# SYBASE SQL Server<sup>TM</sup> Error Messages

SYBASE SQL Server Releases 4.2–10.0.2 Document ID: 38574-01-1000-03 Last Revised: August 23, 1995

#### Principal author: Diana Barrett

Contributing authors: Lance Andersen, Chris McCartney, Anup Mehra, Cristina Nitescu, Kathy Saunders

Document ID: 38574-01-1000

This publication pertains to SYBASE SQL Server releases 4.2 through 10.0.2 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 the agreement.

# **Document Orders**

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

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

Upgrades are provided only at regularly scheduled software release dates.

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

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

# Sybase Trademarks

SYBASE, the SYBASE logo, APT-FORMS, Data Workbench, DBA Companion, Deft, GainExposure, GainInsight, Gain Momentum, PowerBuilder, Powersoft, Replication Server, SQL Advantage, SQL Debug, SQL SMART, SQL Solutions, SQR, Transact-SQL, and VQL are registered trademarks of Sybase, Inc. Adaptable Windowing Environment, ADA Workbench, AnswerBase, Application Manager, Applications from Models, APT-Build, APT-Edit, APT-Execute, APT-Library, APT-Translator, APT Workbench, Backup Server, Bit-Wise, Build Momentum, Camelot, Client-Library, Client/Server Architecture for the Online Enterprise, Client/Server for the Real World, Client Services, Configurator, Connection Manager, Database Analyzer, DBA Companion Application Manager, DBA Companion Resource Manager, DB-Library, Deft Analyst, Deft Designer, Deft Educational, Deft Professional, Deft Trial, Developers Workbench, Easy SQR, Embedded SQL, Enterprise Builder, Enterprise Client/Server, Enterprise CONNECT, Enterprise Manager, Enterprise Meta Server, Enterprise Modeler, Enterprise Momentum, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, EWA, Gain, 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,

Navigation Server Manager, Net-Gateway, Net-Library, Object Momentum, OmniSQL Access Module, OmniSQL Gateway, OmniSQL Server, OmniSQL Toolkit, Open Client, Open Client/Server Interfaces, Open Gateway, Open Server, Open Solutions, PC APT-Execute, PC DB-Net, PC Net Library, PostDoc, 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 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 Momentum, SYBASE SQL Lifecycle, Sybase Synergy Program, SYBASE Virtual Server Architecture, SYBASE User Workbench, SyBooks, System 10, Tabular Data Stream, The Enterprise Client/Server Company, The Online Information Center, Warehouse WORKS, Watcom SQL, WebSights, WorkGroup SQL Server, XA-Library, and XA-Server are trademarks of Sybase, Inc.

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

# **Restricted Rights**

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

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

# **Table of Contents**

# Preface

Audience xvii
What This Manual Contains xvii
Changes to SYBASE SQL Server Error Messages xix
Your Comments About the Troubleshooting Guides xx
Style Conventions
Style Conventions in Text xxi
SQL Syntax Conventions xxi
If You Need Help xxii

# 1. Introduction

Error Message Severity Levels 1	1-1
Variables in Error Message Text	1-1
Finding Object Names from Error Message Text	1-2
Creating Error Messages	1-2
Reporting Errors	1-2
Faxing Error Log Fragments	1-5
Reproducing Problems	1-5

# 2. Writeups

Parser Errors	-1
Error 102	-2
Sequencer Errors	-4
Error 207	-5
Error 208	-7
Error 213 2-2	10
Error 225 2-	11
Error 226 2-2	12
Error 229 2-2	14
Error 232 2-2	15
Error 259 2-2	16
Error 268 2-2	18
Compile Errors	21
Error 403 2-2	22

Error 404	2-23
Error 428	2-24
Error 430	2-25
Query Processor Errors	2-26
Error 511	2-27
Error 515	2-29
Access Errors	2-31
Error 601	2-32
Error 603	2-34
Error 605	2-36
Error 611	2-40
Error 614	2-41
Error 623	2-45
Error 624	2-47
Error 625	2-51
Error 629	2-54
Error 631	2-56
Error 632	2-60
Error 644	2-61
Memory Manager Errors	2-63
Error 701	2-64
Error 702	2-66
Error 703	2-68
Error 706	2-70
Error 707	2-71
Buffer Manager Errors	2-72
Error 803	2-73
Error 804	2-76
Error 806	2-77
Error 813	2-80
Error 821	2-82
Error 822	2-84
Error 823	2-88
Error 840	2-90
Open Database Manager Errors	2-93
Error 903	2-94
Error 905	2-96
Error 906	2-98
Error 911	2-99

Error 913	)1
Error 916	)2
Error 921	)4
Error 924	)6
Error 925	)7
Error 926	0
Error 930	2
Error 933 2-11	15
Error 945	17
Page Manager Errors. 2-11	9
Error 1105	20
Error 1108	35
Error 1117	37
Error 1120	39
Error 1129	11
Error 1133	13
Error 1134	16
Error 1142	17
Error 1143 2-15	50
Lock Manager Errors	52
Error 1203	53
Error 1204	54
Error 1205	57
Error 1207	30
Error 1265	31
Sort Manager Errors 2-16	32
Error 1501	33
Error 1505	35
Error 1508	37
Error 1509	39
Error 1510	71
Error 1520	73
Error 1530	75
Error 1531	76
Initialization Errors	18
Error 1601	19
Error 1602	31
Error 1605	33
Error 1608	37

