in progress. The hourglass and this dialog remain until the distribute operation is complete or fails. After the distribution is finished, the dialog box displays a status message. For information about distribution failures, see “How Distribution Failures are Handled” on page 12-27.
 - Click Preview to see a description of the information that will be distributed. For more information about Preview, see “Previewing a Distribution” on page 12-24.

Shortcuts



Drag the icon for the subscriber to whom you want to distribute onto the icon for the profile that you want to distribute or onto the Profiles container.



Drag the icon for the profile you want to distribute onto the subscriber to whom you want to distribute it or onto the Subscribers container.



Click the right mouse button over the container icon for the profile you want to distribute. From the popup menu, choose Distribute.

Previewing a Distribution

If you select the Preview button on the Distribute Profile dialog box, the Preview Distribution dialog box opens.

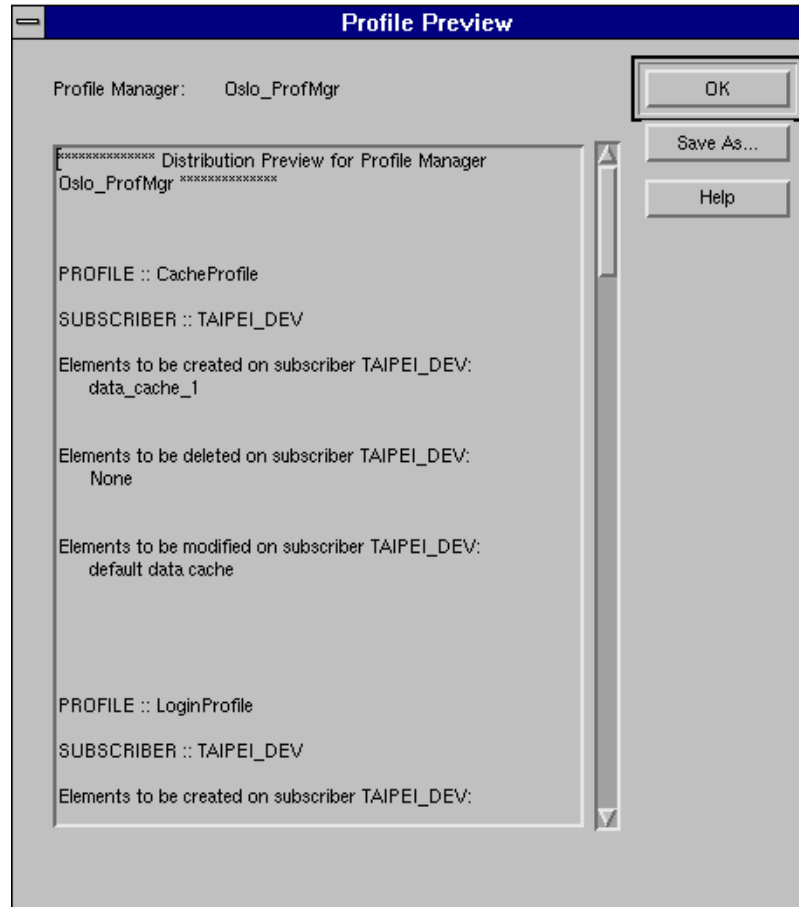


Figure 12-13: Preview Distribution dialog box

The Preview Distribution dialog box displays, on a per profile and per subscriber basis, the information that Enterprise SQL Server Manager will try to distribute. To save the output to a file, click the Save As button. A Save dialog box is displayed. Save the text to a file.

The output of the Preview operation is similar to the following:

```
** Distribution Preview for Profile Manager MineralProfMgr **
PROFILE :: TableProf
SUBSCRIBER :: OSLO_DEV/Animals
Elements to be created on subscriber OSLO_DEV/Animals:
    Gems
    Rocks
Elements to be deleted on subscriber OSLO_DEV/Animals:
    None
Elements to be modified on subscriber OSLO_DEV/Animals:
    None
*End of Distribution Preview for Profile Manager MineralProfMgr
*
```

Distributing Table Data

You can distribute (copy) the data in a table as part of a profile distribution.

The data copy feature of Enterprise SQL Server Manager is not intended to be a replacement for Replication Server®. The process is constrained by system resources such as CPU and, more importantly, available disk space. Therefore, do not use Enterprise SQL Server Manager to copy very large tables such as data entry tables commonly used for transaction processing systems.

Use Enterprise SQL Server Manager to copy supporting application tables such as lookup tables (state abbreviations, zip codes, and so on).

The directory *\$DBDIR* is used for storing the data files. Be sure to allot disk space for *\$DBDIR* on the source and target hosts.

Enabling and Disabling Distribution of Table Data

If table data copy is enabled, the data in all tables in the profile is distributed. You enable or disable distribution of table data by toggling the Data Copy command on the popup menu for table profiles. The command always indicates the current state of data copy.

To enable table data copying:

1. Click the right mouse button on a table profile icon.
2. From the popup menu, choose Copy Data on Distribute. A check box is added to the menu next to the command.

To disable table data copying:

1. Click the right mouse button on a table profile icon.
2. From the popup menu, choose Copy Data on Distribute. The check box is removed from the menu.

How Distribution Failures are Handled

Distribution failures occur on a per profile, per subscriber basis. If an operation fails for one item in a profile, all operations involving that profile are rolled back and nothing in that profile is distributed to that particular subscriber. However, distribution of that same profile to other subscribers can still succeed, because a failure on one subscriber does not affect the other subscribers.

When you select multiple profiles to distribute, Enterprise SQL Server Manager distributes them in the sequence necessary to satisfy all possible SQL Server object dependencies. If the distribution of a profile in the sequence fails on a subscriber, Enterprise SQL Server Manager does not stop; distribution of profiles continues. Therefore, some object dependencies may not be satisfied. If part of a distribution fails, you can fix the problem that caused the failure and run the distribution again, or you can manually undo the distribution and try it again.

To manually undo a distribution, you must identify the objects that were successfully distributed. Then, you must restore each object to the state that it was in before the distribution.

If a distribution that has databases for endpoints fails, review the relevant notice in the Sybase Administration notice group for identification of the SQL Server on which it failed.

13

Managing Policy

What's in this Chapter

This chapter describes the following:

- Definition of policy
- How to define default policy
- How to generate default policy values
- How to define validation policy
- How to write policy scripts
- How to enable validation policy

What is Policy?

Policies are rules that you can implement to control what happens when you create objects or change them. Default policies let you specify default values for attributes of objects you create. Validation policies are rules for objects in a collection. The policies are checked when you modify an object, add a new object to a collection or create a new object in a collection.

Enterprise SQL Server Manager supports use of policy through profile managers. Objects added to profiles, and objects created or changed in the context of a profile, are subjected to that profile's defined policies.

Objects changed using Enterprise SQL Server Manager commands or through the SQL Server window are subjected to policy only if they are members of a profile. This means that if an administrator changes an object by using the SQL Server window or by executing an `sset` command, Enterprise SQL Server Manager checks whether that object is a member of a profile. If it is, and if a validation policy is enabled on the object type, it is compared to the policy of that profile. If the object isn't a member of a profile, the object is not subjected to a policy.

Table 13-1 describes how Enterprise SQL Server Manager applies policy to new and existing objects:

Table 13-1: Application of policy to object changes and creation

Action	in profile manager	in SQL Server window or from command line
Create object	Apply policy	Do not apply policy
Change object	Apply policy	If object is in a profile and is subject to policy, apply policy

The integration of policy and profile management lets you decide how strictly you want to enforce policy in your enterprise. If you want all object creations and modifications to be subjected to policy, you can change the administrator's desktop so that specific administrators have only profile managers, and not the source SQL Server, available on their desktop. Therefore, all their actions would be subjected to policy. You can do this because senior role is required for creating a profile and defining policy. Creating or modifying elements in a profile requires a different set of roles (admin for Tivoli applications; security, space, and so on, for Enterprise SQL Server Manager).

For more information about policy, see *Tivoli Management Platform User's Guide*.

Distributing Policy

Enterprise SQL Server Manager distributes policy along with profiles, but only to subscribing profile managers. Unlike in the generic Tivoli implementation of policy, Enterprise SQL Server Manager does not distribute policy to endpoints (SQL Server installations and databases).

After you distribute policy to a subscribing profile manager, you can change policy at the subscriber. However, if you distribute policy again, the distributed policy overwrites any changes that you made.

Defining Default Policies

Default policy is applied to objects that you create in a profile manager. An example of a default policy is that every new user has the same default group.

	TME	ESSM	SQL Server
Required roles	senior or super	none	none

To define default policy:

1. From the Manage menu in a profile manager window, choose Profiles→ Default Policy. The Default Policy dialog box is displayed.

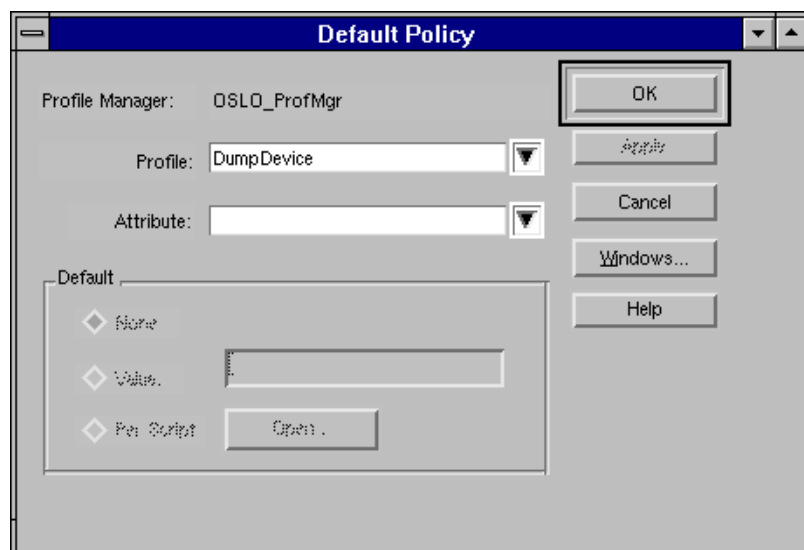


Figure 13-1: Default Policy dialog box

2. Select a profile name from the Profile drop down list.
3. Select the attribute for which you want to specify policy from the Attribute drop down list.
4. Specify a default option:
 - None - no default policy for this attribute.
 - Value - Type a specific value in the text box.
 - Per Script - Click Open. The Policy Script dialog box opens. Complete the Policy Script dialog box.

5. Click OK to apply the changes and close the dialog box, or click Apply to apply the changes and leave the dialog box open so you can create or edit more default policies.

Shortcut



Click the right mouse button over the profile container icon. From the popup menu, choose Default Policy.

Table 13-2 lists the object attributes that can be subjected to policy and the types of values they take.

Table 13-2: Object attributes subject to policy

Profile	Attributes	Type
SQLLoginProfile	Default Database Default Language Login Roles Password (validation policy only)	string string string list string
SQLDbDeviceProfile	Controller Number Logical Name Physical Name Size VDevNo	number string string number number
SQLDumpDeviceProfile	Device Physical Name Size	string string number
SQLRemoteServerProfile	Default Login Map Local Login Remote Server Timeout Encrypt	Boolean string string Boolean Boolean
SQLDatabaseProfile	Database Owner	string string
SQLCacheProfile	Cache Name	string

Table 13-2: Object attributes subject to policy (continued)

Profile	Attributes	Type
SQLUserProfile	Aliases Group Login Name User	list string string string
SQLGroupProfile	Name	string
SQLSegmentProfile	Device Names Name	list string
SQLTableProfile	Table Owner Segment	string string string
SQLViewProfile	Name Owner Name	string string
SQLIndexProfile	Index Name Owner Name Segment Name	string string string
SQLTriggerProfile	Name Owner Name	string string
SQLProcedureProfile	Name Owner Name	string string
SQLRuleProfile	Name Owner Name	string string
SQLDefaultProfile	Name Owner Name	string string
SQLDataTypeProfile	User Datatype Name User Phystype	string string string

Attribute Values

Attributes take a variety of value types, as follows:

- String value attributes can take any ASCII string.
- Numeric attributes can take any number.
- Boolean attributes take 0 or 1.
- List attributes, such as roles for Logins, take a list of values separated by spaces.

Be sure to use correct value types, particularly if you create policy scripts.

Generating Default Values

When you open the create dialog box for an object that has a default policy, the default values are not entered automatically. You have two options for generating the default values:

- Click Get Defaults. The default values are entered in the text boxes of the dialog box. If input values for a script are not entered in the dialog box, you receive a message asking you to enter them. Enter the required information, then click Get Defaults again.
- Enter required information in the create dialog box. Default values are generated when you click Create and are visible in the Properties tab of the new object's dialog box.

To override a default value, enter a different value in the appropriate text box.

Default Policy Behavior

When you use default policy, be aware of how it interacts with the values you type into create dialog boxes:

- The Get Defaults button is available only when you create an object in a profile manager.
- The following default policy attributes are not displayed in create dialog boxes. These default attributes are applied after you click Create.

Table 13-3: Attributes not supported by Get Defaults

Object	Attribute
Remote Server	Timeout
	Encrypted password
User	Aliases
Segments	Devices

- If you enter a value in a text box for which a default value is configured, the value you enter overrides the default value.
- For lists, such as aliases, if you specify list items without getting the defaults, they override the defaults. If you get the default values and then add to the list, all the list items get applied.

Enabling and Disabling Default Policy

Default policy is not enabled or disabled. If a default value or script is assigned to an attribute it is always in effect. If you want to remove a default policy assignment, use the Default Policy dialog box and change the default for the attribute to None.

➤ *Note*

If you select None, Enterprise SQL Server Manager assigns a null string to the attribute ("").

◆ **WARNING!**

The Tivoli command option `wputpolm -n` turns off default policy. If you execute this command, the Default Policy dialog box and the Get Defaults button will not work. If you want to turn off policy from the command line, do not use the `-n` argument, set the attribute value to a null string.

Defining Validation Policies

If validation policy is enabled, it takes effect when you:

- Create an object in a profile
- Populate a profile
- Change an object that is part of a profile for which a validation policy is enabled

	TME	ESSM	SQL Server
Required roles	senior or super	none	none

To define validation policy:

1. From the Manage menu in a profile manager window, choose Profiles → Validation Policy. The Validation Policy dialog box is displayed.

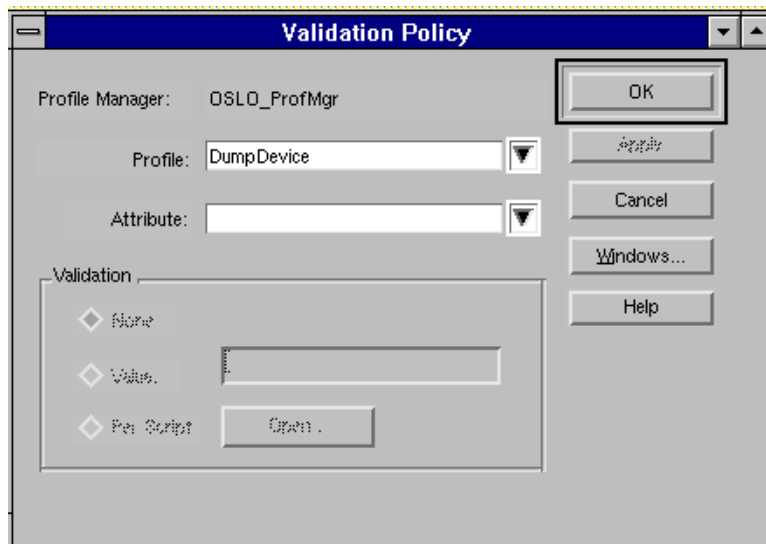


Figure 13-2: Validation Policy dialog box

2. Select a profile name from the Profile drop down list.
3. Select the attribute for which you want to specify policy from the Attribute drop down list.

4. Specify a validation option:
 - None – no validation policy for this attribute.
 - Value – Type a specific value or a Perl regular expression in the text box.
 - Per Script – Click Open. The Policy Script dialog box opens. Complete the Policy Script dialog box.
5. Click OK to apply the changes and close the dialog box, or click Apply to apply the changes and leave the dialog box open so you can create or edit more validation policies.

Shortcut



Click the right mouse button over the profile container icon.
From the popup menu, choose Validation Policy.

Enabling and Disabling Validation Policy

Defining a validation policy does not put it into effect; you must explicitly enable it. The status of a validation policy is indicated on the pop-up menu for the profile. You enable or disable validation policy by toggling Validation on the popup menu for profiles. When validation policy is enabled, there is a check box next to the menu item.

To enable a validation policy:

1. Select the profile for which you want to enable validation policy.
2. From the pop-up menu, choose Validation. A check box is added to the menu.

To disable validation policy:

1. Select the profile for which you want to disable validation policy.
2. From the pop-up menu, choose Validation. The check box is removed from the menu.

When validation is enabled, objects are compared with validation policy for the profile when you:

- Create an object in a profile
- Populate a profile
- Change an object that is part of a profile for which a validation policy is enabled

If an attribute fails validation, an error message is displayed.

To validate objects that are already in a profile, use Validate Profile.

When validation policy is not enabled, objects are not checked against policy.

Validating a Profile

If validation policies are enabled, you can check whether the members of a profile conform to the validation policies. To validate a profile:

1. Select the profile you want to validate.
2. From the Manage menu, choose Profiles→ Validate Profile. Enterprise SQL Server Manager checks the attributes of each member of the profile against the validation policies that are enabled. The Validation Results window displays a report that lists members and attributes that do not conform to the enabled policies. If the display is empty, all members conform to the validation policies.

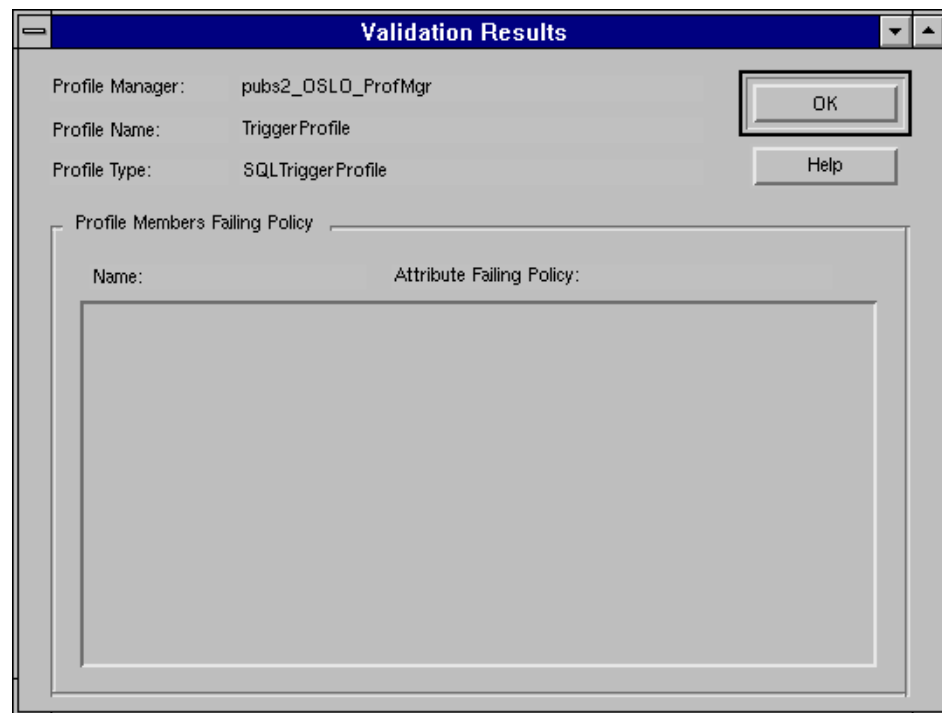


Figure 13-3: Validation Results window

Shortcut



In the left pane of the profile manager window, click the right mouse button over the profile container icon. Select Validate Profile from the popup menu.

Using Policy Scripts

Using scripts allows you to implement complex policies. For example, if you want the name, owner, and segment attributes of a new table to meet certain criteria, you could write validation scripts that check the three attributes. If you want a new object to have certain default values, you could write a default policy script.

You can create a script using any scripting language that your operating system supports. Your script should indicate the shell or interpreter in which it is to run, for example:

```
#!/bin/sh
```

Return Values in Policy Scripts

Default policies and validation policies require different kinds of scripts. A default policy must return a value for the attribute for which you are setting a default. The script must exit with 0 if it succeeds.

A validation policy must print the character strings TRUE or FALSE for the attribute you are validating. The script must exit with 0 if it succeeds.

Creating a Policy Script

The Validation Policy and Default Policy dialog boxes have an option that lets you specify a script for a policy. If you choose that option, you open the Policy Script dialog box to create or change the script.

To create a policy script:

1. In the Validation Policy dialog box or Default Policy dialog box, select the Per Script option button and click Open. The Policy Script dialog box opens.

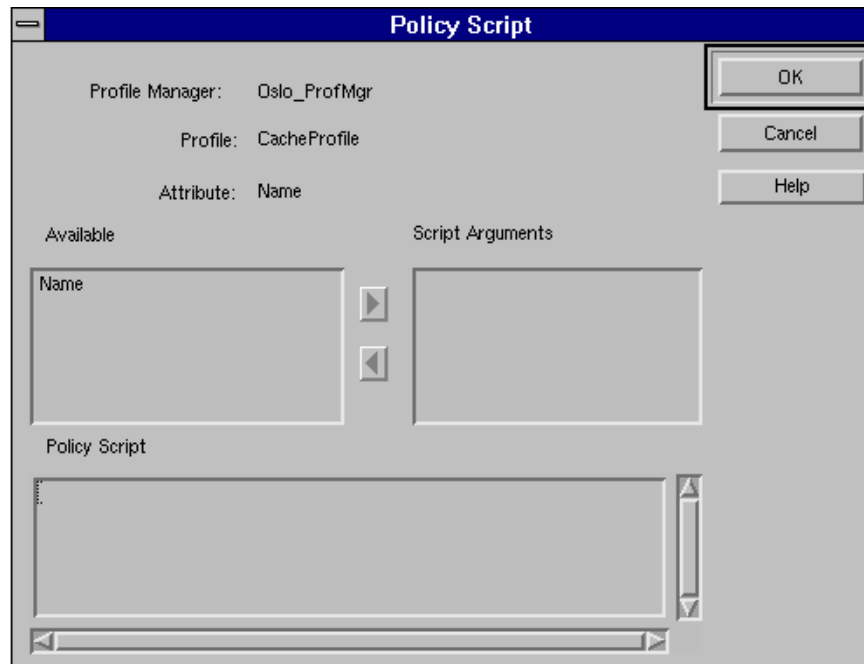


Figure 13-4: Policy Script dialog box

2. In the Available list box, select the policy script arguments that you want to use.
3. Click the right-pointing arrow button to move the selected script arguments to the Script Arguments list.
4. In the Policy Script text box, type the script.
5. Click OK to create the script. The Policy Script dialog box closes and you can complete the Validation Policy or Default Policy dialog box.

► **Note**

The Policy Script dialog box does not verify the syntax of the scripts you enter.

Using Policy Script Arguments

The script arguments you specify in the Script Arguments list of the Policy Script dialog box are available to your script in the order they are listed, from top to bottom. You can access the script arguments using the standard shell or interpreter method of argument parsing. For example, in a shell, you could access the arguments using \$1, \$2, and so on. If the arguments in the Script Arguments list are Index Name, Owner Name, and Segment Name, \$1=Index Name, \$2=Owner Name, and \$3=Segment Name.

When you write a script, be sure to use the attribute names exactly as they appear in the list (and in Table 13-2.) Include space and protect the attribute names with quotes.

► Note

If you use an attribute name that has spaces in a script, you must enclose the attribute name in double quotes.

Sample Validation Policy Script

The following scripts check a login for a minimum password length. The first version is a Perl script. The second is a Bourne shell version of the same script.

Perl Script

```
#!/usr/bin/perl
#####
#
# Filename: pwd_length
#
# Author:
#
# Date: Mon May 20 10:52:39 EDT 1996
#
# Synopsis: ESSM validation policy script that
# checks a new/modified password's length.
#
```

```

# Usage: pwd_length $password
#
#   wputpolm -v args='$Password'
@SQLLoginProfile:<profile_name> Password \
#   < pwd_length
#
#////////////////////////////////////
$min_length = 8; # password must be at least 8
characters
$passwd = shift; # grab password as input argument
if (length($passwd) < $min_length) {
    print "FALSE"; # failed validation
    exit 0;
} else {
    print "TRUE"; # passed validation
    exit 0;
}

```

Bourne Shell Script

```

#!/bin/sh
#////////////////////////////////////
#
# Filename: pwd_length.sh
#
# Author:
#
# Date: Mon May 20 10:52:39 EDT 1996
#
# Synopsis: ESSM validation policy script that
#checks a new/modified password's length.
#
# Usage: pwd_length $password
#

```

```
#    wputpolm -v args='$Password'
@SQLLoginProfile:<profile_name> Password \
#    < pwd_length.sh
#
#////////////////////////////////////
min_length=8 # password must be at least 8
characters
length=`echo $1 | wc -c` # grab length of input
argument ($password)
if [ $length -le $min_length ]
then
    echo FALSE
    exit 0
fi
echo TRUE
exit 0
```

14

Using Event Monitoring Services

What's in this Chapter

Event Monitoring Services (EMON Services) is a feature of Enterprise SQL Server Manager that lets you monitor SQL Server for an event and manage a response to the event. An event occurs when a high-level SQL Server characteristic surpasses a user-defined state or threshold. For example, you can use Event Monitoring Services to notify you when the number of network packets sent exceeds a specific number or if an error log file records an access error.

Event Monitoring Services is based on Tivoli Monitoring Technology (TMT). TMT is the same event management technology framework that Tivoli/Sentry uses. This chapter describes the SQL Server characteristics you can monitor and how to monitor for and manage events. However, because Event Monitoring Services is an extension of TMT, this chapter does not describe all of the features of TMT or Sentry profiles. For more information about using TMT and Sentry profiles, see *Tivoli/Sentry User's Guide*.

Events You Can Monitor

Event Monitoring Services provides the following types of monitoring sources:

- Generic event monitors
- Process-specific event monitors
- Error log file monitors

A **monitoring source** is a specific SQL Server characteristic that you use Event Monitoring Services to monitor. An **event** occurs when Event Monitoring Services determines that the value of a monitoring source has exceeded a threshold or achieved a state specified in a Sentry profile.

You can monitor the following SQL Server characteristics:

- **Generic events**
 - Number of connection attempts
 - Number of connections used
 - Number of device reads
 - Number of device writes
 - Number of faulty databases
 - Number of locks used
 - Number of network packets received
 - Number of network packets sent
 - Percent of connections used
 - Percent of CPU busy
 - Percent of locks used
 - Percentage of segment space available
 - SQL Server status
 - Sybase login status
 - User-specified event
- **Process-specific events**
 - Number of bad status processes
 - Number of blocked processes
 - Number of infected processes
 - Number of stopped processes
- **Error log file events**
 - Search using minimum severity level
 - Search using regular expressions
 - Search using SQL Server error codes
 - Access errors
 - alter database errors
 - Buffer manager errors
 - Bulk copy utility errors
 - Character set conversion errors

- Compile errors
- Create utilities errors
- Data server errors
- **dbcc** errors
- Descriptor manager errors
- Disk errors
- Distributed database network errors
- Drop errors
- **dump and load** errors
- Error handling and exception handling errors
- Initialization errors
- Insert errors
- Lock manager errors
- Memory manager errors
- Open database manager errors
- Page manager errors
- Parser errors
- Procedure manager errors
- Process kill errors
- Query processor errors
- Reconfiguration errors
- Recovery errors
- Sequencer errors
- Site buffer manager errors
- Sort manager errors
- *sysindexes* manager errors
- Text manager errors
- Threshold errors
- Timestamp errors
- Transaction errors
- Undo and redo errors

Before You Use Event Monitoring Services

The following sections describe what you should know before using Event Monitoring Services.

Specifying Roles

The Sentry engine runs as the UNIX user “nobody” or “root”. Therefore you must assign the “root” administrator the minimum Enterprise SQL Server Manager roles of `server` and `schema` even though you are not running Enterprise SQL Server Manager as “root”. Individual administrators do not need a specific Enterprise SQL Server Manager role. However, they must have the appropriate TME roles to use Sentry profiles. For more information, see *Tivoli/Sentry User’s Guide*.

Using the Notices Group Response Action

Event Monitoring Services allows you to send notices to the `SentryStatus` notices group. Before you can access messages in the `SentryStatus` notices group, you must subscribe to it. For more information, see *Tivoli/Sentry User’s Guide*.

Using Environment Variables in Response Scripts

One of the response actions you can perform is to execute a script when a monitoring source exceeds a threshold. To help you use scripts, TMT provides a set of environment variables that enable you to access information about an event. If your script needs to know the name of the SQL Server on which an event occurred, TMT stores the SQL Server name in the `ENDPOINT` environment variable. For more information about using environment variables in response scripts, see *Tivoli/Sentry User’s Guide* and *Enterprise SQL Server Manager Reference Manual*.

Distributing Sentry Profiles

To monitor SQL Server, you must distribute the Sentry profile to a Sybase SQL Server. Before you distribute a Sentry profile, be aware of the following:

- Distribute Sentry profiles containing SQL Server monitors only to managed SQL Servers or SQL Server Profile Managers. Do not distribute Sentry profiles to Database Profile Managers.
- If you distribute a Sentry profile to a SQL Server Profile Manager, only the subscribers to that profile are monitored. The SQL Server associated with the profile manager is not monitored.
- Event Monitoring Services does not support the TMT proxy endpoint capability. Therefore, you should not distribute Sentry profiles containing Sybase SQL Server monitoring sources to proxy endpoints.

Monitoring SQL Servers on Non-Tivoli Platforms

If Enterprise SQL Server Manager can see a SQL Server, it can monitor events that do not use the error log, even if the SQL Server is on a platform that Tivoli does not support. You must monitor the SQL Server from a platform that is running Enterprise SQL Server Manager; that is, from a platform that supports Tivoli.

Enterprise SQL Server Manager cannot monitor log file events on non-Tivoli platforms.

How to Monitor for and Manage SQL Server Events

To monitor SQL Server events, you create a Sentry profile. The Sentry profile contains the events you want to monitor, and the actions you want to take when an event occurs.

When a SQL Server monitoring source surpasses a threshold or is in a state defined in the Sentry profile, the engine responds to the event based on the response actions defined in the Sentry profile.

► *Note*

Although Event Monitoring Services uses Sentry Profiles, the profiles are based on Tivoli Monitoring Technology. Enterprise SQL Server Manager does not include the full Tivoli/Sentry product.

The general procedure is as follows; specific procedures described later in this chapter.

1. Create or open a Sentry profile.
2. Add SQL Server monitors to the profile. For a complete list of monitors, see “SQL Server Monitoring Sources” on page 14-7.
3. Schedule the monitoring of the SQL Server within the profile.
4. Distribute the profile to Sybase SQL Servers (see “Distributing Sentry Profiles” on page 14-5 for more information about distributing profiles).

When you add monitoring sources to a Sentry profile, you define the following:

- **Response level**—identifies the category of the response. You define a threshold and set of response actions for each response level. There are five default response levels: severe, critical, warning, normal, and always. Most Enterprise SQL Server Manager monitoring sources do not have default response levels. For more information see “SQL Server Monitoring Sources” on page 14-7.
- **Threshold**—identifies the threshold to use to initiate a set of response actions. You define a threshold for each response level. The threshold options that are available depend on the individual monitoring source. For more information on threshold options available for each monitoring source, see “SQL Server Monitoring Sources” on page 14-7.

- **Response action**—identifies the actions to perform when a monitor exceeds a threshold. You can set multiple response actions for each response level. Response actions include sending email to specific logins or changing the icon of the indicator collection.

Event Monitoring Services provides a set of default response levels and threshold options for some monitoring sources. For more information, see “SQL Server Monitoring Sources” on page 14-7.

For more information about response levels, threshold options, and response actions, see *Tivoli/Sentry User’s Guide*.

SQL Server Monitoring Sources

The following sections describe each of the SQL Server monitoring sources, arguments you must include in the Sentry profile, threshold options, and default response levels and thresholds.

Generic Event Monitors

Unless otherwise stated, generic event monitors do not have arguments.

Unless otherwise stated, each generic event monitor has the following threshold options:

- Greater than
- Less than
- Equal to
- Not equal to
- Increases beyond
- Decreases below
- Increase of
- % increase of
- Outside range
- Changes by

Unless otherwise stated, generic event monitors do not have a default response level or threshold.

Number of Connection Attempts

Number of Connection Attempts identifies the total number of connection attempts that have occurred since the SQL Server was started. This value includes both successful and unsuccessful connection attempts.

Number of Connections Used

Each SQL Server has a maximum number of connections that it can support. Number of Connections Used identifies the total number of connections currently in use.

Number of Device Reads

Number of Device Reads identifies the total number of device reads by the SQL Server since it was started.

Number of Device Writes

Number of Device Writes identifies the total number of device writes by the SQL Server since it was started.

Threshold Options

- Greater than
- Less than
- Equal to
- Not equal to
- Increases beyond
- Decreases below
- Increase of
- % increase of
- Outside range

Number of Faulty Databases

Number of Faulty Databases identifies the combined total of suspect and unrecoverable databases in the SQL Server. Event Monitoring Services extracts this information from the *Status* column of the *sysdatabases* table.

Default Response Level/Threshold

Response Level	Threshold
Critical	Increases beyond 0.1

Number of Locks Used

Number of Locks Used identifies the total number of locks currently in use.

Number of Network Packets Received

Number of Network Packets Received identifies the total number of network packets received by the SQL Server since it was started.

Number of Network Packets Sent

Number of Network Packets Sent identifies the total number of network packets sent by the SQL Server since it was started.

Percent of Connections Used

Percent of Connections Used identifies the percent of total connections to a SQL Server that are currently in use. Event Monitoring Services determines this value by dividing the number of connections currently in use by the maximum number of user connections available for use.

Default Response Level/Threshold

Response Level	Threshold
Critical	Increases beyond 90%
Severe	Increases beyond 75%
Warning	Increases beyond 50%

Percent of CPU Busy

Each SQL Server maintains an internal clock that counts the amount of time the server was idle and the amount of time spent processing user requests and performing I/O. Percent of CPU Busy identifies the amount of time a SQL Server was busy performing user requests during the last monitoring interval as a percent of total SQL Server CPU.

► *Note*

To determine Percent of CPU Busy, Event Monitoring Services compares information from two monitoring intervals. When you define a threshold option, be aware that Percent of CPU Busy always returns **-1** for the first monitoring interval because there is no historical information for Event Monitoring Services to compare.

Percent of Locks Used

Percent of Locks Used identifies the percent of total locks currently in use. Event Monitoring Services determines this value by dividing the number of locks currently in use by the maximum number of locks available for use.

Percent of Segment Space Available

Percent of Segment Space Available identifies the percent of total segment space available for use. You specify an upper and lower boundary of available space to monitor for. The return value is the number of segments that satisfy the bounding criteria.

Arguments

- *lower_bound* - minimum percent of space available
- *upper_bound* - maximum percent of space available
- *database_name*—name of the database you want to monitor (optional)
- *segment_name*—name of the segment you want to monitor (optional)

Use of optional arguments works as follows:

- If you do not specify a database, Event Monitoring Services monitors all segments on all databases.
- If you specify a database, but not a segment, Event Monitoring Services monitors all segments in the database.
- If you specify a segment, you must specify a database.

SQL Server Status

SQL Server Status identifies the status of SQL Server.

Threshold Options

- SQL Server is up
- SQL Server is down
- SQL Server becomes unavailable
- SQL Server becomes available

Default Response Level/Threshold

Response Level	Threshold
Critical	SQL Server becomes unavailable

Sybase Login Status

Sybase Login Status monitors changes to the association between a Tivoli administrator and a SQL Server login and password. If the password for the associated SQL Server login changes, Event Monitoring Services makes the specified response.

In order for a Tivoli administrator to log in to SQL Server, you must create a link between the Tivoli administrator and a login and password in SQL Server. If the password for the SQL Server login changes, the link is disrupted and the administrator cannot log in to SQL Server.

You link a Tivoli administrator to a SQL Server login using the SQL Server Logins for Administrator dialog box or by using the `ssetsybaselogin` command. For more information about the SQL Server Logins for Administrator dialog box, see “Assigning Enterprise SQL Server Manager Roles to Administrators” in Chapter 2, “Setting Up the Tivoli Management Environment for ESSM”. For more information about `ssetsybaselogin`, see *Enterprise SQL Server Manager Reference Manual*.

Arguments

`unix_login`—UNIX login that is linked to a SQL Server login.

Threshold Options

- Sybase login is valid
- Sybase login is invalid
- Sybase login becomes invalid
- Sybase login becomes valid

Default Response Level/Threshold

Response Level	Threshold
Critical	Sybase login becomes invalid

User-Specified Event

A user-specified event gives you the flexibility to create customized events that monitor events of particular interest to you. For example, you could monitor for the number of rows in a table.

Arguments

SQL Query – a SQL statement that returns a numeric value.

Process-Specific Monitors

Except for Number of Blocked Processes, process-specific monitors do not have arguments.

Each process-specific monitor has the following threshold options:

- Greater than
- Less than
- Equal to
- Not equal to
- Increases beyond
- Decreases below
- Increase of
- % increase of
- Outside range
- Changes by

Process-specific monitors do not have a default response level or threshold.

Number of Bad Status Processes

Identifies the number of processes with a bad status in SQL Server.

Number of Blocked Processes

Identifies the number of blocked processes in SQL Server.

Argument

- Minimum blocked time (in seconds)

Number of Infected Processes

Identifies the number of infected processes in SQL Server.

Number of Stopped Processes

Identifies the number of stopped processes in SQL Server.

Error Log File Monitors

Error log file monitors let you monitor the SQL Server Error Log file for specific criteria or for specific types of errors. To monitor an error log, the management host and SQL Server host must be the same machine. You cannot monitor error log entries in SQL Server running on a machine that is not supported by the TME. Each monitor lets you specify the following options:

- Scan Back Time (seconds) – scans the error log beginning at the end of the error log file less the scanback time. See Figure 14-1.
- Start Time (HH:MM:SS) – monitors the error log starting at the specified time.
- Start Date (MM/DD/YY) – monitors the error log starting at the specified date.

► **Note**

To use the Start Time and Start Date options, set Scan Back Time to 0.

In all cases, the monitor stops when it finds the first error.

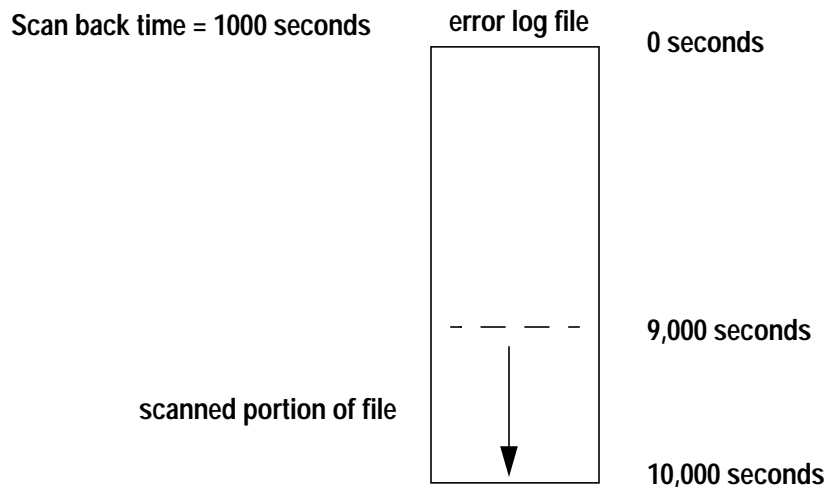


Figure 14-1: Illustration of scan back time

Search Using SQL Server Error Code(s)

You can monitor the error log for specific error codes.

Argument

- Error code(s) to search for

Search Using Minimum Severity Level

You can monitor the error log for a minimum severity level.

Argument

- Minimum severity level. Range: 10 - 24

Search Using Regular Expression

You can monitor the error log for a regular expression.

Argument

- Regular expression to search for

Search for Error Types

You can monitor the error log for error types. Error types and error codes are listed in the following table:

Table 14-1: Error codes monitored by Event Monitoring Services

Error Type	Error Codes
Access errors	601, 603, 605, 611, 623, 624, 625, 629, 631, 632, 644
alter database errors	5006, 5013
Buffer manager errors	803, 804, 806, 813, 821, 822, 823, 840
Bulk copy utility errors	4806
Character set conversion errors	2401, 2402, 2409
Compile errors	403, 404, 428, 430
Create utilities errors	1702, 1808, 1809, 1810, 1811, 1813, 1902, 1903, 1904, 1916
Data server errors	4002
dbcc errors	2501, 2502, 2503, 2506, 2507, 2509, 2510, 2511, 2513, 2514, 2517, 2519, 2520, 2521, 2524, 2525, 2526, 2529, 2540, 2541, 2543, 2544, 2546, 2550, 2558, 2559, 2571, 2572, 2573, 2596, 7902, 7930, 7939, 7940, 7948, 7949
Descriptor manager errors	8201, 8203, 8204, 8207, 8211
Disk errors	5115, 5123
Distributed database network errors	7205, 7211, 7212, 7124, 7215, 7218, 7220, 7221, 7223, 7227
Drop errors	3701, 3702, 3703
dump and load errors	3002, 3004, 3101, 3104, 3105, 3120, 3201, 3202, 3203, 3225, 4204, 4207, 4219, 4305, 4306
Error handling and exception handling errors	3604, 3621, 3626
Initialization errors	1601, 1602, 1605, 1608, 1613, 1623, 2714
Insert errors	2601, 2610, 2615, 2620
Lock manager errors	1203, 1204, 1205, 1207, 1265
Memory manager errors	701, 702, 703, 706, 707
Open database manager errors	903, 905, 906, 911, 913, 916, 921, 924, 925, 926, 930, 933, 945

Table 14-1: Error codes monitored by Event Monitoring Services (continued)

Error Type	Error Codes
Page manager errors	1105, 1108, 1117, 1120, 1129, 1133, 1134, 1142, 1143
Parser errors	102, 7364
Procedure manager errors	2811, 2812, 2824
Process kill errors	6103
Query processor errors	511, 515
Reconfiguration errors	5808, 5813
Recovery errors	3403, 3414, 3418, 3425
Sequencer errors	207, 208, 213, 225, 226, 229, 232, 259, 268
Site buffer manager errors	8704
Sort manager errors	1501, 1505, 1508, 1509, 1510, 1520, 1530, 1531
<i>sysindexes</i> manager errors	8402, 8412
Text manager errors	7130
Threshold errors	7401, 7402, 7403, 7404, 7405, 7406, 7407, 7408, 7409, 7410, 7411, 7412, 7413, 7414, 7415, 17417, 17715, 17716, 17870, 17871, 17873, 17874, 17875, 17903, 17904, 17905, 17906, 17910, 18031, 18032, 18033
Timestamp errors	6902
Transaction errors	3904
Undo and redo errors	3307

Setting the Error Log File Attribute

To specify an error log file, use the `-errorlog` argument to the `ssetserver` command or the `smanageserver` command. For more information, see the *Enterprise SQL Server Manager Reference Manual*.