Error 1613 2-	-189
Error 1623 2-	-190
Create Utilities Errors 2-	-192
Error 1702 2-	-193
Error 1808	-194
Error 1809 2-	-196
Error 1810	-197
Error 1811	-198
Error 1813 2-	-199
Error 1902 2-	-200
Error 1903 2-	-201
Error 1904 2-	-202
Error 1916 2-	-203
Character Set Conversion Errors	-205
Error 2401	-206
Error 2402	-208
Error 2409	-210
<i>dbcc</i> Errors	-212
Error 2501	-213
Error 2502	-215
Error 2503	-217
Error 2506	-219
Error 2507	-222
Error 2509 2-	-225
Error 2510 2-	-228
Error 2511	-230
Error 2513 2-	-232
Error 2514	-234
Error 2517 2-	-237
Error 2519 2-	-239
Error 2520 2-	-240
Error 2521	-241
Error 2524	-247
Error 2525 2-	-250
Error 2526	-253
Error 2529	-254
Error 2540	-256
Error 2541	-261
Error 2543 2-	-263

Error 2544 2-265
Error 2546 2-267
Error 2550 2-272
Error 2558 2-274
Error 2559 2-282
Error 2571 2-284
Error 2572 2-286
Error 2573 2-288
Error 2596 2-289
<i>insert</i> Errors
Error 2601
Error 2610
<i>Error 2615</i>
Error 2620
Create Utilities Errors (Continued) 2-300
<i>Error 2714</i>
Procedure Manager Errors 2-302
Error 2811
<i>Error 2812</i>
Error 2824
<i>dump</i> and <i>load</i> Errors 2-308
<i>Error 3002</i>
<i>Error 3004</i>
Error 3101
<i>Error 3104</i>
Error 3105 2-316
Error 3120 2-317
<i>Error 3201</i>
Error 3202
Error 3203 2-325
Error 3225 2-327
Undo and Redo Errors 2-328
Error 3307 2-329
Recovery Errors
Error 3403 2-332
Error 3414 2-334
Error 3418 2-335
Error 3425 2-336

Error and Exception Handling Errors	2-337
Error 3604	2-338
Error 3621	2-339
Error 3626	2-340
drop Errors	2-342
Error 3701	2-343
Error 3702	2-345
Error 3703	2-347
Transaction Errors	2-349
Error 3904	2-350
Data Server Errors	2-352
Error 4002	2-353
dump and load Errors (Continued)	2-357
Error 4204	2-358
Error 4207	2-362
Error 4219	2-364
Error 4305	2-367
Error 4306	2-368
Bulk Copy Utility Errors	2-370
Error 4806	2-371
alter database Errors	2-373
Error 5006	2-374
Error 5013	2-376
Disk Errors	2-379
Error 5115	2-380
Error 5123	2-383
Reconfiguration Errors	2-386
Error 5808	2-387
Error 5813	2-389
Process Kill Errors	2-390
Error 6103	2-391
Timestamp Errors	2-392
Error 6902	2-393
Text Manager Errors	2-395
Error 7130	2-396
Distributed Database Network Errors	2-398
Error 7205	2-399
Error 7211	2-401
Error 7212	2-403

Error 7214
Error 7215
Error 7218 2-408
Error 7220
Error 7221
Error 7223
Error 7227
Parser Errors
Error 7364
Threshold Errors         2-422
Error 7401
Error 7402
Error 7403
Error 7404
Error 7405
Error 7406
Error 7407 2-432
Error 7408
Error 7409 2-433
Error 7410 2-434
Error 7411
Error 7412
Error 7413
Error 7414
Error 7415
<i>dbcc</i> Errors (Continued) 2-440
Error 7902
Error 7930
Error 7939 2-448
Error 7940
Error 7948
Error 7949 2-458
Descriptor Manager Errors 2-463
Error 8201
<i>Error 8203</i>
<i>Error 8204</i>
Error 8207
Error 8211

sysindexes Manager Errors 2-	-471
Error 8402 2-	-472
Error 8412 2-	-474
Site Buffer Manager Errors 2-	-475
Error 8704 2-	-476
Threshold Errors (Continued) 2-	-477
Error 17417 2-	-478
Error 17715 2-	-481
Error 17716 2-	-482
Error 17870 2-	-483
Error 17871 2-	-484
Error 17872 2-	-485
Error 17873 2-	-486
Error 17874 2-	-487
Error 17875 2-	-488
Error 17903 2-	-489
Error 17904 2-	-490
Error 17905 2-	-491
Error 17906 2-	-492
Error 17910 2-	-493
Error 18031 2-	-494
Error 18032 2-	-495
Error 18033 2-	-496
Unnumbered Errors	-497
Buffer Mismatch Error 2-	-498
Character Set Error	-499
Could Not Create Shared Memory Error 2-	-501
Current Process Infected with %d	-502
Dstart I/O Request Repeatedly Delayed 2-	-504
Failed to Open Device Error    2-	-509
File Already in Use Error    2-	-510
Memory Too Fragmented Error 2-	-511
Network Information Message 2-	-513
No More Alarms Available Error 2-	-515
Open Objects Parameter May Be Too Low	-517
os_attach_region Error	-519
os_create_region Errors 2-	-520
Read/Write Error	-523
Server Is Unnamed 2-	-525

Stack Guardword Error	2-526
t_rcvconnect Error	2-528

Glossary

Index

# List of Tables

Table 1:	SQL syntax conventions	xxi
Table 1-1:	Error message text variables and meanings	1-1
Table 2-1:	Hardware causes of Error 605	2-37
Table 2-2:	Software causes of Error 605	2-37
Table 2-3:	Allocation error: type of space	2-120
Table 2-4:	Causes of Error 1105 in tempdb	2-125
Table 2-5:	Error 1520 correction commands	2-173
Table 2-6:	Compatibility chart for index types and index options	2-203
Table 2-7:	Data types and object/index ID values	2-279

# Preface

SYBASE SQL Server Error Messages describes error conditions and troubleshooting procedures for problems that SYBASE® users may encounter when using SYBASE SQL Server<sup>™</sup>. The error messages addressed here are those which result in the most calls to Sybase Technical Support. The manual's purpose is:

- To provide enough information about certain error conditions so that you can resolve problems without help from Technical Support
- To provide lists of information that you can gather before calling Technical Support, which will help resolve your problem quickly
- To provide you with a greater understanding of Sybase products

# Audience

This manual is intended for the following audience:

- SYBASE System and Database Administrators
- Sybase Technical Support contacts
- Developers of applications using SYBASE software

This manual assumes that you are thoroughly familiar with the Sybase products. If you are unfamiliar with any of the procedures described in this manual, call Sybase Technical Support for assistance.