Using Sentry Profiles to Monitor for SQL Server Events

The following sections describe how to add monitoring sources to a Sentry profile.

Adding Monitoring Sources to a Sentry Profile

TMT has both a graphical user interface (GUI) and a command line interface (CLI). This section describes how to add Sybase SQL Server monitoring sources to a Sentry profile using the GUI. See “Using the waddmon Command” on page 14-26 to use the CLI to add monitoring sources to a Sentry profile. For more detailed information, see *Tivoli/Sentry User’s Guide*.

To monitor for SQL Server events:

1. In a policy region, add SentryProfile and Profile Manager to the list of managed resources. See Chapter 8, “Getting Started” in *Enterprise SQL Server Manager Installation and Planning Guide* for the steps in this task.
2. In the policy region, create a Profile Manager:

From the Create menu, select Profile Manager. The Create Profile Manager dialog box opens.

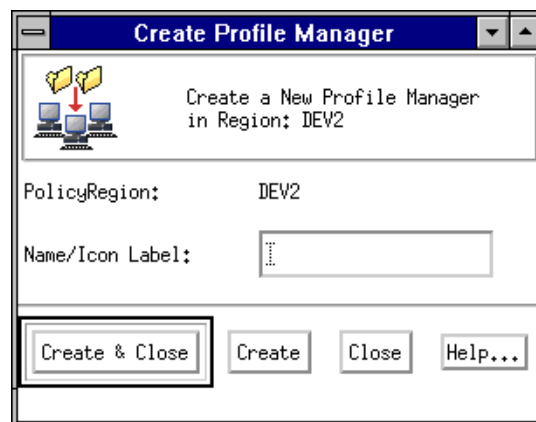


Figure 14-2: Create Profile Manager dialog box

In the Name/Icon Label box, type a name for the profile manager.

Click Create & Close. The profile manager is added to the policy region window.

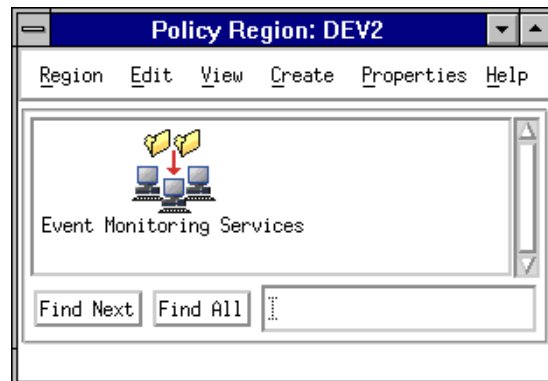


Figure 14-3: Policy region window

3. Add a Sentry Profile to the profile manager:

Open the profile manager. The Profile Manager window opens. From the Create menu, select Profile. The Create Profile dialog box opens.

In the Name/Icon Label box, type a name for the profile.

In the Type box, select SentryProfile.

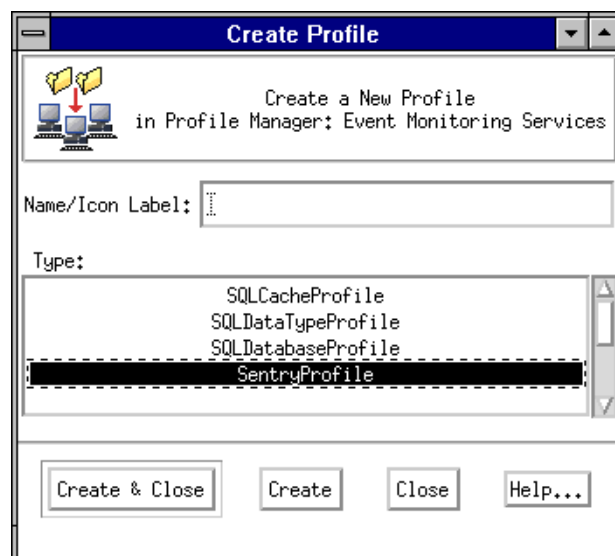


Figure 14-4: Create Profile dialog box

To create just one profile, click Create & Close. To create more than one profile, click Create, then continue creating additional profiles.

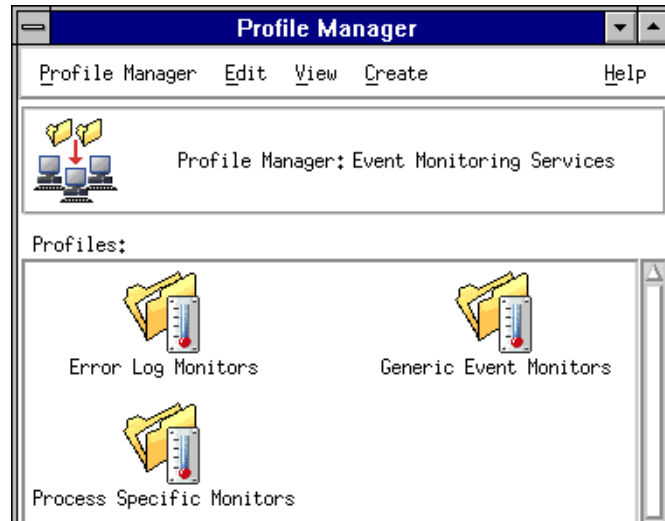


Figure 14-5: Profiles in a Profile Manager window

4. Add monitors to the profile. See the following section, “Adding Monitors to a Profile”.
5. Add subscribers to the subscription list.

After you distribute the Sentry profile to the subscribing SQL Servers, TMT monitors the Sybase SQL Server sources in the profile and manages the responses when an event occurs.

Adding Monitors to a Profile

To add monitors to a profile:

1. Double click the profile to which you want to add monitors. The Sentry Profile Properties dialog box opens.

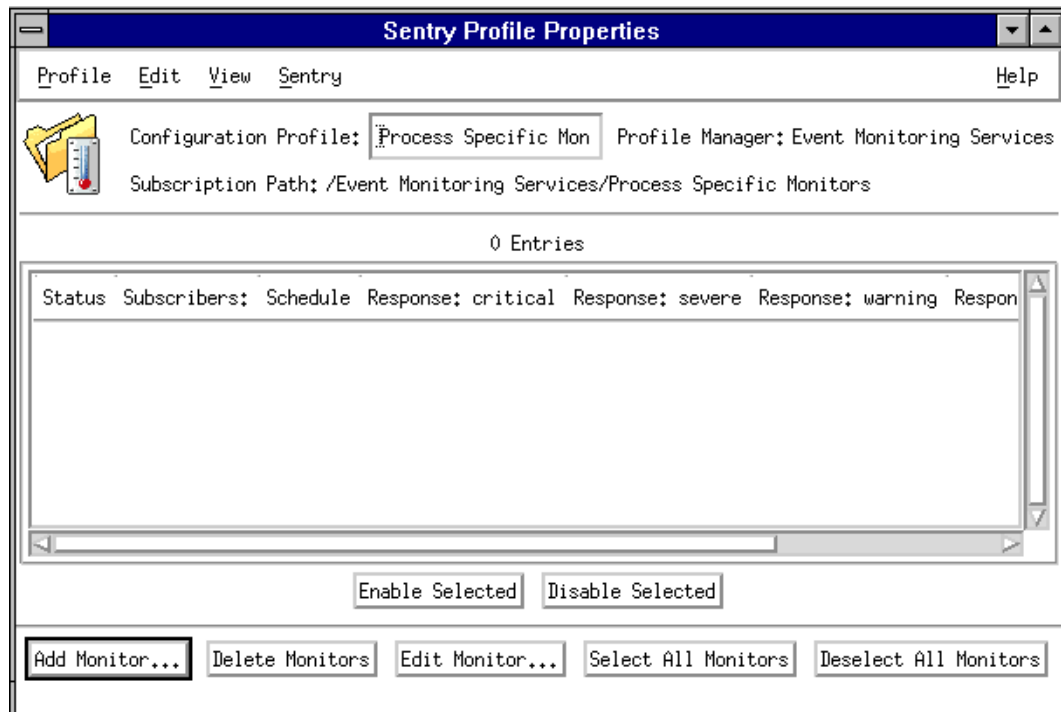


Figure 14-6: Sentry Profile Properties dialog box

2. Choose Add Monitor. The Add Monitor to Tivoli/Sentry Profile dialog box appears. The Monitoring Collections box lists the types of event monitors:
 - SQL Server Generic EMON Services
 - SQL Server Log File EMON Services
 - SQL Server Process EMON Services
3. Select the monitoring collection you want. The Monitoring Sources box lists the monitoring sources for the collection that is selected in the Monitoring collections box. If a monitoring source has arguments, they are listed in the Monitor Arguments box.

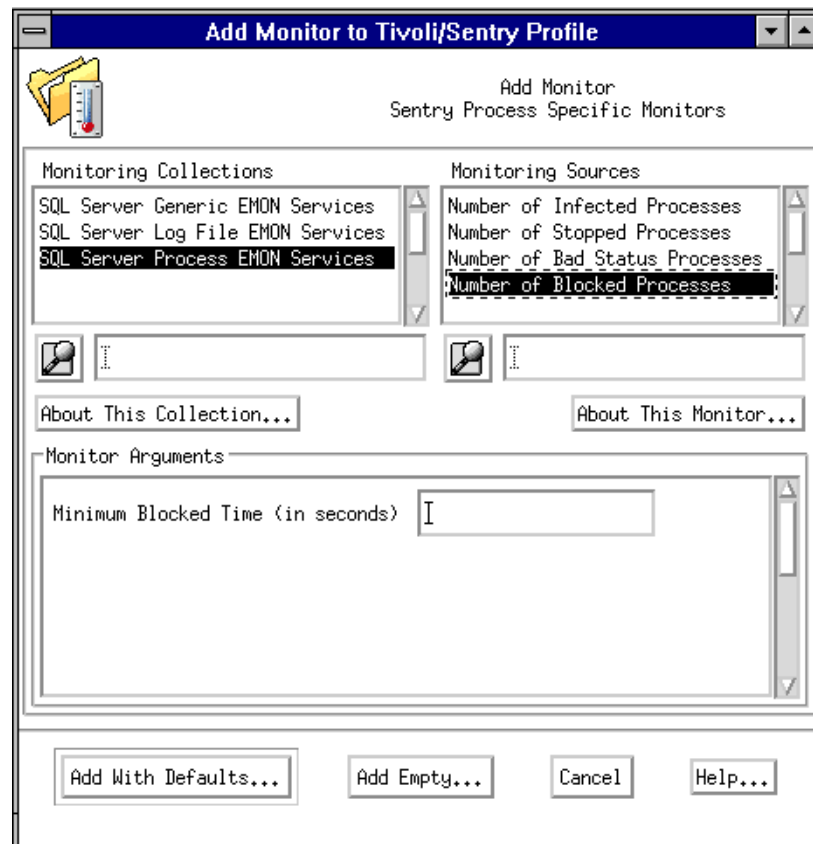


Figure 14-7: Add Monitor to Tivoli/Sentry Profile dialog box

4. Select the monitoring collection form which you want to select a monitoring source.
5. Select the monitoring source you want to use.
6. Choose Add With Defaults or Add Empty.
 - Add With Defaults—adds the monitoring source to the profile with the default response levels and threshold options. Event Monitoring Services does not provide all monitoring sources with default values. However, you can define defaults for any monitoring source. For more information, see *Tivoli/Sentry User's Guide*. For more information about the defaults that Event Monitoring Services provides, see “SQL Server Monitoring Sources” on page 14-7.
 - Add Empty—adds the monitoring source to the profile without using the default response levels and threshold options.

After you choose Add with Defaults or Add Empty, the Edit Sentry Monitor dialog box appears (Figure 14-8.)

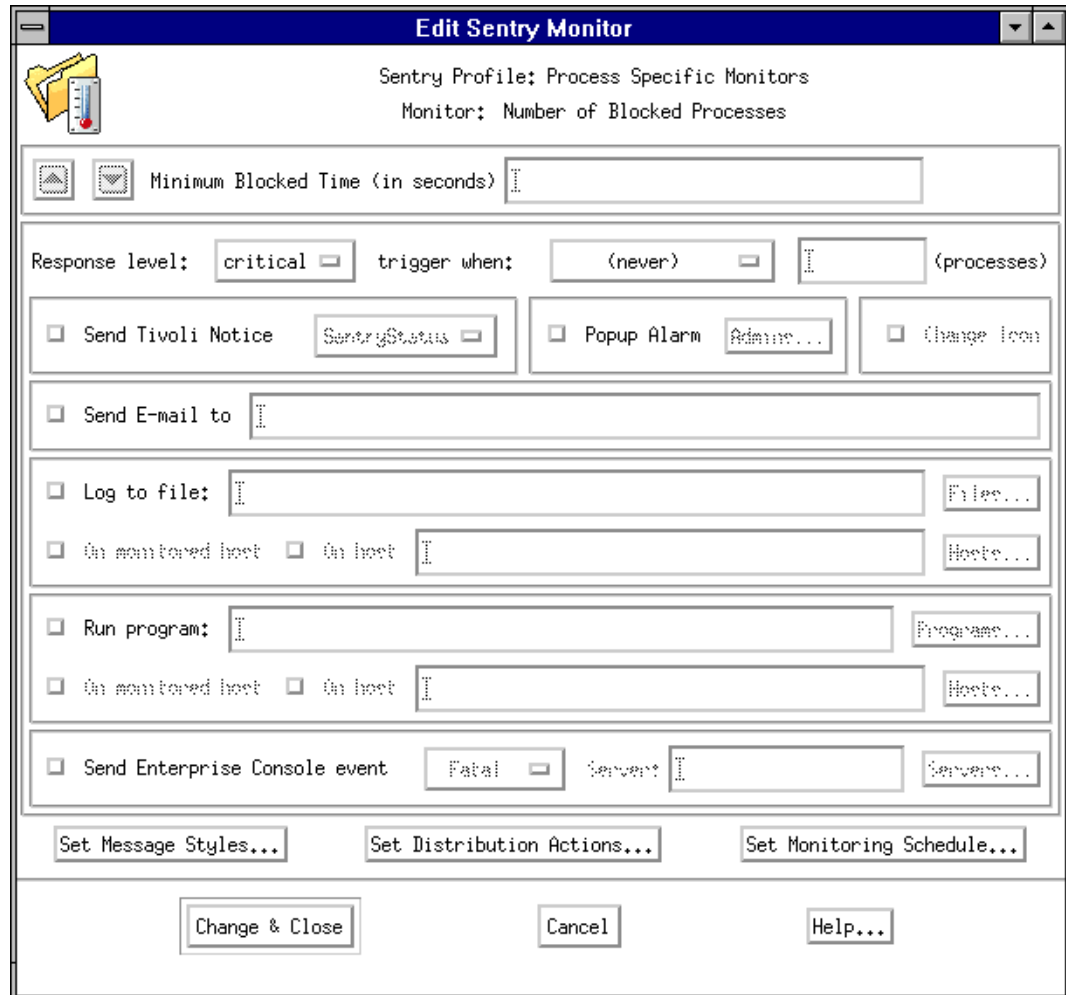


Figure 14-8: Edit Sentry Monitor dialog box

7. In the Edit Sentry Monitor dialog box:

- Define the response levels, thresholds, and response actions you want to use.
- Define the monitoring schedule you want to use.
- If you have not yet done so, supply any required arguments.
- For more information, see *Tivoli/Sentry User's Guide*.

► **Note**

If you use the response action “Run Program”, both the script and the directory in which it is located must have the permissions 777.

8. Choose Change & Close.
9. Repeat this procedure each monitoring source you want to use in the profile.

Subscribing to a Sentry Profile

After you set up the a profile manager Sentry Profiles for monitoring SQL Server events, add subscribers to the profile manager. See *Tivoli Management Platform User's Guide* for information about subscribing to a Tivoli profile manager.

Unsubscribing from a Sentry Profile

If you unsubscribe SQL Server from a Sentry profile, a message dialog box asks you if you want to delete all profile copies or retain them. You must delete the profile copies.

Using the waddmon Command

The TMT command line interface (CLI) provides a set of commands that allow you to create and manage Sentry profiles. To add monitoring sources to a Sentry profile, you use the `waddmon` command. For information about the `waddmon` command, see *Tivoli/Sentry User's Guide*.

When you use `waddmon` with Event Monitoring Services, you must specify the name of the monitoring collection and the name of the monitoring source to use. If the monitoring source has arguments, you must also include the appropriate information.

The monitoring collections are:

- SQL Server Generic EMON Services
- SQL Server Process EMON Services
- SQL Server Log File EMON Services

For example, to add SQL Server Status to a Sentry profile with `waddmon`, issue the following command:

```
waddmon "SQL Server Generic EMON Services" "ServerStatus" profile_name
```

where *profile_name* is the name of the Sentry profile.

If you are using a monitoring source that requires an argument, use the `-a` flag in the `waddmon` command. For example, the following command adds the Sybase Login Status monitoring source to a Sentry profile:

```
waddmon "SQL Server Generic EMON Services" "AdminLP" -a "susie"
profile_name
```

where *profile_name* is the name of the Sentry profile.

For more information on `waddmon` and other commands of the TMT CLI, see *Tivoli/Sentry User's Guide*.

Table 14-2 identifies the name of each Event Monitoring Services monitoring source:

Table 14-2: Monitoring Source Names

Monitoring Source	CLI Monitor
Generic Event Monitors	
Number of connections used	Logins
Number of connection attempts	Connects
Number of device reads	DiskReads
Number of device writes	DiskWrites
Number of faulty databases	DbFaulty
Number of locks used	Locks
Number of network packets received	NetReads
Number of network packets sent	NetWrites
Percent of connections used	ConnectsRate
Percent of CPU busy	CpuRate
Percent of locks used	LocksRate
Percentage of segment space available	SegSize
SQL Server status	ServerStatus
Sybase Login status	AdminLP
User-specified event	CustomEvent

Table 14-2: Monitoring Source Names (continued)

Monitoring Source	CLI Monitor
Process-specific Event Monitors	
Number of bad status processes	ProcessBad
Number of blocked processes	ProcessBlock
Number of infected processes	ProcessInfect
Number of stopped processes	ProcessStop
Error Log File Events	
Search using minimum severity level	LogMiniSeverity
Search using regular expression	LogRegexp
Search using SQL SERVER error code(s)	LogMatchError
Access errors	LogAccess
alter database errors	LogAlterDb
Buffer manager errors	LogBufferMgr
Bulk copy utility errors	LogBcp
Character set conversion errors	LogCharSetConv
Compile errors	LogCompile
Create utilities errors	LogCreateUtil
Data server errors	LogDataServer
dbcc errors	LogDbcc
Descriptor manager errors	LogDescMgr
Disk errors	LogDisk
Distributed database network errors	LogDistDb
Drop errors	LogDrop
dump and load errors	LogDumpLoad
Error handling and exception handling errors	LogErrAndExcept
Initialization errors	LogInit
Insert errors	LogInsert
Lock manager errors	LogLockMgr
Memory manager errors	LogMemoryMgr
Open database manager errors	LogOpenDbMgr
Parser errors	LogParser

Table 14-2: Monitoring Source Names (continued)

Monitoring Source	CLI Monitor
Page manager errors	LogPageMgr
Procedure manager errors	LogProcMgr
Process kill errors	LogProcessKill
Query processor errors	LogQueryProc
Reconfiguration errors	LogReConfig
Recovery errors	LogRecovery
Sequencer errors	LogSequencer
Site buffer manager errors	LogSiteBufMgr
Sort manager errors	LogSortMgr
<i>sysindexes</i> manager errors	LogSysIndexMgr
Text manager errors	LogTextMgr
Threshold errors	LogThreshold
Timestamp errors	LogTimeStamp
Transaction errors	LogTransaction
Undo and redo errors	LogUndoRedo

Troubleshooting Event Monitoring Services

If you believe that Event Monitoring Services is not functioning properly, check the Event Monitoring Services log file and the Tivoli Notices bulletin board for error messages.

Using the Event Monitoring Services Log File

If Event Monitoring Services encounters errors while communicating with SQL Server, it writes error messages to the SentryStatus group in the Notices bulletin board. The messages sent to the bulletin board are marked `EMON ERROR` in the header.

If you want to get more information about Event Monitoring errors than is provided by the messages sent to the Notices bulletin board, you can send the Event Monitoring error messages to a log file by defining the `EMON_LOG` environment variable.

The format for setting the `EMON_LOG` environment variable is:

```
EMON_LOG=pathname
```

where *pathname* is a fully qualified pathname of the log file; for example, `/tmp/essm/EMON_log`.

The directory in which the log file is placed must have world access.

To define `EMON_LOG`, use the Tivoli `odadmin environ` command. For information about using `odadmin` to define environment variables, see the *Tivoli Management Platform Guide*.

Your changes to the Event Monitoring Services log file do not take effect until you stop the Sentry engine, then restart it.

To stop the Sentry engine, enter the following command:

```
wstopeng
```

To start the Sentry engine, use the following command:

```
wlseng -l
```

Using the Notices Bulletin Board

When the Sentry engine encounters errors while communicating with Event Monitoring Services, it writes error messages to the SentryStatus group in the Notices bulletin board. For more information about the error TMT encountered, see the SentryStatus notices group.

Before you can access messages in the SentryStatus notices group, you must subscribe to it. For more information, see the *Tivoli/Sentry User's Guide*.

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 release 10.x parameters, along with a brief description of each.
- An alphabetical list of SQL Server release 11.0 parameters, along with a brief description of each.
- A list of SQL Server release 11.0 parameters grouped according to the area of SQL Server behavior they affect.

For more detailed information about each configuration parameter, see *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 Release 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 Enterprise SQL Server Manager. 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 Enterprise SQL Server Manager 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 Enterprise SQL Server Manager.
- **default sortorder id** is the number of the sort order that is the current default on this Enterprise SQL Server Manager. **Do not change this parameter.** See *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%.

- **i/o accounting flush interval** specifies how many disk I/Os to accumulate before flushing the data to *syslogins* for use in chargeback accounting.
- **i/o 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 Enterprise SQL Server Manager 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 Enterprise SQL Server Manager. 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 Enterprise SQL Server Manager. 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 Enterprise SQL Server Manager. The default is 20.
- **number of remote logins** controls the number of active user connections from this Enterprise SQL Server Manager to remote servers. The default is 20.
- **number of remote sites** controls the number of simultaneous remote sites that can access this Enterprise SQL Server Manager. 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 Enterprise SQL Server Manager 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 Enterprise SQL Server Manager 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 Enterprise SQL Server Manager uses to control access to individual partitions of a table. Enterprise SQL Server Manager 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 Enterprise SQL Server Manager displays on the console during recovery. The default run value is 0, which means that Enterprise SQL Server Manager 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 Enterprise SQL Server Manager's memory needs are met. The default run value is 20.
- `recovery interval in minutes` sets the maximum number of minutes per database that Enterprise SQL Server Manager 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 Enterprise SQL Server Manager'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 Enterprise SQL Server Manager'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 Enterprise SQL Server Manager 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

In SQL Server release 11.0, 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 Enterprise SQL Server Manager Icons

What's in this Appendix

This appendix is a guide to the icons used in Enterprise SQL Server Manager and 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 Enterprise SQL Server Manager.

Table 2-1: Enterprise SQL Server Manager icons






Icon	Object
	Container
	Database
	Database device
	Default
	Dump device

Table 2-1: Enterprise SQL Server Manager icons (continued)
















Icon	Object
	Group
	Index
	Login
	Named Cache
	Procedure
	Remote server
	Rule
	Segment
	SQL Server (unconnected)
	Table (system)

Table 2-1: Enterprise SQL Server Manager icons (continued)

Icon	Object
	Table (user)
	Trigger
	User
	User datatype
	View

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.

► **Note**

You cannot drag-and-drop objects between TME windows and a SQL Server window or ESSM profile manager windows. For example, you cannot drag the icon of a SQL Server from a policy region window into a SQL Server Profile Manager window to add it as a subscriber to a profile.

Table 2-2: 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

Table 2-2: Icon drag-and-drop operations (continued)































Operation	Icon 1	Icon 2	Opened 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
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

Table 2-2: Icon drag-and-drop operations (continued)







Operation	Icon 1	Icon 2	Opened dialog box
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
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
Distribute a profile	 	 	Distribute Profile dialog box



SQL Server Release 11.0 Drag-and-Drop Operations

When you connect to SQL Server release 11.0, the following additional drag-and-drop operations are valid:

Table 2-3: SQL Server release 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
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

C

SQL Server Profile Operations

What's in this Appendix

This appendix lists the distribution order for Enterprise SQL Server Manager profiles and the operations that take place when each Enterprise SQL Server Manager profile type is distributed.

Profile Distribution Order

SQL Server profiles are distributed in the following order:

- SQLServerProfile
- SQLDumpDeviceProfile
- SQLDbDeviceProfile
- SQLDatabaseProfile
- SQLLoginProfile
- SQLRemoteServerProfile
- SQLCacheProfile

Database profiles are distributed in the following order:

- SQLDatabaseProfile
- SQLSegmentProfile
- SQLGroupProfile
- SQLUserProfile
- SQLRuleProfile
- SQLDefaultProfile
- SQLDataTypeProfile
- SQLTableProfile
- SQLIndexProfile
- SQLViewProfile
- SQLProcedureProfile
- SQLTriggerProfile

Profiles in a SQL Server Profile Manager

This section describes what happens when each profile type in a SQL Server Profile Manager is distributed.

SQLServerProfile

When a SQL Server profile is distributed, it modifies the SQL Server configuration of the subscribing SQL Server. If you distribute a change to a static configuration parameter, the subscribing SQL Server must be rebooted to put the change into effect.

Distributing a SQLServerProfile does not depend on the successful distribution of other profiles.

SQLDumpDeviceProfile

When a dump device profile is distributed, it:

- Creates a dump device if the dump device that is being distributed does not exist in the target server.
- Modifies a dump device if the dump device exists in the target server, but is different from the dump device being distributed.
- Deletes a dump device if the dump device exists in the target server, but was deleted in the source server.

Distributing a SQLDumpDeviceProfile does not depend on the distribution of other profiles.

SQLDbDeviceProfile

When a database device profile is distributed, it:

- Creates a database device if the database device that is being distributed does not exist in the target server.
- Modifies a database device if the database device exists in the target server, but is different from the database device being distributed.
- Deletes a database device if the database device exists in the target server, but was deleted in the source server.

Distributing a SQLDbDeviceProfile does not depend on the distribution of other profiles.

SQLDatabaseProfile

When a database profile is distributed from a SQL Server Profile Manager it:

- Creates a database if the database that is being distributed does not exist in the target server.
- Modifies database options.
- Deletes a database if the database exists in the target server, but was deleted in the source server.

Distributing a SQLDatabaseProfile may depend on the successful distribution of SQLDbDeviceProfiles.

SQLLoginProfile

When a login profile is distributed, it:

- Creates a login if the login that is being distributed does not exist in the target server.
- Modifies a login if the login exists in the target server, but is different from the login being distributed.
- Deletes a login if the login exists in the target server, but was deleted in the source server.

Distributing a SQLLoginProfile may depend on the successful distribution of SQLDatabaseProfile profiles.

SQLRemoteServerProfile

When a remote server profile is distributed, it:

- Creates a remote server if the remote server that is being distributed does not exist in the target server.
- Modifies a remote server if the remote server exists in the target server, but is different from the remote server being distributed.
- Deletes a remote server (and remote logins) if the remote server exists in the target server, but was deleted in the source server.

Distributing a SQLRemoteServerProfile does not depend on the distribution of other profiles.



SQLCacheProfile

When you distribute a cache profile, it:

- Creates a cache if the cache that is being distributed does not exist in the target server. Only the cache configuration information gets distributed. Binding and buffer pool information does not get distributed. The cache does not take effect until you stop and restart the SQL Server.
- Modifies a cache if the cache exists in the target server, but is different from the cache being distributed.
- Deletes a cache if the cache exists in the target server, but was deleted in the source server. The deletion does not take effect until you stop and restart the target SQL Server.

Although a SQLCacheProfile is considered a SQL Server profile, distributing it depends on database profiles. Distributing a SQLCacheProfile depends on the successful distribution of SQLDatabaseProfiles, SQLIndexProfiles, and SQLTableProfiles.

Profiles in a Database Profile Manager

This section describes what happens when each profile type in a Database Profile Manager is distributed.

SQLDatabaseProfile

When a database profile is distributed from a Database Profile Manager, it modifies database options.

Distributing a SQLDatabaseProfile does not depend on the successful distribution of other profiles. To create or delete a database through profile management, use a SQLDatabaseProfile in a SQL Server Profile Manager.

SQLSegmentProfile

When a segment profile is distributed, it:

- Creates a segment if the segment that is being distributed does not exist in the target server. The database device on which the segment is being created must already exist.
- Modifies a segment if the segment exists in the target server, but is different from the segment being distributed.
- Deletes a segment if the segment exists in the target server but was deleted in the source server.

Distributing a SQLSegmentProfile may depend on the successful distribution of SQLDatabaseProfiles.

SQLGroupProfile

When a group profile is distributed, it:

- Creates a group if the group that is being distributed does not exist in the target server.
- Deletes a group if the group exists in the target server, but was deleted in the source server.

Distributing a SQLGroupProfile may depend on the successful distribution of SQLDatabaseProfile profiles.

SQLUserProfile

When a user profile is distributed, it:

- Creates a user if the user that is being distributed does not exist in the target server.
- Modifies a user if the user exists in the target server, but is different from the user being distributed.
- Deletes a user if the user exists in the target server, but was deleted in the source server.

Distributing a SQLUserProfile may depend on the successful distribution of SQLDatabaseProfile and SQLGroupProfile profiles.

SQLRuleProfile

When a rule profile is distributed, it:

- Creates a rule if the rule that is being distributed does not exist in the target database. Any bindings to table columns and user-datatypes are not created.
- Changes a rule if the rule exists in the target database, but is different from the rule being distributed. Any changes to bindings to table columns and user-datatypes are not updated.
- Deletes a rule if the rule exists in the target database, but was deleted in the source database. If the rule is bound to a table column or user-datatype, the binding is dropped before the rule is deleted.

Distributing a SQLRuleProfile does not depend on the successful distribution of any other profile.

SQLDefaultProfile

When a default profile is distributed, it:

- Creates a default if the default that is being distributed does not exist in the target database. Any bindings to table columns and user-datatypes are not created.
- Changes a default if the default exists in the target database, but is different from the default being distributed. Any changes to bindings to table columns and user-datatypes are not updated.
- Deletes a default if the default exists in the target database, but was deleted in the source database. If the default is bound to a table column or user-datatype, the binding is dropped before the rule is deleted.

Distributing a SQLDefaultProfile does not depend on the successful distribution of any other profile.

SQLDataTypeProfile

When a user-datatype profile is distributed, it:

- Creates a user-datatype if the type that is being distributed does not exist in the target database. Bindings to rules and defaults are also distributed. All binding processing is applied to existing data, but not to future data.
- Changes the default and rule bindings for a user-datatype. All binding processing is applied to existing data, but not to future data.
- Deletes a user-datatype if the user-datatype exists in the target database, but was deleted in the source database. If the user-datatype is bound to a rule or default, the binding is dropped before the user-datatype is deleted.

Distributing a SQLDataTypeProfile depends on the distribution of SQLRuleProfiles and SQLDefaultProfiles.

SQLTableProfile

When a table profile is distributed, it:

- Creates a table if the table that is being distributed does not exist in the target database. Creates indexes and triggers associated with the table, rule and default table binding changes, and constraints. All binding processing is applied to existing data, but not to future data.
- Changes a table if the table exists in the target database, but is different from the table being distributed. Rule and default table binding changes and constraints are also propagated. Changes to indexes and triggers are not propagated via this profile. You must use index and trigger profiles to distribute changes to those objects. All binding processing is applied to existing data, but not to future data.
- Deletes a table if the table exists in the target database, but was deleted in the source database. Any bindings to the table are dropped before the table is deleted.

Distributing a SQLTableProfile may depend on the successful distribution of SQLDataTypeProfiles, SQLDefaultProfiles, and SQLRuleProfiles.

SQLIndexProfile

When an index profile is distributed, it:

- Creates an index if the index that is being distributed does not exist in the target database.
- Changes an index if the index exists in the target database table, but is different from the index being distributed.
- Deletes an index if the index exists in the target database table, but was deleted in the source database table.

Distributing a SQLIndexProfile may depend on the successful distribution of SQLTableProfiles.

SQLViewProfile

When a view profile is distributed, it:

- Creates a view if the view that is being distributed does not exist in the target database.
- Changes a view if the view exists in the target database, but is different from the view being distributed.
- Deletes a view if the view exists in the target database, but was deleted in the source database.

Distributing a SQLViewProfile may depend on the successful distribution of SQLTableProfiles.

SQLProcedureProfile

When a procedure profile is distributed, it:

- Creates a procedure if the procedure that is being distributed does not exist in the target database.
- Changes a procedure if the procedure exists in the target database, but is different from the procedure being distributed.
- Deletes a procedure if the procedure exists in the target database, but was deleted in the source database.

Distributing a SQLProcedureProfile may depend on the successful distribution of SQLTableProfiles, SQLViewProfiles, and SQLDataTypeProfiles.

SQLTriggerProfile

When a trigger profile is distributed, it:

- Creates a trigger if the trigger that is being distributed does not exist in the target database.
- Changes a trigger if the trigger exists in the target database, but is different from the trigger being distributed.
- Deletes a trigger if the trigger exists in the target database, but was deleted in the source database.

Distributing a SQLTriggerProfile may depend on the successful distribution of SQLTableProfiles, SQLViewProfiles, SQLProcedureProfiles and SQLDataTypeProfiles.

D

Enterprise SQL Server Manager Role Requirements

The tables in this appendix list Enterprise SQL Server Manager roles required for the following procedures:

- Creating objects
- Changing object properties
- Deleting objects
- Displaying object properties
- Managing Enterprise SQL Server Manager resources (SQL Server, profile managers, profiles and policy)
- Miscellaneous procedures

Notes:

- The requirement “any” means you must have at least one role of the column type. For TME roles, an administrator should always have user role.
- To manipulate objects (create, delete, modify, and so on) that are not owned by “dbo”, you must either be the Database Owner or have the System Administrator role.
- To manipulate objects from a profile manager, an administrator needs server role in addition to the roles listed in this appendix.

Role Requirements for Creating Objects

Table D-1: Roles required to create objects

Object	TME	ESSM	SQL Server
named cache	any	cache	System Administrator
database	any	space	System Administrator
database device	any	space	System Administrator
default	any	schema	Database Owner
dump device	any	space	System Administrator
group	any	security	System Administrator or Database Owner
index	any	schema	Table owner
SQL Server login	any	security	System Security Officer
procedure	any	schema	Database Owner
profile	senior, or super	none	none
remote server	any	security	System Security Officer
rule	any	schema	Database Owner. Must be object owner to bind a rule.
segment	any	space	System Administrator
table	any	schema	Database Owner
threshold	any	space	System Administrator
trigger	any	schema	Table owner
user	any	security	System Administrator or Database Owner
user datatype	any	schema	any
view	any	schema	Database Owner

Role Requirements for Changing SQL Server and Database Object Properties

Table D-2: Roles required to change object properties

Activity	TME	ESSM	SQL Server
named cache	any	cache	System Administrator
database options	any	none	System Administrator
database	any	space	System Administrator
database device	user	space	System Administrator
default	any	schema	Database Owner
dump device	any	space	System Administrator
group	any	security	System Administrator or Database Owner
index	any	schema	Table owner and index owner
login	any	security	System Security Officer and System Administrator
procedure	any	schema	Database Owner
remote logins	any	security	System Security Officer and System Administrator
remote server	any	security	System Security Officer
rule	any	schema	Database Owner
segment	any	space	System Administrator
table	any	schema	Database Owner
a threshold	any	space	System Administrator
trigger	any	schema	Owner
user properties including object and command permissions	any	security	System Administrator or Database Owner
view	any	schema	Database Owner
user datatype	any	schema	Owner
database options	any	space	System Administrator or Database Owner
group membership	any	security	System Administrator or Database Owner

Table D-2: Roles required to change object properties

Activity	TME	ESSM	SQL Server
login password	any	security	System Administrator or System Security Officer

Role Requirements for Deleting SQL Server and Database Objects

Table D-3: Roles required to delete objects

Object	TME	ESSM	SQL Server
named cache	any	cache	Database Owner
database or database device	any	space	System Administrator or Database Owner
default	any	schema	Database Owner
dump device	any	space	System Administrator
group	any	security	System Administrator or Database Owner
index	any	schema	Owner
SQL Server login	any	security	System Administrator
procedure	any	schema	Owner
remote login	any	security	System Administrator
remote server	any	security	System Security Officer
rule	any	schema	Owner
segment	any	space	System Administrator
table	any	schema	Owner
threshold	any	space	System Administrator
trigger	any	schema	Owner
user	any	security	System Administrator or Database Owner
user datatype	any	schema	Owner
view	any	schema	Owner
user	any	security	System Administrator or Database Owner

Role Requirements for Displaying SQL Server and Database Object Properties

Table D-4: Roles required to display object properties

Object	TME	ESSM	SQL Server
named cache	any	cache	any
database properties and options	any	any	System Administrator or valid database user role
database device	any	any	any
default	any	schema	any
dump device	any	any	none
group	any	any	System Security Officer or valid database user
index	any	schema	any
login	any	any	System Administrator
procedure	any	any	any
remote server	any	any	none
rule	any	any	any
segment	any	any	System Security Officer, or valid user in database. Displaying the threshold option requires System Administrator.
SQL Server processes	any	server	none
SQL Server	any	any	none
SQL Server status	any	any	none
table	any	any	any
trigger	any	any	any
user datatype	any	any	any
user	any	any	System Security Officer, System Administrator, or any valid database user
view	any	any	any

Role Requirements for Managing Enterprise SQL Server Manager Resources

Table D-5: Roles required to manage ESSM resources

Activity	TME	ESSM	SQL Server
Add subscribers to a profile manager	admin, senior or super	none	none
Add a SQL Server login	super	none	none
Create a SQL Server Profile Manager or Database Profile Manager	senior	any	none
Define default or validation policy	senior or super	none	none
Delete a profile	admin, senior, or super	none	none
Delete a SQL Server login	super	none	none
Delete a SQL Server Profile Manager or Database Profile Manager	admin, senior, or super	none	none
Distribute a profile:	admin, senior, or super	see below	see below
SQL Server Profile		server	System Administrator
Database, Database Device, Dump Device, Segment Profiles		space	System Administrator
Login, Remote Login Profiles		security	System Administrator
Remote Server, Group, User Profiles		security	System Security Officer
Cache, Default, Rule, Table, Trigger, User - datatype, Index, View, Procedure Profiles		schema	System Administrator
Enable copying of table data	any	schema	System Administrator
Enable validation policy	senior or super	none	none
Manage SQL Server	server	none	none

Table D-5: Roles required to manage ESSM resources

Activity	TME	ESSM	SQL Server
Populate or unpopulate a profile	senior or super	server ; for database objects, you need schema	System Security Officer
Remove subscribers from a profile manager	senior or super	none	none
Synchronize a profile	senior or super	none	System Security Officer
Configure Event Monitoring Services	admin	none ^a	none
Validate a profile	senior or super	none	System Security Officer

a. The user “root” must have **server** and **schema** roles.

Role Requirements for Miscellaneous Database Management Tasks

Table D-6: Roles required for miscellaneous database management tasks

Activity	TME	ESSM	SQL Server
Add a remote login	any	security	System Administrator
Back up a database	admin , senior , or super	dump	Operator
Change a volume	any	dump and load	none
Check database consistency	any	space	System Administrator or Database Owner
Generate DDL	any	dump	System Administrator
Execute DDL	any	load	System Administrator
Restore a database	any	load	System Administrator

Role Requirements for Miscellaneous SQL Server Management Tasks

Table D-7: Roles required for miscellaneous database management tasks

Activity	TME	ESSM	SQL Server
Change SQL Server configuration	any	server	System Administrator
Display SQL Server status	any	any	none
Kill SQL Server processes	any	server	System Administrator
Start SQL Server or Backup Server	any	server	System Administrator
Start SQL Server Monitor	any	monitor	System Administrator
Stop SQL Server or Backup Server	any	server	System Administrator

Glossary

access

In the audit system, the use of the `select`, `insert`, `update`, or `delete` command on a table or view.