# What This Manual Contains

This manual contains information about SYBASE SQL Server error messages, including the text of each message that is documented in this manual, potential causes of the error, and what you can do to recover from the error. The messages are listed in numerical order. Only the most commonly occurring error messages are documented. You can create a complete listing of the SQL Server error messages for your installation by typing the following:

```
1> use master
2> go
1> select * from sysmessages
2> go
```

The error message writeups are arranged numerically in the following sections:

- Parser Errors
- Sequencer Errors
- Compile Errors
- Query Processor Errors
- Access Errors
- Memory Manager Errors
- Buffer Manager Errors
- Open Database Manager Errors
- Page Manager Errors
- Lock Manager Errors
- Sort Manager Errors
- Initialization Errors
- Create Utilities Errors
- Character Set Conversion Errors
- dbcc Errors
- insert Errors
- Procedure Manager Errors
- dump and load Errors
- Undo and Redo Errors
- Recovery Errors
- Error and Exception Handling Errors
- drop Errors
- Transaction Errors
- Data Server Errors
- Bulk Copy Utilities Errors
- alter database Errors
- Disk Errors
- Reconfiguration Errors
- Process Kill Errors

- Timestamp Errors
- Text Manager Errors
- Distributed Database Network Errors
- Threshold Errors
- Descriptor Manager Errors
- sysindexes Manager Errors
- Site Buffer Manager Errors
- Unnumbered Errors

Some error message types fall in more than one number sequence. For example, dbcc errors can be in the 25xx range and also in the 79xx range. Since the error message writeups are presented in numerical order, there is more than one section entitled "dbcc Errors."

# Changes to SYBASE SQL Server Error Messages

The following SQL Server error message writeups have been added:

- Parser error: 102
- Sequencer errors: 213, 226
- Compile errors: 403, 428, 430
- Query Processor error: 515
- Access errors: 601, 632
- Memory Manager errors: 706, 707
- Open Database Manager errors: 903, 905, 916, 921, 925, 945
- Page Manager error: 1134
- Lock Manager error: 1265
- Sort Manager error: 1501
- Initialization error: 1623
- Create utilities errors: 1808, 1813
- Procedure Manager errors: 2811, 2824
- dump and load errors: 3225, 4204, 4219
- Recovery error: 3418
- drop errors: 3702, 3703
- Transaction error: 3904

SYBASE SQL Server Error Messages

- Distributed database network errors: 7212, 7223
- dbcc error: 7930
- Descriptor Manager errors: 8201, 8203, 8204, 8211
- sysindexes Manager errors: 8402, 8412
- Site Buffer Manager error: 8704
- Unnumbered (kernel) errors:
  - Could Not Create Shared Memory Error
  - Failed to Open Device Error
  - File Already in Use Error
  - Memory Too Fragmented Error
  - No More Alarms Available Error
  - os\_attach\_region Error
  - os\_create\_region Errors
  - t\_rcvconnect Error

A glossary of terms used frequently in this book has been added.

# Your Comments About the Troubleshooting Guides

In order to continue to improve the troubleshooting guides (*SQL* Server Troubleshooting Guide, *SQL* Server Error Messages, and Tools and Connectivity Troubleshooting Guide), we need your feedback. Please send your comments about the troubleshooting guides to the email address tsg@sybase.com.

These comments might include:

- Corrections
- Requests for specific additions
- Material you would like to submit
- Comments about which sections are particularly helpful
- Comments about which sections are not clear
- Any other input you might have

#### ► Note

The tsg@sybase.com email address is for comments about the troubleshooting guides. It is not for reporting problems or asking technical questions. To report a problem or ask a technical question, please contact Sybase Technical Support.

# **Style Conventions**

Wherever possible, *SQL Server Error Messages* uses the style conventions of the various product manuals. This section contains a brief summary of those conventions.

## Style Conventions in Text

Commands and script names appear in **bold** type, for example:

To change the isql command terminator....

Object names appear in italics, for example:

Use the installmodel script to complete the installation of the *model* database.

## SQL Syntax Conventions

The conventions for syntax statements in this manual are as follows:

 Table 1:
 SQL syntax conventions

Кеу	Definition	
command	Command names, command option names, utility names, utility flags, and other keywords are in bold.	
variable	Variables, or words that stand for values that you fill in, are in italics.	
{ }	Curly braces indicate that you choose at least one of the enclosed options. Do not include braces in your option.	
[ ]	Brackets mean choosing one or more of the enclosed options is optional. Do not include brackets in your option.	
( )	Parentheses are to be typed as part of the command.	

Кеу	Definition
	The vertical bar means you may select only one of the options shown.
,	The comma means you may choose as many of the options shown as you like, separating your choices with commas to be typed as part of the command.

 Table 1:
 SQL syntax conventions (continued)

SQL syntax statements (displaying the syntax and options for a command) are printed as follows:

sp\_dropdevice [device\_name]

Examples showing the use of Transact-SQL® commands are printed as follows:

1> select \* from publishers
2> go

Examples of output from the computer are printed as follows:

pub_id	pub_name	city	state
0736	New Age Books	Boston	MA
0877	Binnet & Hardley	Washington	DC
1389	Algodata Infosystems	Berkeley	CA
(3 rows	affected)		

# If You Need Help

Help with your SYBASE software is available in the form of documentation and Sybase Technical Support. If you have any questions about the procedures contained in this guide, ask the designated person at your site to contact Sybase Technical Support.

# Introduction

This manual contains explanations of and instructions for resolving some of the most frequently occurring SQL Server error messages. Most of the messages presented here contain error numbers (from the *master..sysmessages* table), but some are error messages that do not have error numbers, and which occur only in the SQL Server error log.

# **Error Message Severity Levels**

A severity level is associated with each numbered error message. It indicates the type of problem SQL Server has encountered. For more detailed information about severity levels, see the *System Administration Guide*.

Inform the SYBASE System Administrator whenever errors with severity levels over 16 occur. The System Administrator is responsible for resolving them and tracking their frequency. The System Administrator should monitor all errors that generate severity levels of 17 through 24. To assist the System Administrator in resolving problems, print out a hard copy of the error log, which contains the back trace from the fatal error.

# Variables in Error Message Text

The following chart explains the symbolic constants that appear in the error message text provided with each error message explanation:

Symbol	Stands For
%d,%D	Decimal number
%x, %X, %.*x, %lx, %04x, %08lx	Hexadecimal number
%s	Null-terminated string
%.*s, %*s, %*.s	String, usually the name of a particular database object
%S_type	SYBASE-defined structure
%с	Single character

#### Table 1-1: Error message text variables and meanings

Symbol	Stands For
%f	Floating-point number
%ld	Long decimal number
%lf	Double floating-point number

Table 1-1: Error message text variables and meanings (continued)

# Finding Object Names from Error Message Text

Many SQL Server error messages print only the logical page number, not the name of the table or index affected. "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* supplies a method to find the table or index name using the logical page number supplied by the error message. You may want to review that section now so that you are familiar with the necessary procedures before an error occurs.

# **Creating Error Messages**

With SQL Server release 4.9 and later, you can create your own error numbers and messages based on SYBASE error numbers. See sp\_addmessage in the *SQL Server Reference Manual* for more information.

# **Reporting Errors**

When you report an error to Sybase Technical Support, include the following information:

- The environment in which the error occurred, including:
  - SQL Server version
  - EBF (Emergency Bug Fix) number, if available
  - Client version
  - Operating system version
  - Hardware platform
  - Production or development environment
- The message number, state, and complete error message text for any SQL Server errors. Include any numbers, database object

types, or database object names that are included in the error message.

- The context in which the message was generated—what command was running at the time. You can help by providing a hard copy of the error log, including all messages displayed from just prior to start-up to immediately before or immediately after the message. Include the version number and date information printed at the beginning of the error log.
- The number of users who were logged into SQL Server when the error occurred.
- The frequency with which the error occurs.
- The client program that encountered the error (isql, Open Client<sup>™</sup> DB-Library<sup>™</sup>, Data Workbench<sup>®</sup>, and so on).

A checklist for reporting errors follows. The purpose of this checklist is to help you collect the information you need to give to Sybase Technical Support. Your primary site contact should have a copy of this checklist.

# **Technical Support Checklist**

To expedite the resolution of your question and help you track it, please fill out this checklist before you call your Technical Support Center.

Site ID (customer number):	Case Number:	Priority:
Support Engineer:		
Date Case Opened:	Date Case Closed:	
SYBASE Server version (include SWR	[EBF] version if applicable):	

Operating system type and version:

Front-end (client) product and version (include SWR [EBF] version if applicable):

Client machine operating system type and version:

Description of the problem (include error numbers, error messages, circumstances in which problem occurred):

Other useful information (include copy of error logs, reproducible case, number of users on the system, average percent of CPU usage, any other pertinent information):

# **Faxing Error Log Fragments**

Whenever Technical Support asks you to fax the error log to Sybase, fax the log from start-up to the error message in question. Be sure to include the following:

• The lines just prior to start-up; for example:

```
kernel: SQL Server/4.9.1/P/Sun4/OS 4.1.2/1/OPT/Mon Aug 24 14:03:45
```

• All the lines between start-up and the error; for example:

```
Error 2559, Severity Level 16: Data page number 0781 is empty.
```

• Any stack traces or unusual messages that occur near the error message in question.

When trace flag 3601 is turned on, SQL Server writes a stack trace to the error log every time a SQL Server error is raised. For troubleshooting purposes, Sybase Technical Support may ask you to turn on this trace flag. If they do, refer to "How to Start SQL Server with Trace Flags" in the *SQL Server Troubleshooting Guide* for instructions for turning on a trace flag.

# **Reproducing Problems**

In general, the time that Sybase Technical Support needs to resolve a problem is greatly reduced if you generate a reproducible test case demonstrating the error. These test cases can range from a single query that is not behaving as expected to a relatively complex code fragment that encounters a problem.

Whenever possible, test cases should be created using the *pubs* or *pubs2* database. In many cases, this is a task that saves you from having to send large amounts of data to Technical Support.

# **2** Writeups

The error message writeups are in order by message number.

Parser Errors

This section includes error messages for the SQL Server Parser.

# Error 102

**Severity Level** 

15

**Error Message Text** 

Incorrect syntax near '%.\*s'.

## Explanation

This error occurs when SQL Server detects a syntax error in a Transact-SQL command or query.

This error can occur when:

- A keyword is spelled incorrectly.
- A keyword or parameter is missing.
- The order of keywords is incorrect.
- You use an editor invoked from isql to write a SQL command or batch, end your SQL with "go" or another terminator, and then run the file from isql. This only causes this error in some cases.
- A script that used to work no longer works after you change the sort order or the character set of a SQL Server (for example, a script that was created on a SQL Server that originally was not case-sensitive won't work now that SQL Server's sort order has been changed to case-sensitive).
- A reserved word has been used in a query (for example select user = suser\_name() raises Error 102 because "user" is a reserved word).
- You have used a variable instead of a specific period of time following a waitfor delay statement. Variables are not supported in this case. For example, the following would raise Error 102:

```
1> declare @t char(9)
2> select @t = "00:00:01"
3> waitfor delay @t
```

- You have used a variable for a database name and your query contains a parser error; Error 102 is raised because the batch is never executed. Steps for executing a query are:
  - 1. Read the batch.
  - 2. Parse the batch.