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).

association

The relationship between a SQL Server Profile Manager and SQL Server or a Database Profile Manager and a database. The profiles in a profile manager are copies of the associated SQL Server objects. Changes made to objects in a profile manager are also made in the associated SQL Server or database.

auditing

ESSM provides the ability to configure recording of security-related Server and database activity. The information is recorded in a traceable audit trail.

audit trail (system table `sysaudits`)

The `sysaudits` table contains one row for each audit record in the table.

authentication

ESSM includes commands and a graphical user interface for configuring components that support controlled and secure access to SQL Server.

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.

batch

One or more Transact-SQL statements submitted as a group to SQL Server for processing and terminated by an end-of-batch signal.

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.

bulk copy

The utility for copying data in and out of databases, called `bcp`.

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.

character set

A set of specific characters with an encoding scheme that uniquely defines each character. ASCII is a common character set.

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.

clear

To deselect an option on a dialog box.

CLI (command line interface)

ESSM offers a complete set of commands for managing SQL Servers and databases in the enterprise. These commands are compatible with Tivoli commands and can be entered from the UNIX command line or from scripts. Scripts can be defined as tasks in the Tivoli task library and scheduled as jobs in the Tivoli scheduler.

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.

collection

A grouping of objects that reside in the same SQL Server or database. For example, a managed server is a collection of the databases, logins, database devices, dump devices, and remote servers that reside on the same SQL Server. There can even be collections within collections (for example, SQL Server collections contain several database collections, one for each database in SQL Server).

command

A statement that instructs the computer to perform an operation. The command begins with a word, often a verb, that names the operation. In addition, the command can include one or more keywords with or without variable values that tailor the command.

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.

desktop

The Tivoli Management Environment window contains menus and icons that let you visualize and control the various elements of the distributed environment.

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.

distributed computing environment

An environment comprising a variety of platforms and applications connected by one or more networks.

distribution

The process of copying SQL Server or database information to other SQL Servers or profile managers in the distributed computing environment.

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.

dump

A backup copy of a database or a transaction log, or the process of creating a backup.

dump device

A single tape, partition, or file used for a database or transaction dump. A dump can span many devices, or many dumps can be made to a single tape volume.

dump file

The name of a dump file used to identify a specific backup on the backup media. The name cannot exceed 17 characters and must conform to operating system conventions. If you do not enter a name, Backup Server generates a default name based on:

- The last 7 characters of the database name
- The two-digit year
- The three-digit day of the year (1 through 366)
- Hexadecimal-encoded time at which the dump file was created

For example, the file *catons93059E100* contains a copy of the Publications database created on the fifty-ninth day of 1993.

dynamic configuration parameter

A SQL Server configuration parameter that is updated immediately when you reset it; a restart of SQL Server is not needed.

enterprise

An environment comprising a variety of platforms and applications connected by one or more networks.

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.

ESSM (Enterprise SQL Server Manager)

An application that provides centralized control of SQL Servers in a distributed computing environment. ESSM runs within the Tivoli Management Environment (TME).

ESSM administrator

A system administrator managing SQL Servers within the Tivoli Management Environment (TME). The administrator must have appropriate SQL Server roles and ESSM/TME roles. See also **roles**.

ESSM roles

The extended set of TME roles required by an ESSM administrator to perform SQL Server management tasks. The set of ESSM roles consists of:

- **dump**
- **load**
- **server**
- **security**
- **schema**
- **space**

ESSM roles are attributes of every ESSM administrator and may be assigned using the TME Administrators window. See also **roles**.

Event Monitoring (EMON) Services

A feature of ESSM that enables you to monitor a SQL Server for a high-level event and to manage a response to the event. An event occurs when a high-level SQL Server characteristic surpasses a user-defined threshold. When you use HLE Services to monitor for events, you specify response levels, thresholds, and response actions.

events

The user interface lets you monitor events, such as number of server connections, password expiration dates, I/O activity on a device or the network, and so on. In addition, you can specify levels at which you are notified of an event and a variety of actions that are triggered by an event.

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.

filter

An operation that allows you to specify which objects to hide or display in a window or dialog.

for load

Specifies that a database will be created for restoration from tape.

free-space threshold

A user-specified threshold that specifies the amount of space on a segment and the action to be taken when the amount of space available on that segment is less than the specified space.

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.

GUI (graphical user interface)

ESSM provides a windows environment with menus, icons, and dialog boxes for managing SQL Servers and databases in a distributed computing environment. These graphic elements are compatible with and exist within the Tivoli Desktop.

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*.

managed database

A database that has been registered as a managed resource object in the TMR database. A managed database is a collection of the users, groups, and segments that reside in the same database. A managed database has the resource name SQLDatabase in a policy region managed resource list.

managed resource

An object, such as a device, administrator, or SQL Server, that has a default policy defined in a policy region. For example, a SQL Server can be defined as a managed resource in a TME policy region. Each resource is one of several types (SQL Server is a ManagedSQLServer resource type). Before a resource can be managed in a policy region, the corresponding resource type must appear in the list of valid managed resource types for that policy region. A managed resource can belong to only one policy region at a time.

managed server

A SQL Server that has been registered as a managed resource object in the TMR database. A managed server is a collection of the databases, logins, database devices, dump devices, and remote servers that reside in the same SQL Server. A managed server has the resource name ManagedSQLServer in a policy region managed resource list.

management host

Name of the host machine (TME client) on which ESSM management activities for the specified SQL Server are to occur. ESSM must be installed and running on the client if the machine is to be an ESSM management host.

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.

mirror device

A duplicate SQL Server database device. All writes to the primary device are copied (mirrored) to a second physical device. Writes can be either serial (consecutive) or parallel (simultaneous). If one device fails, the other contains an up-to-date copy of all transactions.

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.

monitoring source

A specific high-level SQL Server characteristic that you use HLE Services to monitor.

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.

notices

A message concerning some operation or change in the distributed system. Messages can be notices on the TME Bulletin Board, e-mail messages, pop-up dialog boxes, and so on.

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**.

parameter

1. A variable value used in conjunction with a command or a stored procedure.
2. A keyword and value that define a SQL statement.

password encryption

The process of storing a password in nondecipherable encrypted form.

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**.

pop-up menu

A menu opened by clicking the second mouse button over an icon. Also called a shortcut menu.

policy region

A collection of TME resources that are governed by a common set of policies. ESSM administrators are given the authority to manage resources in one or more policy regions. A policy region contains a list of resource types that are valid for that policy region. You can add or remove resource types from the list so that you can control the kinds of resources the policy region will govern.

policy

Rule that governs the management of resources, such as requiring login accounts to have passwords. ESSM policy methods take the form of shell scripts. Default policy methods govern default characteristics of resources. Validation policy methods protect the integrity of resources.

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.

profile

A collection of SQL Server or database objects or information that you can distribute (copy) to a set of subscribers (SQL Servers or other profile managers) through a SQL Server Profile Manager or Database Profile Manager. ESSM profiles contain objects of the same type, and the objects are actual, existing objects and information in a SQL Server or database. For example, a user profile contains a set of actual database users in a database that you wish to distribute as a set to the corresponding databases in one or more subscribing SQL Servers (or Database Profile Managers). Profiles can be distributed across multiple platforms.

profile endpoint

A SQL Server or database that subscribes to a profile. The endpoint is the final destination of data distributed from a profile manager. See also **subscriber**.

profile manager

A managed resource that contains set of profiles and a set of subscribers that receive the profiles when they are distributed. ESSM provides two types of profile managers: SQL Server Profile Managers and Database Profile Managers. Each SQL Server Profile Manager you create is associated with an existing SQL Server that contains the actual SQL Server objects and information you wish to distribute. Likewise, each Database Profile Manager you create is associated with an existing database that contains the actual database objects and information you wish to distribute.

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.

resource

A system, device, service, or facility in a distributed system. For example, file systems, workstations, administrators, and SQL Servers can be resources in the Tivoli Management Environment. See also **managed resource**.

response action

A Tivoli Monitoring Technology term that identifies the actions to perform when a monitor exceeds a threshold. See the *Tivoli/Sentry User's Guide* for a complete list of response actions.

response level

A Tivoli Monitoring Technology term that identifies the category of a response. There are five default response levels: severe, critical, warning, normal, and always. You define a threshold and set of response actions for each response level.

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.

roles

Attributes an administrator or SQL Server login possesses that authorize execution of specific administration tasks. ESSM administrators must be aware of three kinds of roles: TME roles, ESSM roles, and SQL Server roles. SQL Server roles are attributes of every SQL Server login. ESSM and TME roles are attributes of every ESSM administrator (which must also have one or more associated SQL Server logins). For any SQL Server administration operation, an ESSM administrator must have the appropriate combination of ESSM, TME, and SQL Server roles.

rule

The domain of acceptable values for a table column or a user datatype.

runserver file

The file used as a reference in restarting SQL Server or Backup Server. By default, the runserver file is named *RUN_servername* and is created when you install SQL Server. Runserver files are created in the *\$\$SYBASE/install* directory.

sa

See **System Administrator**.

scale

The number of digits to the right of the decimal point in a *numeric* or *decimal* datatype.

scheduler

The TME service that enables ESSM administrators with the TME **admin** role to schedule and run tasks and jobs. The scheduler is represented by a clock icon on the TME Desktop.

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.

single-user mode

Starting SQL Server in single-user mode allows only one System Administrator to log in, and turns on the allow updates configuration variable. Use this mode to restore the master database. This option creates a *m_RUN_servername* file and overwrites any existing *m_RUN_servername* file.

source

1. The original SQL Server or database that is being copied or compared with another SQL Server or database.
2. The SQL Server or database associated with a profile manager.
3. The profile manager that is being distributed to another profile manager or SQL Server.

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 login

The name a user uses to log on to SQL Server. A login is valid if SQL Server has an entry for that user in the system table *syslogins*.

SQL Server Manager

A forms-based interface for administration of SQL Servers.

sso

The SQL Server system security officer role, *sso_role*, in charge of security-sensitive tasks in a SQL Server, such as creating, dropping, and locking user accounts and changing user passwords.

statement

A command that instructs the computer to perform an operation. The command begins with a word, often a verb, that names the operation. In addition, the command can include one or more keywords with or without variable values that tailor the command.

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**.

subscriber

A SQL Server or profile manager that you designate to receive the profiles distributed in a profile manager. Profile managers can have one or more subscribers, and the subscribers must match the profile manager type (SQL Server Profile Manager or Database Profile Manager). For example, a Database Profile Manager may have only managed SQL Servers and other Database Profile Managers as subscribers (SQL Server Profile Managers cannot subscribe to a Database Profile Manager).

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.

target

1. The SQL Server or database with which the source SQL Server or database is being compared.
2. The profile manager or SQL Server that is receiving distributed information from a source profile manager.

task library

The Tivoli Management Environment lets you create a task library in which you can create and store tasks and jobs. These tasks and jobs can be run immediately or scheduled to run at a specific time. A task is a TME resource that encapsulates daily operations, such as clearing the printer queue. Jobs are created from tasks. A job lets you specify details of task execution, such as where to display output.

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.

Tivoli Monitoring Technology

Tivoli Monitoring Technology (TMT) allows you to periodically monitor your system for events and to manage a response when an event occurs. HLE Services is an extension of TMT that enables you to monitor SQL Servers for events and to respond appropriately when an event occurs.

Tivoli Name Registry

Contains the name and object ID of objects in the TMR database, including the Sybase login name (same as the UNIX login name) and encrypted password information for each ESSM administrator.

TME (Tivoli Management Environment)

A software environment providing centralized control of integrated software products for a distributed system.

TME Desktop

The window containing menu bars and icons that lets you visualize and control the various elements of the distributed environment.

TME roles

The set of authorization roles assigned to a Tivoli administrator to establish which administration tasks that administrator may perform.

- admin
- backup
- install_client
- install_product
- senior
- super
- restore
- user

TME roles are attributes of every Tivoli administrator and may be assigned using the TME Administrators window. Every administrator must have at least user role to open the TME desktop. See also **roles**.

TMP (Tivoli Management Platform)

The foundation for managing resources in a distributed environment. The TMP is the runtime platform for TME applications and provides the administrator with a view into the network and a set of tools that are applicable across functions and applications in the TME.

TMR (Tivoli Management Region)

A TMR server and the set of clients it serves. A single TMR can contain a maximum of 200 clients. If your network contains more than 200 clients, you can create several TMRs and connect them.

TMR database

The distributed, persistent database that stores all management data and resource descriptions for a Tivoli Management Region (TMR).

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.

transaction log options

You have the following options for the transaction log:

- **Backup, truncate, and log**—back up the transaction log, remove the inactive part 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 entries.
- **Truncate and log**—remove the inactive part of the log without backing it up and create a new transaction log entry recording the dump.

- **Truncate only**—remove the inactive part of the log without backing it up and without creating a new transaction log entry to record the dump.

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

A graphical, hierarchical representation of the object relationships in the SQL Server installations specified in a user's *sql.ini* file. Voyager provides access to all objects and activities managed by Enterprise 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.

wildcard

A special character used to represent one or more characters in a pattern-matching string. Any character or set of characters can replace a wildcard character. The `-wildcard` option is available for some commands to allow using wildcards for specifying names.

with cascade option

An option, when revoking permission, that revokes permission from a specified user or group and also from anyone to whom the specified user or group has granted permission.

with grant option

An option, when granting permission, that allows a specified user or group to grant the same permission to other users or groups.

Index

Symbols

@@thresh_hysteresis system
variable 7-32

A

A, in columns list 10-17

Abort Transaction If Log Full database
option 7-21

Access 8-1
aliases 8-1
command permissions 8-1
dialog boxes 8-48
granting 8-48
object permissions 8-1
revoking 8-48
roles 8-2, 8-48
shared logins 8-1
tabs 8-48

Active transaction 4-21

Activities
drag-and-drop B-4

activities
drag and drop B-4

Add/Remove 11-18

Add button 7-4, 9-33, 10-8, 10-16

Add Identity Column in Non-Unique
Indexes database option 7-21

Adding
columns 9-60
disk mirroring 6-14
monitors to a profile 14-23
objects to a profile 12-5
parameters 9-33
subscribers to an ESSM profile
manager 12-14

additional netmem configuration
parameter A-1

additional network memory
configuration parameter A-4

Add Monitor to Tivoli/Sentry Profile
dialog box 14-24

address lock spinlock ratio configuration
parameter A-4

Administration

notices 1-10

Administration tasks 1-2

Administrator icon pop-up menu 2-3,
2-7

Administrators

collection icon 2-3
and Event Monitoring Services 2-6
roles. *See* TME roles

Administrators resource 2-3

Advanced configuration parameter
display level 4-17

Aliases

access 8-1
navigating to logins 8-24
sharing logins 8-2, 8-15

Allocating

databases on devices 7-4
index pages 9-28
space for logs 7-2
space on new device 7-9, 7-10
storage space 7-2, 7-9
table pages 10-43

Allocation

errors 10-43
maps 10-43
pages 10-43

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 4-14
 - allow updates configuration
 - parameter A-1
 - allow updates to system tables configuration
 - parameter A-4
 - Apply button 2-8, 9-3
 - Assigning
 - administrator roles 2-2 to 2-5
 - columns 9-22
 - indexes 9-22
 - roles 8-10, 8-13
 - SQL Server logins 2-6 to 2-8
 - Association
 - to profile manager 11-2
 - Associations
 - to database profile managers 11-11
 - Attributes
 - subject to policy 13-4
 - audit queue size 4-14
 - audit queue size configuration
 - parameter A-1, A-4
 - Authorization
 - roles. *See* TME roles
 - Auto Identity database option 7-21
 - Automating tasks 1-6
 - Available Resources list box 2-12
 - Available Roles list box 2-4
- B**
- Backing up
 - configuration parameters A-11
 - dump devices 7-38
 - guidelines 7-42
 - identifying Backup Server 7-37
 - master database 7-6, 7-36
 - planning 7-36
 - scheduling 7-42
 - transaction logs 7-20
 - using dbcc 7-55
 - Backup
 - Changing volumes 7-47, 7-53
 - changing volumes 7-47
 - notices 1-10
 - Backup, Truncate, and Log database
 - backup option 7-45
 - Backup/Recovery configuration
 - parameters A-11
 - Backup and Log database backup
 - option 7-45
 - Backup Server 7-37
 - starting 4-2
 - stopping 4-4
 - Bad status processes
 - monitoring 14-15
 - Bars
 - Contents 9-6
 - displaying 3-16
 - help 3-19
 - hiding 3-16
 - Batch processing 1-6
 - bcp utility 7-20
 - Binding
 - defaults 9-16, 9-17
 - deleting. *See* Unbinding.
 - future use 9-17, 9-46
 - objects to a named cache 5-10
 - rules 9-45, 9-46
 - Blocked Processes
 - monitoring 14-15
 - Blocked time, process 4-21
 - Blocking spid 4-21
 - Buffer pools
 - creating 5-7
 - deleting 5-10
 - Buffer pool wash area 5-9
 - Buffer prefetch 9-24, 10-6, 10-20
 - Building indexes 9-24
 - Bulletin board
 - notices 14-31
 - Buttons
 - Add 7-4, 9-33, 10-8, 10-16
 - Apply 9-3
 - Apply button 2-8
 - Change 7-4, 9-33, 10-8, 10-16
 - Change & Close button 2-5
 - Consistency... 9-28
 - Create 9-3

- Done 9-4
 - Done button 2-8
 - Get Defaults 12-11, 13-6
 - OK button 2-17
 - Refresh 3-27
 - Remove 7-5, 9-33, 10-8, 10-16
 - Set SQL Server Logins button 2-8
 - tab 3-20, 9-7
 - toolbar 3-16
- C**
- Cache
 - See also Named caches
 - and large I/Os 9-24, 10-6, 10-20
 - data 5-1
 - management of 5-1
 - procedure 5-1
 - replacement strategy, index 9-24
 - replacement strategy, table 10-6, 10-20
 - Cache Bindings tab 5-12
 - Cache management configuration
 - parameters A-12
 - Cache Properties tab 5-5
 - cache role 2-2
 - Cache strategy
 - setting 10-20
 - Case sensitivity
 - search strings 3-8
 - Change & Close button 2-5
 - Change button
 - allocating space on a device 7-4
 - defining columns 10-8, 10-16
 - defining parameters 9-33
 - Change Password dialog box 8-14
 - Changing
 - database defaults 7-18
 - default devices 6-19
 - profile manager name 11-10
 - properties, role requirements D-3
 - volumes 7-47
 - Changing log buffer size 7-12
 - Changing name
 - profile manager 11-8
 - characteristics 14-2
 - Characters
 - masking 3-8
 - wildcard 3-8
 - Check constraints 10-2, 10-19
 - column 9-40, 10-3, 10-9
 - creating 10-30
 - format of 10-11, 10-31
 - table 9-40
 - ways of creating 10-3
 - Checking
 - database consistency 7-55
 - index consistency 9-28
 - SQL Server network I/O 4-38
 - Checkpoints 7-22
 - issuing automatically 7-22
 - transaction logs 7-22
 - Choosing management hosts 2-13
 - Class on system administration xxxii
 - CLI 1-6
 - Clients
 - as management hosts 2-13
 - Closing dialog boxes 9-4
 - Clustered indexes 9-19, 9-22
 - Codes
 - error, monitoring 14-17
 - Column check constraint 9-40, 10-3
 - Column constraints 10-9, 10-17
 - Columns
 - adding 10-8, 10-15
 - assigning indexes 9-22
 - binding defaults 9-16
 - binding rules 9-45
 - default values 9-10
 - displaying 9-14
 - identity 9-56
 - removing 10-8, 10-16
 - unbinding 9-18, 9-47
 - views 9-60
 - Command
 - waddmon 14-26
 - Command line
 - TMT 14-26

- Command line interface 1-6
- Commands
 - access 8-2
 - create 8-5
 - dialog boxes 3-24
 - Edit Logins menu command 2-7
 - Edit TMR Roles menu command 2-3
 - executing triggers 9-48
 - granting permissions 8-30
 - Managed Resources menu
 - command 2-12
 - ManagedSQLServer menu
 - command 2-11, 2-16
 - odadmin 14-30
 - permissions 8-2, 8-5, 8-43 to 8-47
 - revoking permissions 8-30
 - scripting 1-6
 - SQL 9-4
 - Transact-SQL 4-32, 7-13
 - wlseng 14-30
 - wstopeng 14-30
- Common procedures 9-1
- Complete configuration parameter
 - display level 4-17
- Configuration 11-20
- Configuration... command 4-15, 4-18
- configuration file configuration
 - parameter A-4
- Configuration parameters
 - backup/recovery A-11
 - cache manager A-12
 - disk I/O A-12
 - functional groups of A-11 to A-15
 - general information A-12
 - languages A-12
 - lock manager A-12
 - processors A-13
 - resetting 4-15, 4-18
 - SQL Server 10.x A-1 to A-4
 - SQL Server 11.0 A-4 to A-10
 - SQL Server administration A-14
 - user environment A-15
- Configuration Parameters dialog box
 - SQL Server 10.x 4-15
 - SQL Server 11.0 4-18
- Configuring
 - SQL Server 11.0 4-17
 - SQL Servers 4-14
- Configuring policy regions 2-11
 - See also* Policy regions
- Confirmation dialog box 9-4
- Confirmation dialog boxes 9-8
- Confirm Password box 2-8
- Connection attempts
 - monitoring 14-8
- Connections used
 - monitoring 14-8, 14-10
- Consistency...
 - button 9-28
- Consistency... command 4-38
- Consistency checking
 - database 7-55
 - dbcc utility 7-55
 - index 9-28
- Constraints
 - A as indicator of in table
 - properties 10-17
 - and foreign keys 10-37, 10-38
 - and primary keys 10-37
 - check 10-2, 10-11, 10-19, 10-30, 10-31
 - column 10-9, 10-17
 - creating 10-1
 - default 10-2
 - description of 10-2 to 10-4
 - primary key 9-22
 - primary keys in 10-39
 - referential 10-2, 10-3, 10-35
 - rules 10-2
 - summary of 10-4
- Contacting Technical Support xxxvii
- Container icons 3-3, 11-19
- Contents bar 9-6
- Context-sensitive menus
 - description of 3-13
 - pop-up 3-14
 - pull-down 3-13
 - shortcut 3-14
- Context-specific menus