- 3. Optimize the batch.
- 4. Execute the batch.

Any step that fails ends the processing of the query.

## Action

Check the spelling and syntax of the command specified in the error message. If it is wrong, correct it in your Transact-SQL statement and run it again.

You can invoke an editor such as vi from isql to edit a SQL statement or statements. However, for the statements to execute, you have to enter the command termination string (usually "go") after you return to isql. If you include the command termination string in the editor, then save the file and return to isql, the statement will not execute. If you enter the command termination string again after returning to isql, you get the following error ("vi" is the editor in this example; you may be using a different editor):

```
1> vi
1> select * from sysmessages where error = 102
2> go
3> go
Msg 102, Level 15, State 1:
Server 'REL1002_NAME', Line 2:
Incorrect syntax near 'go'.
```

# **Additional Information**

Refer to the documentation for the utility you are using for correct syntax information. Some examples are:

- SQL Server Reference Manual
- Transact-SQL User's Guide
- SQL Server Utility Programs for your operating system

# **Sequencer Errors**

This section includes error messages for the SQL Server Sequencer. The Sequencer is the subsystem that controls command execution. The Sequencer handles the processing between steps in commands and also handles the procedural steps between commands such as if, goto, and while.

# Error 207

#### **Severity Level**

16

#### **Error Message Text**

Invalid column name '%.\*s'.

## Explanation

This error occurs when SQL Server tries to use a column name that does not exist.

As of release 10.0, SQL Server supports delimited identifiers for table, view, and column names. Delimited identifiers are object names enclosed within double quotation marks. Using them allows you to avoid certain restrictions on object names. Delimited identifiers can begin with nonalphabetic characters, including characters that would not otherwise be allowed. Delimited identifiers may even be Transact-SQL reserved words.

Because of this change, some SQL statements that worked in SQL Server versions previous to 10.0 may no longer work. For example, the following would **not** have generated an error message previous to SQL Server 10.0:

```
1> select au_name, "none", au_fname
2> from pubs2..authors
3> go
```

As of release 10.0, Error 207 would be raised for the column *"none"*. See below for how to correct this error.

#### Action

Check the spelling of the column name specified in the error message. If it is wrong, correct it in your SQL statement and run it again.

If this error message is being displayed because of delimited identifiers, either use single quotes instead of double quotes or turn delimited identifiers off. To turn delimited identifiers off, do the following:

```
1> use database_name
2> go
1> set quoted_identifier off
2> go
```

When delimited identifiers is turned on, delimiting strings with double quotes causes SQL Server to treat them as identifiers.

Additional Information

Refer to the *SQL Server Transact-SQL User's Guide* for information about delimited identifiers.

# Error 208

Severity Level

16

**Error Message Text** 

Release 10.0 and Later

```
%.*s not found. Specify owner.objectname or use
sp_help to check whether the object exists
(sp_help may produce lots of output).
```

## **Release 4.9.x and Earlier**

Invalid object name '%.\*s'.

## Explanation

This error occurs when SQL Server tries to use an object name that does not exist. If the object does exist, you may need to include the owner's name in the object name.

If the object is not owned by the user who is attempting to access it, and it is not owned by the database owner ("dbo"), all references to it must include the owner name. For example, if "user1" creates a table called *test*, all other users must prefix the table name with the owner name "user1" every time they refer to the object. The complete name in this example is *user1.test*. This requirement is based on SQL Server's naming convention for database objects:

database.owner.object\_name.column\_name

The default value for *database* is the current database and the default value for *owner* is the current user. Remember that the owner is part of the object name and it is therefore possible for two different users to have two different tables with the same *object\_name* in the same database, for example, *user1.test* and *user2.test*. Refer to the *Transact-SQL User's Guide* for more details on naming conventions.

Temporary tables reside in *tempdb* and are automatically dropped when the user process or SQL Server connection that created them is exited. Furthermore, users cannot share temporary tables, even if they are created by the "dbo."

Including the fully qualified object name in an application may complicate the maintenance of the application. For example, if all references to a table include the database name, changing the database name could become quite difficult.

## Action

To resolve this error, use one of the following sections, depending on how much you know about the object in question.

# If You Do Not Know Who Owns the Table

You can either use the sp\_help procedure to display the owner (if the procedure is executed with no parameters), or query the system catalog to determine the object's owner and type. For example, to determine the name and type of the object *table1*, use the following query:

```
1> select owner = user_name(uid), name, type
2> from sysobjects where name = "table1"
3> go
```

If no rows are returned by this query, the object may reside in a different database or may not exist at all.

If You Do Not Own the Object in Question

Avoid this error using either procedure:

• Include the owner's name in the object name. For example:

```
1> select * from user1.table1
2> go
```

• Have the "dbo" create the object. This allows any user to find the object without specifying an owner name.

If the Object Does Not Reside in the Database

The error can be avoided by:

- Moving to the correct database with the use database\_name command.
- Fully qualifying the object name with the database name. For example:

1> select \* from database1.user1.table1
2> go

The owner name is not needed if you own the object or if it is owned by the "dbo." For example:

```
1> select * from database1..table1
2> go
```

The appropriate permissions must also be set to allow access to this object. If these permissions are not provided, a 229 or 230 error results.

#### **Severity Level**

16

#### Error Message Text

```
Insert error: column name or number of supplied values does not match table definition.
```

#### Explanation

This error occurs when there is a mismatch between the information specified in the insert statement and the definition of the table.

Following are some examples of insert statements for the *titles* table (from the *pubs2* database) that cause this error (assuming that the *titles* table has nine columns):

• No column names are specified after the table name in an insert statement and there are two or more columns in the table.

```
1> insert titles values ('this is a test')
2> go
Msg 213, Level 16, State 4:
Line 1:
Insert error: column name or number of supplied
values does not match table definition.
```

• No values are specified in an insert statement.

```
1> insert titles (title_id, title) values ()
2> go
Msg 213, Level 16, State 3:
Line 1:
Insert error: column name or number of supplied
values does not match table definition.
```

#### Action

Check your insert statement to make sure you have specified column names and values for the columns.

If this error message is being displayed while an application is being run, refer to "Determining SQL Sent to Server" in the *Tools and Connectivity Troubleshooting Guide* to determine which insert statement is causing the problem.

#### **Severity Level**

11

#### **Error Message Text**

```
Cannot run query--referenced object (name %.*s) dropped during query optimization.
```

#### Explanation

This error occurs when SQL Server fails to build an execution plan for a stored procedure or an ad-hoc query because a referenced object was not found.

Some reasons for this error include:

- A problem exists within SQL Server.
- The value of the open objects configuration parameter is too low.

This is not a serious error as it does **not** indicate corruption. However, it may prevent you from running some stored procedures and ad-hoc queries.

#### Action

Using sp\_configure, review the value of the configuration parameter open objects and increase this value if feasible. Refer to "SQL Server Memory Issues" in the *SQL Server Troubleshooting Guide* for information about the values of configuration parameters.

This error usually means a problem has occurred with SQL Server. Call Sybase Technical Support for assistance.

#### **Severity Level**

16

#### **Error Message Text**

```
%s command not allowed within multi-statement transaction.
```

#### Explanation

SQL commands are grouped into the following categories:

- SQL commands that are not allowed in transactions at all.
- SQL commands, such as Data Definition Language (DDL) commands, that are allowed in transactions only if the required database option (ddl in tran) is set to TRUE.
- SQL commands that are allowed only if the transaction affects some other database. These commands include create table, drop table, and other commands that are run across databases to create or drop objects in another database when the database in which the objects are being created or dropped has the database option ddl in tran set to TRUE.

Error 226 occurs when SQL Server detects a command that is not allowed in a multi-statement transaction. A multi-statement transaction is a set of commands prefaced with the begin transaction command.

The following commands are never allowed in multi-statement transactions:

- alter database
- create database
- dbcc reindex, dbcc fix\_text
- disk init
- drop database
- dump database, dump transaction
- load database, load transaction
- reconfigure
- select into

- set transaction isolation level
- truncate table
- update statistics

The following commands are not normally allowed in multistatement transactions but you can use them if you use sp\_dboption to set ddl in tran to TRUE first:

- create default, create index, create procedure, create rule, create schema, create table, create trigger, create view
- drop default, drop index, drop procedure, drop rule, drop table, drop trigger, drop view
- grant
- revoke

#### Action

If the command is allowed in a multistatement transaction when ddl in tran is set to TRUE, set ddl in tran to TRUE before running the transaction. You can check the current setting of ddl in tran with sp\_helpdb.

If the command is never allowed in a multi-statement transaction, execute it outside the multi-statement transaction.

#### **Additional Information**

Some applications take SQL statements as input and run them for you. If the application uses begin and commit or rollback transaction to surround those statements, Error 226 may occur. Refer to the user guide for your application to determine if this is the case.

#### **Severity Level**

14

#### Error Message Text

```
%s permission denied on object %S_OBJID, database
%S_DBID, owner %.*s
```

#### Explanation

This error occurs when SQL Server attempts to access an object for which you do not have the appropriate permission.

Error 229 can occur during any transaction which involves objects, such as bcp, select, or update.

This error causes a rollback of the current transaction. Remaining commands in the batch are not executed and processing continues at the next batch.

#### Action

Check permission on the object named in the error message:

```
1> use database_name
2> go
1> sp_helprotect object_name
2> go
```

where *database\_name* is the name of the database in the error message and *object\_name* is the name of the object in the error message.

Grant the needed permission to the object (you either need to own the object or be the "sa" to do this). For example, to give Mary permission to insert into and delete from the *titles* table, type the following:

```
1> grant insert, delete
2> on titles
3> to mary
4> go
```

Additional Information

For detailed information about permissions, refer to the *System Administration Guide*.

#### **Severity Level**

16

#### **Error Message Text**

Arithmetic overflow error for type %s, value = %f.

#### Explanation

This error occurs when SQL Server detects an arithmetic overflow error during execution of a SQL statement. Arithmetic overflows occur when a variable has too few places to store the converted data. In particular, Error 232 is raised when an overflow occurs for a variable of datatype *float*.

Error 232 can occur during execution of the convert function. It can also occur during implicit conversion within SQL Server, either during a stored procedure run or during execution of a SQL statement which converts a *float* value to some other datatype.

#### Action

Determine which variable is causing the problem based on the datatype reported in the error message output and the SQL statement on which the error occurred. If the error occurs within a stored procedure, an additional error message will be displayed showing the name of the stored procedure and the line number where the error occurred.

Increase the size of the variable that is causing the error. If it is in a procedure, re-create the procedure.

If this error occurs on a Sybase-supplied system stored procedure, contact Sybase Technical Support.

Severity Level

16

### **Error Message Text**

#### Release 10.0 and Later

Ad-hoc updates to system catalogs not enabled. A user with System Security Officer (SSO) role must reconfigure system to allow this.

#### **Release 4.9.x and Earlier**

Ad-hoc updates to system catalogs not enabled. System Administrator must reconfigure system to allow this.

#### **Explanation**

System tables are all the Sybase-supplied tables in the *master* database and in each user database that begin with "sys" and have an ID value in the *sysobjects* table of less than or equal to 100. The system catalog is the collection of system tables.