- in profile managers 11-19
- Copies
 - removing of profiles 14-26
- Copying
 - objects 9-8
 - table data 10-45, 12-25
- Copying device contents 6-12
- Corrupted databases *See Damaged Databases*
- CPU, percent busy
 - monitoring 14-10
- cpu accounting flush interval configuration
 - parameter A-4
- cpu flush configuration option A-1
- cpu grace time configuration
 - parameter A-4
- CPU time 4-21
- CPU usage
 - configuration parameters affecting A-4
- Create
 - button 9-3
 - menu command 9-2
- Create Database Profile Manager dialog box 11-10
- Create dialog boxes
 - Cache 5-2
 - Database 7-3
 - Database Device 6-4
 - Default 9-11
 - Dump Device 7-39
 - Group 8-32
 - Index 9-21
 - Login 8-9
 - Procedure 9-32
 - Remote Server 4-24
 - Rule 9-41
 - Segment 7-24
 - Trigger 9-49
 - User 8-19
 - User Datatype 9-55
 - View 9-61
- Create menu 2-11, 2-16
- Create object 11-20
- Create Profile dialog box
 - Dialog boxes
 - Create Profile 12-2
- Create Profiles 11-18
- Create SQL Server Profile Manager dialog box 11-8
- Create toolbar button 3-17
- Creating
 - buffer pools 5-7
 - check constraints 9-33, 10-11, 10-19, 10-30
 - column constraints 10-9, 10-17
 - constraints 10-1
 - database devices 6-4
 - Database profile managers 11-10
 - databases 7-1, 7-3
 - defaults 9-10
 - devices 6-2
 - dump devices 7-39
 - groups 8-31
 - guest users 7-4
 - indexes 9-20
 - logins 8-9
 - named cache binding 5-12
 - named cache buffer pools 5-7
 - named caches 5-2
 - objects 9-2
 - objects, roles D-2
 - primary key index 9-22
 - procedures 9-32
 - profile manager
 - Tivoli 14-20
 - referential constraints 10-35
 - rules 9-33, 9-40, 9-41
 - schemas 9-4
 - segments 7-24
 - segment thresholds 7-32
 - shortcuts 9-3
 - SQL Server profile managers 11-8
 - table columns 10-8, 10-15
 - tables 10-1, 10-5
 - toolbars and 1-3
 - triggers 9-49
 - user datatypes 9-55

- users 8-19
- views 9-61
- Creating check constraints 9-40
- Creating ESSM profiles 12-2
- Creating managed SQL Servers
 - roles required for 2-16
- Creating profile managers 11-7
- Creating remote servers 4-24
- Cumulative CPU time 4-21
- Current Login Names list box 2-8
- Current Resources list box 2-12
- Current Roles list box 2-5
- Cycling SQL Server 4-2, 4-4

D

- d_master* device 6-3
- Damaged databases, deleting 7-7
- Data
 - copying 10-45
 - copying table 12-25
 - retrieving 9-19
 - sorting 9-24
- Data and log cache 5-6
- Database
 - managing
 - roles required D-8, D-9
- Database Allocations dialog box 7-11
- Database Backup dialog box 7-44
- Database backup options 7-45
- Database Consistency Check dialog box 7-56
- Database Definition Language (DDL).
See DDL.
- Database Device Databases tab 6-8
- Database Device menu 6-7
- Database Device Properties tab 6-7
- Database Device Segments tab 6-10
- Database Generate DDL dialog box 7-14
- Database objects
 - copying 9-8
- Database Options dialog box 7-18
- Database Profile Manager 11-6
 - associations to 11-11

- opening 11-13
- profiles in C-4
- Database Profile Manager. *See also*
Profile manager xxxvii
- Database Profile Managers
 - overview 1-8
- Database profile managers 11-4
 - creating 11-10
- Database Properties dialog box 7-8
- Database Run DDL dialog box 7-15
- Databases 7-1 to 7-7
 - allocating on devices 7-4
 - backing up 7-6
 - backup guidelines 7-42
 - binding to a named cache 5-10
 - changing log buffer size 7-12
 - Changing volumes during a restore or
backup 7-47, 7-53
 - checking consistency 7-55
 - checkpoints 7-22
 - creating 7-1, 7-3
 - default options 7-18
 - deleting damaged 7-7
 - deleting normal 7-6
 - devices 6-2, 6-8
 - displaying 8-15
 - displaying objects 9-6
 - displaying properties 7-7
 - estimating size 7-2
 - executing DDL 7-15
 - guest users 7-4
 - increasing storage 7-9
 - master* 6-2
 - model* 6-3
 - navigating to 6-9
 - permissions 8-4
 - recovering 7-49
 - reducing size 7-9
 - segments 6-10, 7-23
 - setting online 7-60
 - storing 6-3, 6-8
 - sybssystemprocs* 6-3
 - tempdb* 6-3
 - transferring ownership 7-12

- database size configuration
 - parameter A-1
- Data cache 5-1
- Data Definition Language in Transaction
 - database option 7-21
- Data logs 6-20
- dbcc
 - index consistency checking 9-28
 - network I/O checking 4-38
 - options 7-57
 - table consistency checking 10-40
 - utility 7-55, 10-40
- DDL
 - editing 4-35 to 4-37, 7-16 to 7-17
 - executing scripts 4-32, 7-15
 - files containing 4-35 to 4-37, 7-16 to 7-17
 - generating scripts 4-32
 - Transact-SQL commands 7-13
- ddl directory 4-35, 7-16
- deadlock checking period configuration
 - parameter A-4
- Default Bindings tab 9-16
- default character set id configuration
 - parameter A-5
- default database size configuration
 - parameter A-5
- Default Dependencies tab 9-14
- 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 Policy 11-18
- Default policy 1-8, 13-1
 - See also* Policy 1-8
 - enabling and disabling 13-7
 - generating values 12-11, 13-6
 - overview 1-8
 - See also* Policy specifying 13-3
- Default policy. *See also* Policy, Validation policy xxxvii
- Default Policy dialog box 13-3
- Defaults
 - as constraints 10-2
 - binding 9-16
 - Bindings tab 9-16
 - changing device 6-19
 - creating 9-11
 - databases 8-10, 8-12
 - dependencies 9-14
 - Dependencies tab 9-14
 - displaying properties 9-13
 - languages 8-10, 8-12
 - multiple devices 6-19
 - navigating to dependencies 9-14
 - options 7-18
 - permissions 8-43 to 8-47
 - storage devices 6-3, 6-5
 - unbinding 9-18
 - ways of creating 9-10, 10-2
- default segment 7-23
- default sortorder id configuration
 - parameter A-5
- Default values
 - generating 12-10, 13-6
- Defining
 - default policies 13-3
- Defining validation policy 13-8
- Delete... menu command
 - database 7-6
 - deleting objects 9-8
- delete permission 8-27, 8-39, 9-70
- Delete toolbar button 3-17
- Deleting
 - bindings. *See* Unbinding
 - buffer pools 5-10
 - database devices 6-6
 - databases 7-6
 - databases, damaged 7-7
 - dump devices 7-40
 - ESSM profiles 12-4

- groups 8-33
- logins 8-11
- named cache binding 5-14
- Named cache buffer pools 5-10
- named caches 5-4
- objects 9-8
- objects, roles required D-5
- profile managers
 - Profile managers
 - deleting 11-12
- segments 7-26
- segment thresholds 7-34
- tables 10-12
- users 8-21
- Dependencies
 - displaying default 9-14
 - displaying procedure 9-35
 - displaying rule 9-43
 - displaying table 10-21
 - displaying trigger 9-53
 - displaying user datatype 9-58
 - displaying view 9-66
 - distribution C-1
 - for distribution 11-6
- Desktop scheduler 1-6
- Details display format 3-6
- Details toolbar button 3-17
- Device reads
 - monitoring 14-8
- Devices
 - allocating databases 7-4
 - changing default 6-19
 - copying contents 6-12
 - creating 6-2, 6-4
 - d_master* 6-3
 - databases 6-2, 6-8
 - default storage 6-3, 6-5
 - deleting 6-6
 - disks 6-3
 - displaying properties 6-7
 - dump 6-2, 7-38
 - extending segments 7-29
 - master 6-19
 - mirroring 6-5
 - multiple default 6-19
 - navigating to 7-29
 - segments 6-10
 - segments pointing to 7-23
 - separating usage 6-20
 - storage space on 6-8
 - storing transaction logs 7-4
- devices.ddl* file 4-35
- devices configuration parameter A-1
- Device writes
 - monitoring 14-8
- Dialog box
 - Default Policy 13-3
- Dialog Boxes
 - Create Database Profile
 - Manager 11-10
 - Filter by Name 3-8
- Dialog boxes
 - access 8-48
 - Add Monitor to Tivoli/Sentry
 - Profile 14-24
 - closing 9-4
 - commands 3-24
 - confirmation 9-8
 - Create SQL Server Profile
 - Manager 11-8
 - Distribute Profiles 12-22
 - Edit Sentry Monitor 14-25
 - granting permissions 8-48
 - Manage SQL Server dialog box 2-16
 - messages 3-25
 - navigating to 3-23
 - object 3-20
 - Populate/Unpopulate 12-6
 - refreshing 3-27
 - Set Login Names dialog box 2-7
 - Set Managed Resources dialog
 - box 2-12
 - Set TMR Roles dialog box 2-4
 - SQL Server Logins for Administrator
 - dialog box 2-8
 - subscribers 12-14
 - tabs 3-20
 - Tivoli/Sentry Profile 14-23

- Volume Change 7-47
- Direct updates to system tables A-4
- Disabling
 - default policy 13-7
- Disabling disk mirroring 6-16
- Disk I/O configuration
 - parameters A-12
- disk i/o structures configuration
 - parameter A-5
- Disk reads and writes 4-21
- Disks
 - adding mirroring 6-14
 - devices 6-3
 - disabling mirroring 6-16
 - mirroring 6-1, 6-12
 - re-enabling mirroring 6-18
- Display
 - synchronizing windows 3-27
 - updating 3-27
- Display formats, selecting 3-6
- Displaying
 - bars 3-16
 - columns 9-14
 - database device properties 6-7
 - database properties 7-7
 - databases 8-15
 - dependencies 9-14
 - details about an object 9-4
 - dump device properties 7-41
 - groups 8-35
 - icons 3-16
 - indexes 10-28
 - logins 8-11
 - management host 4-12
 - named caches 5-5
 - objects 9-7
 - roles required D-6
 - procedure dependencies 9-35
 - procedures 10-21
 - properties used in schema 9-4
 - rule dependencies 9-43
 - segment properties 7-27
 - segments 9-27
 - SQL Server and database objects 9-6
 - SQL Server configuration
 - parameters 4-17, 4-18
 - SQL Server processes 4-20
 - SQL Server properties 4-12
 - status of SQL Server 4-3
 - table dependencies 10-21
 - tables 9-35
 - trigger dependencies 9-53
 - triggers 9-35, 10-21, 10-28
 - user datatype dependencies 9-58
 - user datatypes 9-14, 9-35
 - user processes 4-20
 - users 8-22
 - view dependencies 9-66
 - views 9-35, 10-21
- Displaying properties
 - SQL Server 4-12
- Display menu 3-12
- Distribute 11-18
- Distribute Profiles dialog box 12-22
- Distributing
 - table data 12-25
- Distributing ESSM Profiles 12-20
- Distributing objects and information 1-8
- Profiles
 - concepts 11-2
- Distributing policy 13-2
- Distributing profiles
 - order of C-1
 - using drag-and-drop 12-23, 12-24
- Distribution
 - behavior and dependencies C-1
 - dependencies 11-6
- Distribution errors 12-27
- Done button 2-8, 9-4
- Drag-and-drop 1-3
 - activities summarized B-4
 - icons 3-9
 - profile distribution 12-23, 12-24
 - tables 10-25
 - users 9-71, 10-25
 - views 9-71
- Dragging. *See* Drag-and-drop.

Dragging and dropping icons 3-9 to 3-10
 Drop Connections with No Activity remote server option 4-25
 Dump
 Backup Server, starting 4-2
 dump database command 7-20
 Dump Device Properties tab 7-41
 Dump devices 6-2
 backing up 7-38
 creating 7-39
 deleting 7-40
 displaying properties 7-41
 Dumping transaction logs 7-20
 dump role 2-2
 Duplicating. *See* Mirroring and Copying.

E

Editing
 DDL scripts 4-32
 Edit Logins menu command 2-7
 Edit Sentry Monitor dialog box 14-25
 Edit TMR Roles menu command 2-3
 Enabling
 default policy 13-7
 Encrypt Password remote server option 4-25
 Endpoints
 distribution of policy to 13-2
 ENDPOINT variable 14-4
 Enforcing referential integrity 9-48, 10-35
 engine adjust configuration
 parameter A-2
 Engine number, process 4-21
 Enterprise Level tab 2-8
 Enterprise SQL Server Manager
 management hosts 2-13 to 2-15
 overview 1-1
 resources, adding to policy region 2-11
 roles 2-2

 starting 3-1
 stopping 3-1
 user interface 1-5 to 1-6
 Enterprise-wide SQL Server logins 2-6
 Environment variables
 in Event Monitoring Services 14-4
 Error Codes
 monitoring 14-17
 Error log file
 specifying 14-19
 Error log file events 14-2
 Error Log File Monitors 14-16
 Errors
 distribution 12-27
 Error types
 monitoring 14-18
 ESSM
 see also Enterprise SQL Server Manager
 overview 1-1
 ESSM administrators
 and Event Monitoring Services 2-6
 roles, assigning 2-2 to 2-5
 roles for 2-2
 SQL Server logins, assigning 2-6 to 2-8
 UNIX logins of 2-6
 Estimating
 database size 7-2
 running `sp_estspace` 7-2
 space requirements 7-2
 event, definition of 14-1
 Event Monitoring Services 14-1 to 14-31
 adding monitoring sources to Sentry profiles 14-20 to 14-27
 before you use 14-4
 characteristics monitored 14-2
 distributing Sentry profiles 14-5
 Edit Sentry Monitor dialog box 14-25
 and environment variables 14-4
 error log file events
 list of 14-2
 and ESSM administrators 2-6
 generic events 14-2

- log file for 14-30
 - monitoring SQL Server logins 2-6
 - Number of Connection Attempts 14-8
 - Number of Connections Used 14-8
 - Number of Device Reads 14-8
 - Number of Device Writes 14-8
 - Number of Faulty Databases 14-9
 - Number of Locks Used 14-9
 - Number of Network Packets
 - Received 14-9
 - Number of Network Packets
 - Sent 14-9
 - overview 14-1 to 14-7
 - Percent of Connections Used 14-10
 - Percent of CPU Busy 14-10
 - Percent of Locks Used 14-10
 - Percent of Segment Space
 - Available 14-11
 - process-specific events
 - list of 14-2
 - scripts, using with 14-4
 - and Sentry notices group 14-4
 - and Sentry profiles 14-4 to 14-27
 - SQL Server Monitoring Sources 14-7
 - SQL Server Status 14-12
 - subscribing to 14-26
 - Sybase Login Status 14-13
 - Tivoli/Sentry Profile dialog box 14-23
 - and Tivoli Monitoring
 - Technology 14-1 to 14-7
 - and TME roles 14-4
 - troubleshooting 14-30
 - UNIX name for Sentry 14-4
 - unsubscribing from 14-26
- Events
- monitoring 14-1 to 14-31
- events 14-7 to 14-13
- Number of Connection Attempts 14-8
 - Number of Connections Used 14-8
 - Number of Device Reads 14-8
 - Number of Device Writes 14-8
 - Number of Faulty Databases 14-9
 - Number of Locks Used 14-9
 - Number of Network Packets
 - Received 14-9
 - Number of Network Packets
 - Sent 14-9
 - Percent of Connections Used 14-10
 - Percent of CPU Busy 14-10
 - Percent of Locks Used 14-10
 - Percent of Segment Space
 - Available 14-11
 - SQL Server Status 14-12
 - Sybase Login Status 14-13
- executable code size configuration
- parameter A-5
- execute permission 9-36, 9-38
- Executing
- DDL scripts 4-32, 7-15
 - procedures 9-36
 - triggers 9-48, 9-50
- Expression, rule 9-42
- Extending segments 7-29
- extent i/o configuration parameter A-2
- F**
- Faulty databases
- monitoring 14-9
- File menu 3-12
- Files
- .ddl* 4-35 to 4-37, 7-16 to 7-17
 - devices.ddl* 4-35
 - remote.ddl* 4-35
- Fill factor 9-23
- fill factor configuration option A-2
- fill factor configuration parameter A-2
- Filter By Name... menu command 3-8
- Filter by Name dialog box 3-8
- Filter by Name toolbar button 3-17
- Filtering and status bar 3-9, 3-19
- Finding
- case sensitivity 3-8
 - search strings 3-8
- Foreign keys 10-37
- Referential constraints
- foreign keys in 10-38

framework 1-4
 freelock transfer block size configuration
 parameter A-5
 Future bindings 9-17, 9-46

G

General information configuration
 parameters A-12
 Generating
 Server DDL 4-32
 Generating default policy values 12-11,
 13-6
 Generating default values 12-10, 13-6
 Generic event monitors
 thresholds 14-7
 Generic events
 list of 14-2
 Get Defaults button 12-11, 13-6
 Getting started 1-1
 Granting
 command permissions 8-30, 8-41
 object permissions 9-38
 procedure permissions 9-36
 table permissions 10-22
 view permissions 9-68
 with grant 8-6
 with grant 9-38
 Graphical user interface 1-5
 Group Command Permissions tab 8-41
 Group Object Permissions tab 8-38
 Group Properties tab 8-34
 Groups
 creating 8-31
 deleting 8-33
 displaying 8-35
 modifying 8-24
 navigating to 9-39, 9-71, 10-25
 object permissions 8-37
 procedure permissions 9-36
 Public 8-20
 removing users 8-34
 sharing permissions 8-24, 8-31
 table permissions 10-22

 view permissions 9-68
 Guest users
 creating 7-4
 databases 7-4
 GUI 1-5
 Guidelines, backing up 7-42

H

Help
 button 3-17
 how to use xxxvi
 menu 3-12
 status bar 3-19
 toolbar buttons 3-18
 Hiding
 bars 3-16
 icons 3-16
 High-Level Event Services
 overview 1-9
 HLE
 See High-Level Event Services
 Host, management
 displaying 4-12
 Host, process 4-21
 Host machine
 See also Clients; Management hosts
 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 5-7, 9-24, 10-6, 10-20
 Icons 1-3
 administrator 2-3
 container 3-3
 displaying 3-16
 dragging and dropping B-4
 filtering by name 3-8