Error 259 occurs when you try to directly modify a SQL Server system table without first using sp\_configure to set the allow updates configuration parameter. Setting this variable allows the System Administrator ("sa") to make changes to system tables.

#### ♦ WARNING!

Serious problems can result from an incorrectly modified system table! Make changes only if Sybase Technical Support or this manual instructs you to do so.

#### Action

Do not use any of the procedures in this section before reading all warnings and cautions! Incorrect alterations of a system table can result in database corruption and data loss. To successfully modify the system catalog as directed by Technical Support or this guide, use the following procedure:

1. As "sa," use sp\_configure to allow changes to the system catalog:

```
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
```

2. Make changes to the system catalog as directed by Technical Support or elsewhere in this guide. Always use begin transaction when changing a system table, to provide protection against errors which could corrupt your database.

#### ♦ WARNING!

Be sure to include the next step to protect your system.

3. Immediately following the changes, use sp\_configure to disallow changes to the system catalog (the normal state for SQL Server):

```
1> sp_configure "allow updates", 0
2> go
1> reconfigure
```

```
2> go
```

You must return SQL Server to its normal state, or anyone who can log on as "sa" can make changes to the system tables, or create stored procedures which do so.

4. If you update *sysusages*, *sysobjects*, *sydatabases*, or *sysdevices*, restart SQL Server. You need to do this because structures related to those tables are stored in cache and will not be updated until you restart SQL Server.

#### **Additional Information**

See the *System Administration Guide* or *SQL Server Reference Manual* for more information about (sp\_configure) allow updates.

### **Severity Level**

16

# **Error Message Text**

```
You can't run SELECT INTO in this database. Please check with the Database Owner.
```

### Explanation

This error occurs when you try to add rows to a table via the select into clause in a database that has the sp\_dboption option select into/bulkcopy parameter disabled.

If this error occurs while trying to select into a temporary table, or while running a Sybase-supplied stored procedure, it is likely that the select into/bulkcopy option is not enabled in *tempdb*. Some stored procedures, for example sp\_help and sp\_helpsort, trigger this error because they select into temporary tables (which all belong to *tempdb*) to get reports from the system tables, and to update them. If *tempdb* does not have the sp\_dboption option select into/bulkcopy enabled, Error 268 will occur.

#### ► Note

When you install SQL Server, the select into/bulkcopy option is enabled in *tempdb* and turned off in all other databases.

The sp\_dboption option select into/bulkcopy must be enabled to perform nonlogged operations. For example:

- Performing a select into a permanent (nontemporary) table.
- Performing a fast bulk copy with the bcp utility. Tables without triggers or indexes use the fast version of bcp by default, that is, they are not logged in order to save time.
- Executing the Transact-SQL command writetext (using the with log option of writetext causes it to be logged, so the select into/bulkcopy parameter is not needed).
- Executing the DB-Library routine dbwritetext.

#### ♦ WARNING!

You cannot dump the transaction log to a device after performing a nonlogged operation. Always perform a dump database after the nonlogged operation is complete, since performing nonlogged operations leaves changes recorded in the transaction log unrecoverable.

#### Action

Ask the System Administrator ("sa") or database owner ("dbo") to enable the sp\_dboption option select into/bulkcopy for all databases affected by the query (including *tempdb* if applicable). The "sa" or "dbo" can use this procedure:

1. Use the master database and change the option:

```
1> use master
2> go
1> sp_dboption database_name,
2> "select into/bulkcopy", true
3> go
1> use database_name
2> go
1> checkpoint
2> go
2. To verify that the change is active, use sp_helpdb:
```

```
1> sp_helpdb database_name
2> go
```

In the output, the phrase select into/bulkcopy should appear, indicating that the option is enabled.

#### ♦ WARNING!

Dump your database before disabling the select into/bulkcopy option. If you insert nonlogged data (via a nonlogged operation) into the database, you will not be able to dump the transaction log to a device, and will not be able to recover your data.

By default, the select into/bulkcopy option is disabled when a database is first created. To change this default, use the procedure above to allow the option in the *model* database.

# **Additional Information**

If a table has indexes or triggers, then bcp will not run in the fast mode, so you do not need to set the select into/bulkcopy option to run bcp.

You cannot dump the transaction log to a device after performing a minimally logged operation. Trying to dump the transaction log after a minimally logged operation causes error messages that instruct you to dump the database instead.

# **Compile Errors**

This section contains error messages pertaining to compile errors.

### **Severity Level**

16

# **Error Message Text**

```
Invalid operator for datatype op: %s type: %s.
```

# Explanation

This error occurs during SQL Server expression processing, when an operator in the query is applied to a datatype for which that operator is not valid. An operator is a logical or arithmetic expression such as "+" or "-".

For example:

```
1> select pub_id * pub_name from publishers
2> go
Msg 403, Level 16, State 1:
Line 1:
Invalid operator for datatype op: MULTIPLY type: CHAR.
```

# Action

Correct your query and run it again.

# **Additional Information**

Refer to the *SQL Server Reference Manual* and the *Transaction-SQL User's Guide* for information about the use of operators in queries.

#### **Severity Level**

19

#### **Error Message Text**

```
Too many ANDs or ORs in expression (limit %d per expression level). Try splitting query or limiting ANDs and ORs.
```

#### Explanation

This error occurs when you use more than 251 and or or expressions in a query.

This error also occurs if you exceed the limit of 251 values in a where in clause. For example:

```
1> select * from old_table
2> where old_column in (1,2,3...,253,254,...)
3> go
```

If you exceed the limit by a large margin, your query may cause a stack trace and Error 3626.

#### Action

Write the query so that the limit is not exceeded. Often, this requires that you split the query.

#### **Additional Information**

See "Search Conditions" in the *SQL Server Reference Manual* for more information.