- hiding 3-16
 - object 3-3
 - profile manager 11-4
 - Icons, list of B-1
 - 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 9-28
 - Indexes
 - allocating pages 9-28
 - assigning columns 9-22
 - binding to a named cache 5-10
 - building 9-24
 - cache replacement strategy 9-24
 - checking consistency 9-28
 - clustered 9-19, 9-22
 - Create tab 9-20
 - creating 9-20
 - creating as primary key 9-22
 - displaying 10-28
 - estimating space requirements 7-2
 - fill factor 9-23
 - keys 9-22
 - limiting page size 9-23
 - maximum rows per page 9-21
 - modifying 9-19
 - navigating to 7-31, 10-29
 - nonclustered 9-19
 - nonunique 9-19
 - primary key 10-37
 - segments 7-30, 9-21
 - Segments tab 9-26
 - sorting data 9-24
 - tables 9-21
 - unique 9-19, 9-22
 - values 9-22
 - Index Properties tab 9-25
 - Index Segments tab 9-26
 - Infected Processes
 - monitoring 14-15
 - insert permission 8-27, 8-39, 9-70
 - Integrity
 - checks 9-28
 - primary key constraints 9-22
 - referential 9-48
 - Interfaces file and remote servers 4-23
 - Introduction 1-1
 - Issuing checkpoints 7-22
- ## K
- Key
 - primary as constraints 9-22
 - Keys, index 9-22
 - Killing user processes 4-22
- ## L
- language in cache configuration
 - parameter A-2
 - Languages configuration
 - parameters A-12
 - Large buffer prefetch 9-24, 10-6, 10-20
 - Large I/Os 5-7, 9-24, 10-6, 10-20
 - Large Icon display format 3-6
 - Large Icon toolbar button 3-17
 - Last-chance threshold 7-32
 - Level
 - monitoring severity 14-17
 - Library, Task 1-6
 - Limiting page size 9-23
 - List display format 3-6
 - List toolbar button 3-17
 - load role 2-2
 - Locking logins 8-13, 8-21
 - 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
 - Locks used
 - monitoring 14-9, 14-10
 - Log buffer size, changing 7-12
 - Log file, for Event Monitoring Services 14-30
 - Login Alias/Users tab 8-16
 - Login Databases Owned tab 8-15
 - Login Properties tab 8-12
 - Logins
 - access 8-1
 - aliases 8-2
 - assigning aliases 8-15
 - creating 8-9
 - default database 8-10
 - deleting 8-11
 - displaying 8-11
 - language 8-10
 - locking 8-13, 8-21
 - modifying 8-12
 - monitoring with EMON 2-6
 - remote server 4-26
 - roles 8-10, 8-13
 - sharing 8-15, 8-24
 - UNIX. *See* UNIX logins
 - Log only caches 5-6
 - Logs
 - allocating space 7-2
 - data 6-20
 - maintaining 6-1
 - separating 6-20
 - storing 6-3, 7-4
 - transaction 6-3, 6-20
 - logsegment* segment 7-23
- M**
- Maintaining logs 6-1
 - Managed resources
 - SQLServerProfileMgr and SQLDatabaseProfileMgr 11-7
 - Managed Resources menu
 - command 2-12
 - ManagedSQLServer menu
 - command 2-11, 2-16
 - ManagedSQLServer* resource
 - See also* Managed SQL Servers
 - creating 2-13 to 2-17
 - Managed SQL Servers
 - creating *ManagedSQLServer* resources 2-13 to 2-17
 - host machine of 2-14
 - management hosts 2-13 to 2-15
 - policy regions, configuring for 2-11 to 2-13
 - preparing Tivoli Management Environment for 2-1 to 2-13
 - SQL Server logins for 2-6
 - Tivoli Management Environment preparations for 2-1 to 2-13
 - Management host
 - displaying 4-12
 - Management Host box 2-16
 - Management hosts 2-13 to 2-15
 - choosing 2-13
 - Managers
 - Profile 11-4
 - Manage SQL Server dialog box 2-16
 - Managing
 - objects 9-1
 - resources 6-1
 - roles required D-7
 - subscribers 12-11
 - Managing SQL Server 2-16 to 2-17
 - Managing Subscribers 12-11
 - Mapping remote server logins 4-26
 - Maps, allocation 10-43
 - Masking characters 3-8
 - master* database
 - backing up 7-6, 7-36
 - creating devices 6-2
 - Master device 6-19

- 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 6-20
- Maximum index rows per page 9-21
- maximum network configuration
 - parameter A-2
- Maximum table rows per page 10-6, 10-19
- max network packet size configuration
 - parameter A-6
- max number of network listeners configuration
 - parameter A-7
- max online engines configuration
 - parameter A-2
- max online engines configuration
 - parameter A-6
- Memory, process 4-21
- memory alignment boundary configuration
 - parameter A-7
- memory configuration parameter A-2
- Memory pools 5-7
- Memory use configuration
 - parameters A-13
- Menus
 - administrator icon pop-up menu 2-3, 2-7
 - context-sensitive 3-13
 - context-specific in profile managers 11-19
 - Create menu 2-11, 2-16
 - pop-up 3-14
 - pop-up, administrator icon 2-3, 2-7
 - Properties menu 2-12
 - pull-down 3-13
 - shortcut 3-14
 - standard 3-11
- Menus and toolbars 3-11 to 3-19
- Messages 1-10
 - dialog boxes 3-25
- Minimum Severity Level
 - monitoring 14-17
- min online engines configuration
 - parameter A-2
- Mirroring
 - adding 6-14
 - benefits 6-12
 - costs 6-12
 - devices 6-5
 - disabling 6-16
 - disks 6-1, 6-12
 - recovery benefits 6-20
 - re-enabling 6-18
- model* database 6-3, 9-55
- Modifying
 - default databases 8-12
 - default languages 8-12
 - groups 8-24
 - indexes 9-19
 - logins 8-12
 - objects, roles required D-3
 - procedures 9-31
 - rules 9-40
 - user datatypes 9-54
 - views 9-60
- Monitoring
 - bad status processes 14-15
 - blocked processes 14-15
 - error codes 14-17
 - error log files 14-16
 - error types 14-18
 - events 14-1 to 14-31
 - infected processes 14-15
 - minimum severity level 14-17
 - non-Tivoli platforms 14-5
 - processes 14-15
 - regular expression, with 14-17
 - SQL Server events 14-1 to 14-31
 - stopped processes 14-15
 - user-specified events 14-14
- Monitoring source
 - names 14-27
- monitoring source 14-1
- Monitoring sources 14-7 to 14-13
 - bad status processes 14-15

- error log file 14-16
- Number of Connection Attempts 14-8
- Number of Connections Used 14-8
- Number of Device Reads 14-8
- Number of Device Writes 14-8
- Number of Faulty Databases 14-9
- Number of Locks Used 14-9
- Number of Network Packets Received 14-9
- Number of Network Packets Sent 14-9
- Percent of Connections Used 14-10
- Percent of CPU Busy 14-10
- Percent of Locks Used 14-10
- Percent of Segment Space Available 14-11
- SQL Server Status 14-12
- Sybase Login Status 14-13
- user-specified 14-14
- monitor role 2-2
- Monitors 14-7 to 14-13
 - adding to a profile 14-23
 - Number of Connection Attempts 14-8
 - Number of Connections Used 14-8
 - Number of Device Reads 14-8
 - Number of Device Writes 14-8
 - Number of Faulty Databases 14-9
 - Number of Locks Used 14-9
 - Number of Network Packets Received 14-9
 - Number of Network Packets Sent 14-9
 - Percent of Connections Used 14-10
 - Percent of CPU Busy 14-10
 - Percent of Locks Used 14-10
 - Percent of Segment Space Available 14-11
 - SQL Server Status 14-12
 - Sybase Login Status 14-13
- Most Recently Used replacement strategy 9-24, 10-6, 10-20
- Moving
 - Profile Managers 11-11
- MRU replacement strategy 9-24, 10-6, 10-20
- N**
- Name
 - changing profile manager 11-10
- Name box 2-16
- Named caches
 - binding objects to 5-10
 - changing size of 5-6
 - changing type of 5-6
 - creating 5-2
 - creating a binding 5-12
 - data and log 5-6
 - defining buffer pools 5-7
 - deleting 5-4
 - deleting a binding 5-14
 - displaying 5-5
 - log only 5-6
 - status 5-3, 5-11
 - wash buffers 5-9
- Names, filtering 3-8
- Navigating
 - Voyager window lists 3-4
- Navigating to
 - aliases 8-24
 - databases 6-9
 - default dependencies 9-14
 - devices 7-29
 - dialog boxes 3-23
 - indexes 7-31, 9-27, 10-29
 - objects 8-40
 - procedure dependencies 9-36
 - procedure permissions 9-39
 - rule dependencies 9-44
 - segments 6-11, 9-27
 - table dependencies 10-21
 - tables 7-31
 - trigger dependencies 9-53
 - triggers 10-29
 - user datatype dependencies 9-59
 - users 8-18, 8-36
 - users and groups 9-39, 9-71, 10-25

- view dependencies 9-67
- nested triggers configuration
 - parameter A-2
- Network communications configuration
 - parameters A-13
- Network I/O tasks, checking 4-38
- Network packet size 4-21
- Network packets received
 - monitoring 14-9
- Network packets sent
 - monitoring 14-9
- 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 9-19
- Non-Tivoli platforms
 - monitoring 14-5
- Nonunique indexes 9-19
- Notices
 - administration 1-10
 - rollback 1-10
- Notices bulletin board 14-31
- 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 3-19
- 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-8
- 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-5

O

- Object display, sorting 3-7
- Object icons 3-3
- Objects
 - and Contents bar 9-6
 - copying 9-8
 - creating 9-2
 - roles required D-2
 - database, displaying 9-6
 - deleting 9-8
 - roles required D-5
 - dialog boxes 3-20
 - displaying
 - roles required D-6
 - displaying details 9-4
 - displaying ownership 8-25
 - group permissions 8-37
 - hiding 1-3
 - icons 1-3
 - managing 9-1
 - navigating to 8-40

- owner permissions 8-5
 - owners 9-48
 - permissions 8-6, 8-37
 - properties 9-4
 - relative to other objects 9-7
 - sorting Details display 3-7
 - SQL Server, displaying 9-6
 - user permissions 8-26
 - visual representation 1-3
 - odadmin command 14-30
 - OK button 2-17
 - Oldest active transaction 4-21
 - Online help xxxvi
 - Open 11-18
 - open databases configuration
 - parameter A-2
 - Opening
 - profile manager 11-13
 - SQL Server window 4-10
 - Opening a new window 3-5
 - open objects configuration parameter A-2
 - oper_role 2-7
 - Operating system resources
 - configuration parameters A-13
 - Operator permissions 8-4
 - Options
 - dbcc 7-57
 - default 7-18
 - Options... menu command 7-18
 - Order
 - for distributing profiles C-1
 - Output dialog box 3-26
 - Overview 1-1
 - Event Monitoring Services 14-1 to 14-7
 - Ownership
 - displaying 8-25
 - permissions and 8-4
 - transferring databases 7-12
- P**
- Packets
 - monitoring 14-9
 - Packet size, network 4-21
 - page lock spinlock ratio configuration
 - parameter A-8
 - Pages
 - allocating 9-28, 10-43
 - limiting size 9-23
 - page utilization configuration
 - parameter A-8
 - Parameters
 - adding 9-33
 - removing 9-33
 - partition groups configuration
 - parameter A-8
 - Partitioning a table 10-32
 - partition spinlock ratio configuration
 - parameter A-8
 - Password, encrypting 4-25
 - password expiration configuration
 - parameter A-2
 - password expiration interval 4-14
 - Passwords, SQL Server 2-8
 - Percentage, fill factor 9-23
 - Performance
 - benefits 6-12
 - costs 6-12
 - maximizing 6-20
 - segment benefits 7-23
 - perform disk i/o on engine 0 configuration
 - parameter A-9
 - permission cache entries configuration
 - parameter A-9
 - Permissions
 - access 8-1, 8-48
 - Changing Volumes 7-47, 7-53
 - command 8-5, 8-43 to 8-47
 - creating referential constraints 8-6, 8-27, 8-39
 - damaged databases, deleting 7-7
 - database devices, creating 6-3
 - database devices, deleting 6-6
 - database devices, displaying 6-7
 - database devices, modifying 6-7
 - database options, displaying 7-18
 - database options, modifying 7-18

- database owners 8-4
- database properties, displaying 7-7
- database properties, modifying 7-7
- databases, backing up 7-44
- databases, checking consistency 7-55
- databases, creating 7-3
- databases, creating backup tasks 7-44
- databases, creating restore tasks 7-50
- databases, deleting 7-6
- databases, restoring 7-50
- default 8-43 to 8-47
- deleting a row 8-6, 8-27, 8-39
- deleting a view 9-70
- dialogs 8-48
- dump devices, creating 7-39
- dump devices, deleting 7-40
- execute 9-38
- executing a procedure 8-6
- granting 8-41, 9-38
- groups 8-37
- groups, creating 8-31
- groups, deleting 8-33
- groups, displaying 8-36
- groups, modifying 8-36
- inserting a row 8-6, 8-27, 8-39, 9-70
- logins, creating 8-9
- logins, deleting 8-11
- modifying 9-36, 9-68, 10-22
- object 8-6, 8-26, 8-37
- object owners 8-5
- Operators 8-4
- procedure 9-36
- Public group 8-43 to 8-47
- referential constraint referencing a view 9-70
- remote servers, creating 4-24
- remote servers, deleting 4-26
- remote servers, displaying 4-30
- remote servers, modifying 4-30
- revoking 8-41, 9-38
- roles 8-2
- segments, creating 7-24
- segments, deleting 7-26
- segments, displaying 7-27
- segments, modifying 7-27
- selecting a table or view 8-6, 8-27, 8-39
- selecting a view 9-70
- sharing 8-2, 8-15, 8-24, 8-31
- shortcuts 9-71, 10-25
- SQL Server, starting 4-2
- SQL Server configuration variables, modifying 4-14
- SQL Server processes, displaying 4-20
- SQL Server processes, killing 4-22
- summary 8-43 to 8-47
- System Administrators 8-3
- System Security Officers 8-3
- table 10-22
- tabs 8-48
- transfers and 8-43
- updating 9-68
- updating a row 8-6, 8-27, 8-39, 9-70
- users, creating 8-19
- users, deleting 8-21
- users, displaying 8-23
- users, modifying 8-23
- view 9-68
- with cascade 8-6
- with cascade 9-38
- with grant 8-6
- with grant 9-38
- Physical memory configuration parameters A-13
- Planning
 - backup 7-36
 - recovery 7-36, 7-49
- Platforms
 - non-Tivoli 14-5
- Policy
 - application of 13-2
 - attributes subject to 13-4
 - default 12-11, 13-6
 - default, enabling or disabling 13-7
 - default policy 1-8
 - definition of 13-1
 - distributing 13-2

- overview 1-8
- validation policy 1-8
- Policy. *See also* Validation policy, Default policy xxxvii
- Policy region
 - resources 1-7
- Policy regions
 - adding managed SQL Servers to 2-13
 - configuring for managed SQL Servers 2-11 to 2-13
 - SQL Server logins for 2-6
- Pools, buffer 5-7
- Populate 11-18
- Populate/Unpopulate dialog box 12-6
- Populating ESSM profiles 12-5
- Pop-up menu commands 3-14, 11-19
- Pop-up menus 3-14
- pre-read packets configuration
 - parameter A-2
- Primary key 9-22
 - constraints 9-22
- Primary keys 10-37, 10-39
- 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 9-34
- Procedures
 - adding parameters 9-33
 - Create dialog box 9-32
 - creating 9-32
 - Dependencies tab 9-35
 - displaying 9-35, 10-21
 - displaying dependencies 9-35
 - executing 9-36
 - modifying 9-31
 - navigating to dependencies 9-36
 - permissions 9-36
 - Permissions tab 9-37
 - recompiling 10-40
 - sp_estspace 7-2
- Process details 4-21
- Processes
 - monitoring blocked 14-15
 - monitoring infected 14-15
 - monitoring bad 14-15
 - monitoring stopped 14-15
- Processors configuration
 - parameters A-13
- Process-specific events 14-2
- Process-specific monitors 14-15
- Profile 11-1
 - adding objects to 12-5
 - associations 11-2
 - copies, removing 14-26
 - definition 11-2
 - in database profile managers 11-6
 - list order 11-15
 - target 11-2
- Profile distribution
 - drag-and-drop 12-23, 12-24
- Profile management
 - overview 1-8
- Profile manager
 - changing name 11-8
 - database 11-6
 - icons 11-4
 - left pane of window 11-15
 - name change 11-10
 - opening 11-13
 - opening SQL Server window
 - from 11-21
 - policy, used in 13-1
 - SQL Server 11-5
 - subscribers 11-12
- Tivoli
 - creating 14-20
- window 11-13

- Profile manager. *See also* Database Profile Manager xxxvii
 - Profile manager. *See also* SQL Server Profile Manager xxxvii
 - Profile Managers
 - moving 11-11
 - Profile managers 11-4
 - adding subscribers 12-14
 - as subscribers 12-12
 - creating 11-7
 - creating SQL Server 11-8
 - Database, creating 11-10
 - errors 12-27
 - Profiles 12-1
 - adding event monitors to 14-23
 - creating ESSM 12-2
 - deleting 12-4
 - distributing 12-20
 - distribution order C-1
 - in Database Profile Manager C-4
 - managing subscribers 12-11
 - populating 12-5
 - SQLDatabaseProfile* 11-5, 11-6
 - SQLDbDeviceProfile* 11-5
 - SQLDumpDeviceProfile* 11-5
 - SQLGroupProfile* 11-6
 - SQLLoginProfile* 11-5
 - SQLRemoteServerProfile* 11-5
 - SQLSegmentProfile* 11-6
 - SQLServerProfile* 11-5
 - SQL Server Profile Manager in C-2
 - SQLUserProfile* 11-6
 - subscribing to Sentry 14-26
 - unsubscribing from Sentry 14-26
 - Profiles container 11-15, 11-19
 - Program requesting, process 4-21
 - Prompts
 - volume change 7-47
 - Properties
 - changing
 - roles required D-3
 - displaying database 7-7
 - displaying default 9-13
 - displaying dump device 7-41
 - displaying segment 7-27
 - object 9-4
 - tab 9-4
 - Properties menu 2-12
 - Properties toolbar button 3-17
 - Public
 - group permissions 8-43 to 8-47
 - membership 8-20
 - Pull-down menus 3-13
- ## R
- Read Only Database database
 - option 7-20
 - Recompiling procedures and triggers 10-40
 - Recovery
 - clean 6-19
 - configuration parameters A-11
 - database 7-49
 - mirroring benefits 6-12, 6-20
 - notices 1-10
 - planning 7-36, 7-49
 - separating logs 6-20
 - strategies 6-1
 - transaction log 7-49
 - recovery flags configuration
 - parameter A-3
 - recovery interval configuration option A-3
 - recovery interval in minutes configuration
 - parameter A-9
 - Reducing database size 7-9
 - Re-enabling disk mirroring 6-18
 - references 8-6
 - references permission 8-27, 8-39, 9-70
 - Referential constraints 10-2, 10-3
 - description of 10-35
 - foreign keys in 10-37
 - primary keys in 10-37, 10-39
 - Referential integrity, enforcing 9-48, 10-35
 - Refresh button 3-27
 - Refreshing windows and dialog boxes 3-27

- Refresh toolbar button 3-17
- Regular expression
 - monitoring with 14-17
- remote.ddl* file 4-35
- remote access 4-14
- remote access configuration
 - parameter A-3
- remote connections configuration
 - parameter A-3
- Remote login mapping 4-26
- remote logins configuration
 - parameter A-3
- Remote procedure calls (RPCs) 4-23
- remote server pre-read packets configuration
 - parameter A-9
- Remote Server Properties dialog
 - box 4-30
- Remote servers
 - communicating with 4-23 to 4-31
 - deleting 4-26
 - displaying 4-30
 - interfaces file and 4-23
 - login management 4-26
 - options 4-25
 - setting up for 4-23
- Remote servers, creating 4-24
- remote sites configuration parameter A-3
- Remove button 7-5, 9-33, 10-8, 10-16
- Removing
 - columns 10-8, 10-16
 - device allocation 7-5
 - parameters 9-33
 - users from groups 8-34
- Removing profile copies 14-26
- Replacement strategy, MRU 9-24
- Representation, visual 1-3
- Requesting program, process 4-21
- Resetting configuration
 - parameters 4-15, 4-18
- Resources 1-7
 - Administrators* resource 2-3
 - default attribute values for 1-8
 - managing
 - roles required D-7
 - policy regions, configuring 2-11
 - roles 2-3
 - roles required for adding types to policy regions 2-12
- Response action 14-7
- Response level 14-6
- Restarting SQL Server 4-2, 4-4
- Restore
 - Changing volumes 7-47, 7-53
 - changing volumes 7-47
- Retrieving data 9-19
- Revoking
 - access 8-48
 - command permissions 8-30, 8-41
 - object permissions 9-38
 - procedure permissions 9-36
 - table permissions 10-22
 - view permissions 9-68
 - with cascade 8-6
 - with cascade 9-38
- Right pane
 - profile manager window 11-15
- Roles
 - access 8-2, 8-48
 - assigning 2-2 to 2-5, 8-10, 8-13
 - changing object properties D-3
 - creating objects D-2
 - damaged databases, deleting 7-7
 - database devices, creating 6-3
 - database devices, deleting 6-6
 - database devices, displaying 6-7
 - database devices, modifying 6-7
 - database management D-8, D-9
 - database options, displaying 7-18
 - database options, modifying 7-18
 - database properties, displaying 7-7
 - database properties, modifying 7-7
 - databases, backing up 7-44
 - databases, checking consistency 7-55
 - databases, creating 7-3
 - databases, creating backup tasks 7-44
 - databases, creating restore tasks 7-50
 - databases, deleting 7-6
 - databases, restoring 7-50

- deleting objects D-5
- displaying objects D-6
- dump devices, creating 7-39
- dump devices, deleting 7-40
- groups, creating 8-31
- groups, deleting 8-33
- groups, displaying 8-36
- groups, modifying 8-36
- logins, creating 8-9
- logins, deleting 8-11
- managing resources D-7
- remote servers, creating 4-24
- remote servers, deleting 4-26
- remote servers, displaying 4-30
- remote servers, modifying 4-30
- segments, creating 7-24
- segments, deleting 7-26
- segments, displaying 7-27
- segments, modifying 7-27
- SQL Server, starting 4-2
- SQL Server configuration variables,
 - modifying 4-14
- SQL Server processes,
 - displaying 4-20
- SQL Server processes, killing 4-22
- System Administrator 4-34
- System Administrators 6-1
- users, creating 8-19
- users, deleting 8-21
- users, displaying 8-23
- users, modifying 8-23

Rollback

- notices 1-10

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 9-41
- creating 9-41
- defining values 9-40
- Dependencies tab 9-43

- displaying 10-21
- displaying dependencies 9-43
- expression in definition of 9-42
- modifying 9-40
- navigating to dependencies 9-44
- policy 13-1
- unbinding 9-47
- ways of creating 9-33, 9-40, 10-3

runnable process search count configuration

- parameter A-9

Running sp_estspace 7-2

S

sa_role 2-7

Scan Back Time 14-16

Scheduling back ups 7-42

Scheduling tasks 1-6

schema role 2-2

Schemas

- creating objects 9-4

Scripts 1-6

- with Event Monitoring Services 14-4

Scripts, DDL

- editing 4-35 to 4-37, 7-16 to 7-17
- files containing 4-35 to 4-37, 7-16 to 7-17
- generating 4-32

Searching. *See* Finding.

Search strings

- case sensitivity 3-8
- masking characters 3-8
- wildcard characters 3-8

security role 2-2

Segment Contains tab 7-30

Segment Properties tab 7-28

Segments

- creating 7-24
- databases 6-10
- definition 7-23
- deleting 7-26
- devices 6-10
- displaying 9-27
- displaying properties 7-27

- extending 7-29
- indexes on 7-30, 9-21
- navigating to 6-11
- navigating to devices 7-29
- navigating to objects 7-31
- performance benefits 7-23
- storing databases 7-23
- tables on 7-30
- thresholds on 7-31 to 7-35
- Segment space
 - monitoring 14-11
- Segment Thresholds tab 7-33
- select
 - statement 9-62
- Selecting display formats 3-6
- select permission 8-27, 8-39, 9-70
- Sentry
 - command line 14-26
 - UNIX name 14-4
- Sentry engine
 - starting and stopping 14-30
- Sentry Profile
 - adding to profile manager 14-21
- Sentry profile
 - subscribing to 14-26
 - unsubscribing 14-26
 - using with Event Monitoring Services 14-6
- Sentry profiles 14-4 to 14-27
 - adding monitoring sources to 14-20 to 14-27
 - distributing 14-5
- Separating
 - disk usage 6-20
 - logs 6-20
 - performance benefits 6-20
- Server Consistency Check dialog box 4-38
- Server Generate DDL dialog box 4-33
- Server menu 3-12
- Server Processes dialog box 4-20
- Server Properties dialog box 4-12
- server role 2-2
- Server Run DDL dialog box 4-34
- Servers
 - configuring 4-14
 - displaying properties 4-12
 - killing processes 4-22
- Server user ID 8-13
- Set Database Online command 7-60
- Set Login Names dialog box 2-7
- Set Managed Resources dialog box 2-12
- Set SQL Server Logins button 2-8
- Setting
 - Cache Strategy 10-20
 - default policy 13-3
 - error log file attribute 14-19
 - validation policy 13-8
- Setting a database online 7-60
- Set TMR Roles dialog box 2-4
- Severity level
 - monitoring 14-17
- shared memory starting address configuration parameter A-9
- Sharing
 - logins 8-2, 8-15, 8-24
 - permissions 8-2, 8-15, 8-24, 8-31
- Shortcut menu commands 3-14, 11-19
- Shortcut menus 3-14
 - in profile managers 11-19
- Shortcuts xxxiii
 - drag-and-drop 3-9
 - navigating to segments 6-11
 - updating permissions 9-71, 10-25
- Single-User Mode database option 7-20
- Size
 - estimating database 7-2
 - increasing database 7-9
 - reducing database 7-9
- size of auto identity column configuration parameter A-9
- Sorting data 9-24
- Sorting object details 3-7
- sort page count configuration parameter A-9
- source, monitoring 14-1
- sp_estspace procedure 7-2
 - estimating space requirements 7-2

- Space
 - allocating on segments (tables) 10-26
- space role 2-2
- Specifying
 - validation policy 13-8
- spid, blocking 4-21
- SQLCacheProfile 11-5
 - distribution behavior and dependencies C-4
 - policy, use in 13-4
- SQLDatabaseProfile 11-15
 - distribution behavior and dependencies C-3, C-4
 - policy, use in 13-4
- SQLDatabaseProfile* 11-5, 11-6
- SQLDatabaseProfile container 11-20
- SQLDatabaseProfileMgr 11-7
- SQLDataTypeProfile 11-6
 - distribution behavior and dependencies C-7
 - policy, use in 13-5
- SQLDbDeviceProfile
 - distribution behavior and dependencies C-2
 - policy, use in 13-4
- SQLDbDeviceProfile* 11-5
- SQLDefaultProfile 11-6
 - distribution behavior and dependencies C-6
 - policy, use in 13-5
- SQLDumpDeviceProfile
 - distribution behavior and dependencies C-2
 - policy, use in 13-4
- SQLDumpDeviceProfile* 11-5
- SQLGroupProfile
 - distribution behavior and dependencies C-5
 - policy, use in 13-5
- SQLGroupProfile* 11-6
- SQLIndexProfile 11-6
 - distribution behavior and dependencies C-8
 - policy, use in 13-5
- SQLLoginProfile
 - distribution behavior and dependencies C-3
 - policy, use in 13-4
- SQLLoginProfile* 11-5
- SQLProcedureProfile 11-6
 - distribution behavior and dependencies C-8
 - policy, use in 13-5
- SQLRemoteServerProfile
 - distribution behavior and dependencies C-3
 - policy, use in 13-4
- SQLRemoteServerProfile* 11-5
- SQLRuleProfile 11-6
 - distribution behavior and dependencies C-6
 - policy, use in 13-5
- SQLSegmentProfile
 - distribution behavior and dependencies C-5
 - policy, use in 13-5
- SQLSegmentProfile* 11-6
- SQL Server
 - See also* Managed SQL Servers
 - connecting to 2-6
 - host machine 2-14
 - managing 2-16 to 2-17
 - starting 4-2
 - status, displaying 4-3
 - stopping 4-4
 - on unsupported platforms 2-15
- SQL Server 11.0 features
 - Add Identity Column in Non-Unique Indexes database option 7-21
 - changing database log buffer size 7-12
 - checking network I/O 4-38
 - configuration 4-17
 - how described xxxiii, xxxv
 - maximum index rows per page 9-21
 - maximum table rows per page 10-6, 10-19
 - named cache management 5-1 to 5-14

- process information 4-21
- table partitioning 10-32
- table unpartitioning 10-34
- SQL Server administration
 - configuration parameters A-14
- sql server clock tick length configuration
 - parameter A-9
- SQL Server events
 - monitoring 14-1 to 14-31
- SQL Server Host box 2-16
- SQL Server login 14-13
 - monitoring 14-13
- SQL Server Login box 2-8
- SQL Server logins
 - assigning to administrators 2-6 to 2-8
 - for multiple management levels 2-6
 - roles required for assigning 2-7
 - for UNIX logins 2-6
- SQL Server Logins for Administrator
 - dialog box 2-8
- SQL Server Manager
 - about 1-1
 - customizing display of 3-6
 - windows and dialog boxes of 3-1
- SQL Server monitoring sources 14-7
- SQL Server Password box 2-8
- SQLServerProfile 11-15
 - distribution behavior and dependencies C-2
- SQLServerProfile* 11-5
- SQLServerProfile container 11-20
- SQL Server Profile Manager 11-5
 - associations to 11-9
 - creating 11-8
 - opening 11-13
 - profiles in C-2
- SQL Server Profile Manager. *See also* Profile manager xxxvii
- SQL Server Profile Managers
 - overview 1-8
- SQL Server profile managers 11-4
 - associations to 11-9
- SQLServerProfileMgr 11-7
- SQL Server Reference Manual* xxxii, 7-21
- SQL Server roles
 - and administrator SQL Server logins 2-7
 - oper_role 2-7
 - sa_role 2-7
 - sso_role 2-7
- SQL Server Status
 - monitoring 14-12
- SQL Server Utility Programs*
 - manual xxxii
- SQL Server window
 - opening 4-10
 - opening from a profile manager 11-21
- SQL statements 9-4, 9-33, 9-50
- SQLTableProfile 11-6
 - distribution behavior and dependencies C-7
 - policy, use in 13-5
- SQLTriggerProfile 11-6
 - distribution behavior and dependencies C-9
 - policy, use in 13-5
- SQLUserProfile
 - distribution behavior and dependencies C-5
 - policy, use in 13-5
- SQLUserProfile* 11-6
- SQLViewProfile 11-6
 - distribution behavior and dependencies C-8
 - policy, use in 13-5
- sso_role 2-7
- stack guard size configuration
 - parameter A-9
- stack size configuration parameter A-3, A-10
- Standard configuration parameter
 - display level 4-17
- Standard menus 3-11
- Standard toolbar 3-16
- Start Date 14-16
- Starting
 - Enterprise SQL Server Manager 3-1
 - Sentry engine 14-30

- Starting Backup Server 4-2
 - Starting SQL Server 4-2
 - Start Time 14-16
 - Start time, transaction 4-21
 - Statements
 - SQL 9-4, 9-50
 - Statistics, updating 10-45
 - Status
 - SQL Server
 - monitoring 14-12
 - Status, SQL Server 4-3
 - Status bar 3-19
 - command help 3-19
 - filtering and 3-9, 3-19
 - object count 3-19
 - Status Processes
 - monitoring bad 14-15
 - Stopped Processes
 - monitoring 14-15
 - Stopping
 - Backup Server 4-4
 - Enterprise SQL Server Manager 3-1
 - Sentry engine 14-30
 - Stopping SQL Server 4-4
 - Storing
 - allocating space 7-2
 - considerations 6-19, 7-2
 - costs 6-12
 - databases 6-3
 - default devices 6-3
 - increasing space 7-9
 - logs 6-3
 - segment benefits 7-23
 - strategies 6-19
 - transaction logs 7-4
 - Strategies
 - recovery 6-1
 - storing 6-19
 - Strategy
 - setting cache 10-20
 - Subscriber
 - distribution of policy to 13-2
 - Subscriber container 11-15
 - Subscribers
 - adding to profile managers 12-14
 - in a profile manager 11-12
 - managing 12-11
 - profile managers as 12-12
 - Subscribers container 11-19
 - Subscribers dialog box 12-14
 - Subscribing to a Sentry Profile 14-26
 - suid. *See* Server user ID.
 - Summary of constraints 10-4
 - Sybase Administration Notice
 - Group 1-10
 - Sybase Backup/Recovery Notice
 - Group 1-10
 - Sybase Login
 - monitoring 14-13
 - Sybase Troubleshooting Guide* xxxii
 - sybssystemprocs* database 6-3
 - Symmetrical drag-and-drop 3-9
 - Synchronize Profile 11-18
 - sysdevices* table 6-2
 - System Administration Guide* xxxii, 7-36, A-1
 - System Administration Guide Supplement* 4-23
 - System administration training xxxii
 - System Administrators
 - changing database defaults 7-18
 - generating DDL scripts 4-34
 - issuing checkpoints 7-22
 - managing resources 6-1
 - permissions 8-3
 - transferring database ownership 7-12
 - System Security Officers 8-9
 - permissions 8-3
 - remote login mapping 4-26
 - system* segment 7-23
 - System table 10-1
 - System tables, updates to A-4
 - systemwide password expiration
 - configuration parameter A-10
- T**
- Table, system 10-1

- Table, user 10-1
- Table check constraint 9-40
- Table Check Constraints tab 10-30
- Table Column Advanced dialog box 10-9, 10-17
- Table Consistency dialog box 10-41
- Table data
 - copying 10-45
- Table Dependencies tab 3-22, 10-21
- Table Indexes/Triggers tab 9-7, 10-28
- table lock spinlock ratio configuration
 - parameter A-10
- Table Partitions tab 10-33, 10-34
- Table Permissions tab 10-22
- Table Properties tab 3-21, 9-5, 10-14
- Table Referential Constraints tab 10-36
- Tables
 - adding columns 10-15
 - allocating pages 10-43
 - binding to a named cache 5-10
 - cache replacement strategy 10-6, 10-20
 - check constraints 10-11, 10-19, 10-30, 10-31
 - column constraints 10-17
 - column definitions 10-8
 - copying data 12-25
 - Create dialog box 10-5
 - creating 10-1, 10-5
 - deleting 10-12
 - Dependencies tab 10-21
 - displaying 9-35, 10-21
 - displaying dependencies 10-21
 - estimating space requirements 7-2
 - indexes 9-21
 - Indexes/Triggers tab 10-28
 - maximum rows per page 10-6, 10-19
 - navigating to 7-31
 - navigating to dependencies 10-21, 10-27
 - partitioning 10-32
 - permissions 10-22
 - Permissions tab 10-22
 - referential constraints 10-35
 - removing columns 10-8, 10-16
 - segments 7-30
 - Segments tab 10-26, 10-27
 - shortcuts 10-25
 - sysdevices* 6-2
 - unpartitioning 10-34
 - updating statistics 10-45
- Table Segments tab 10-26
- Tabs
 - access 8-48
 - binding objects 9-16
 - buttons 3-20, 9-7
 - dialog boxes 3-20
 - displaying dependencies 9-14
 - displaying properties 9-4
 - granting permissions 8-48
- tape retention configuration
 - parameter A-3
- tape retention in days configuration
 - parameter A-10
- target 11-2
- Task Library 1-6
- Tasks, administration 1-2
- tcp no delay* configuration parameter A-10
- Technical Support, contacting xxxvii
- tempdb* database 6-3
- Threshold 14-6
- Thresholds
 - for process-specific monitors 14-15
 - generic event monitors 14-7
 - monitoring with EMON 14-1
- Thresholds, segment
 - creating 7-32
 - deleting 7-34
 - hysteresis value 7-32
 - last chance 7-32
 - modifying 7-35
 - purpose of 7-31
- Time blocked, process 4-21
- time slice configuration parameter A-3, A-10
- Tivoli/Sentry 14-1
 - and High-Level Event Services 1-9
- Tivoli/Sentry Profile dialog box 14-23

- Tivoli Management Environment
 - preparing for managing SQL Servers 2-1 to 2-13
 - See TME
 - Tivoli Management Platform
 - ESSM and 1-4
 - Tivoli Management Regions
 - roles, TMR 2-3
 - Tivoli Monitoring Technology 1-9, 14-1
 - and Event Monitoring Services 14-1 to 14-7
 - TME roles
 - assigning 2-2 to 2-5
 - cache role 2-2
 - dump role 2-2
 - Enterprise SQL Server Manager additions to 2-2
 - load role 2-2
 - monitor role 2-2
 - resource-level roles 2-3
 - schema role 2-2
 - security role 2-2
 - server role 2-2
 - space role 2-2
 - TMR-level roles 2-3
 - TMP
 - See Tivoli Management Platform
 - TMR roles 2-3
 - roles required for assigning 2-3
 - TMT See Tivoli Monitoring Technology
 - TMT command line interface 14-26
 - Toolbar, standard 3-16
 - Toolbar buttons 3-16
 - Toolbars
 - creating objects 1-3
 - hiding 3-16
 - ToolTip 3-18
 - total data cache size configuration
 - parameter A-10
 - total memory configuration
 - parameter A-10
 - Training xxxii
 - Transaction, active 4-21
 - Transaction log backup options 7-45
 - Transaction logs 6-3, 6-20
 - allocating space for 7-2
 - backing up 7-20
 - checkpoints 7-22
 - dumping 7-20
 - recovering 7-49
 - storing 7-4
 - Transaction start time 4-21
 - Transact-SQL commands 4-32
 - DDL 7-13
 - Transferring database ownership 7-12
 - Trigger Dependencies tab 9-53
 - Trigger Properties tab 9-52
 - Triggers
 - commands 9-48
 - Create dialog box 9-49
 - creating 9-49
 - Dependencies tab 9-53
 - displaying 9-35, 10-21, 10-28
 - displaying dependencies 9-53
 - enforcing referential integrity 9-48
 - executing 9-50
 - navigating to 10-29
 - navigating to dependencies 9-53
 - nested A-4
 - object owners 9-48
 - recompiling 10-40
 - Troubleshooting, Event Monitoring Services 14-30
 - Truncate and Log transaction log
 - backup option 7-45
 - Truncate and No Log transaction log
 - backup option 7-45
 - Truncate Transaction Log on Checkpoint
 - database option 7-20
- ## U
- Unbinding
 - columns 9-18, 9-47
 - datatypes 9-47
 - defaults 9-18
 - objects from a named cache 5-14
 - user datatypes 9-18

- Unique
 - indexes 9-19, 9-22
 - values 9-22
 - unix_login 14-13
 - UNIX logins 2-6
 - Unpartitioning a table 10-34
 - Unpopulate 12-6
 - Unsubscribing from a Sentry
 - Profile 14-26
 - update permission 8-27, 8-39, 9-70
 - Updating
 - Voyager windows 3-27
 - Updating permissions
 - procedures 9-36
 - tables 10-22
 - views 9-68
 - Updating statistics 10-45
 - upgrade version configuration
 - parameter A-3, A-10
 - Usable by Database Owner Only
 - database option 7-20
 - User Command Permissions tab 8-30
 - user connections configuration
 - parameter A-4
 - User Datatype Dependencies tab 9-58
 - User Datatype Properties tab 9-57
 - User datatypes
 - binding defaults 9-17
 - binding rules 9-46
 - Create dialog box 9-55
 - creating 9-55
 - Dependencies tab 9-58, 9-59
 - displaying 9-14, 9-35, 10-21
 - displaying dependencies 9-58
 - modifying 9-54
 - navigating to dependencies 9-59
 - unbinding 9-18, 9-47
 - User-defined datatypes. *See* User datatypes
 - User environment configuration
 - parameters A-15
 - User interface 1-5 to 1-6
 - command line interface 1-6
 - graphical user interface 1-5
 - user log cache size configuration
 - parameter A-10
 - user log cache spinlock ratio configuration
 - parameter A-10
 - User Object Permissions tab 8-26
 - User Objects Owned tab 8-25
 - User permissions
 - procedures 9-36
 - tables 10-22
 - views 9-68
 - User processes, displaying 4-20
 - User Properties tab 8-23
 - Users
 - assigning aliases 8-24
 - changing groups 8-24
 - creating 8-19
 - deleting 8-21
 - displaying 8-22
 - displaying objects owned 8-25
 - granting command permissions 8-30
 - granting object permissions 8-26, 8-37
 - guest 7-4
 - navigating to 8-18, 8-36, 9-39, 9-71, 10-25
 - removing from groups 8-34
 - User-specified events
 - monitoring 14-14
 - User table 10-1
 - using online help xxxvi
 - Utilities
 - bcp 7-20
 - dbcc 7-55, 10-40
 - writetext 7-20
- ## V
- Validate Profile 11-18
 - Validation Policy 11-18
 - Validation policy 1-8, 13-1
 - See also* Policy 1-8
 - defining 13-8
 - effect of 13-8
 - overview 1-8
 - See also* Policy

Validation policy. *See also* Policy, Default policy xxxvii

Valid resources, policy region list of 2-11

Values

default column 9-10

index keys 9-22

null 9-56

rules defining 9-40

unique 9-22

Variables

ENDPOINT 14-4

environment

in Event Monitoring Services 14-4

View Dependencies tab 9-66

View Permissions tab 9-69

View Properties tab 9-65

Views

adding columns 9-60

Create dialog box 9-61

creating 9-61

Dependencies tab 9-66, 9-67

displaying 9-35, 10-21

displaying dependencies 9-66

modifying 9-60

navigating to dependencies 9-67, 9-71

permissions 8-45, 9-68

Permissions tab 9-68

Visual representation 1-3

Volume Change dialog box 7-47

Voyager window 3-2

container icons 3-3

Contents bar 9-6

customizing the display of 3-6 to 3-9

description of 3-2 to 3-10

dragging and dropping 3-9 to 3-10

menus and toolbars 3-11 to 3-19

moving through 3-4

object icons 3-3

opening new windows 3-5

refreshing 3-27

W

waddmon command 14-26

Wash area 5-9

Wildcard characters 3-8

Window

left pane of profile manager 11-15

profile manager 11-13

right pane of profile manager 11-15

Window menu 3-12

Windows

status bar 3-19

With cascade 8-6

with cascade permission 9-38

With grant 8-6

with grant permission 9-38

wiseng command 14-30

writetext utility 7-20

wstopeng command 14-30