#### **Severity Level**

20

#### Error Message Text

```
Too many table names or referential constraints in the query, maximum allowed table references is '%d'.
```

#### Explanation

During query processing, SQL Server checks for the existence of foreign keys and dependent foreign keys (a foreign key is a column or combination of columns whose values match the primary key). A range table entry is created for each reference check and foreign key constraint. *%d* in the error message is the maximum number of table references allowed (16). Error 428 is raised when this limit is reached.

#### Action

When setting up constraints on your tables, determine the maximum number of tables that might be touched by an update, insert, or delete statement.

To determine which constraints exist for a table, type:

```
1> use database_name
2> go
1> sp_helpconstraint table_name
2> go
```

If necessary, drop some of the constraints on the table:

1> alter table table\_name
2> drop constraint constraint\_name
3> go

#### **Additional Information**

See the *SQL Server Reference Manual* for information about constraints.

#### **Severity Level**

18

#### **Error Message Text**

```
Constraint id info in the resvar for check
constraints is missing. This is an internal error,
database = '%S_DBID', table = '%S_OBJID'.
```

#### Explanation

This error occurs during query processing when SQL Server fails to find an expected constraint. A "resvar" is a result variable and it is used internally to represent information in a column SQL Server is updating or returning to the user.

This error usually occurs as a result of a problem with SQL Server.

#### Action

Call Sybase Technical Support.

As a workaround, use alter table to drop the constraints on the table referenced in the error message. To determine which check constraints exist for the table, type:

```
1> use database_name
2> go
1> sp_helpconstraint table_name
2> go
```

where *database\_name* and *table\_name* are those named in the error message. The *defn* column displays "CHECK" for check constraints.

Then, drop each check constraint on that table:

```
1> alter table table_name
2> drop constraint constraint_name
3> go
```

#### Additional Information

See the *SQL Server Reference Manual* for information about constraints.

# **Query Processor Errors**

This section contains error messages pertaining to query processor errors.

#### **Severity Level**

16

#### **Error Message Text**

```
Updated or inserted row is bigger than maximum size (%d bytes) allowed for this table.
```

#### Explanation

This error occurs when you try to insert or update a row that is longer than the maximum length of 1962 characters. (On Stratus platforms, the row size can be up to 4010 characters because Stratus platforms have a larger page size than other platforms.)

Error 511 can be caused by database design errors (for example, a table designed with the potential for rows containing more than 1962 characters) or a SQL Server problem.

In SQL Server releases 10.0 and later, the following warning is given when you create a table that has the potential for rows containing more than 1962 characters (that is, the maximum length of all columns added up is greater than 1962 characters):

Msg 1708, Level 16, State 1: Warning: Row size could exceed row size limit, which is %d bytes.

In SQL Server releases previous to 10.0, no warning is given.

#### Action

If Error 511 is being caused by a table containing rows with more than 1962 characters, divide the tables into two or more tables so that no row length is more than 1962 characters (4010 characters for Stratus platforms).

If Error 511 occurs when you are running a system stored procedure, the cause may be a corrupted system stored procedure. In this case, rerun installmaster to rebuild the system stored procedures. Refer to the *System Administration Guide* for information about using installmaster to rebuild *sybsystemprocs*.

# ♦ WARNING!

If you have modified any of the system stored procedures, running installmaster will override those modifications.

If the 511 error occurring on your database does not appear to be caused by either of these situations, call Sybase Technical Support.

**Severity Level** 

16

### **Error Message Text**

```
Attempt to insert NULL value into column '%.*s',
table '%.*s'; column does not allow nulls. Update
fails.%S_EED
```

#### **Explanation**

Each column in a table has an explicit NULL or NOT NULL entry. NOT NULL means that some data entry is required. The default value for all columns in a database is set by the sp\_dboption option allow nulls by default. When the allow nulls by default option is set to TRUE, the default value of a column is NULL, meaning that null values are allowed in a column. You can override this setting on a column-bycolumn basis using alter table.

Error 515 occurs when a column has a NOT NULL default and you try to insert a NULL value into that column (for example, if a user does not enter a value for that column). The error message includes:

- The name of the affected column.
- The name of the affected table.
- The EED (extended error data array) includes the database name, owner, table name, and column name. This information is not displayed but you will see it in the message definition if you select Error 515 from *sysmessages*.

The following sample would cause a 515 error:

```
1> create table table1 (column1 varchar (5))
2> go
1> declare @c varchar(5)
2> insert into table1 values (@c)
3> go
Msg 515, Level 16, State 3:
Server 'SERVER_NAME', Line 2:
Attempt to insert NULL value into column
'column1', table 'test.dbo.table1'; column does
not allow nulls. Update fails.
```

#### Action

To determine whether a column has NULL or NOT NULL defined, enter commands like the following where "pubs2" is replaced by the database in which your table resides and "titleauthor" is replaced by the table in which your column resides:

```
1> use pubs2
           2> go
           1> sp_help titleauthor
           2> go
Name
          Owner Type
_____
titleauthor dbo
               user table
Data_located_on_segment
                      When_created
_____ ____
                      Oct 27 1994 10:09AM
default
Column_name Type Length Prec Scale Nulls Default_name Rule_name Identity
_____
au_idid11NULL0NULLNULL0title_idtid6NULLNULL0NULLNULL0au_ordtinyint1NULLNULL1NULL0royaltyperint4NULLNULL1NULLNULL0
```

The *Nulls* column indicates whether null values are allowed. A value of 0 for the column means nulls are not allowed and 1 means null values are allowed.

To change the default for a column from NOT NULL to NULL, enter a command like the following:

```
1> alter table table_name
2> replace column_name default null
3> go
```

To change the default value of **all** columns in a database from NOT NULL to NULL, enter a command like the following:

```
1> use master
2> go
1> sp_dboption database_name,
2> "allow nulls by default", true
3> go
```

where *database\_name* is the name of the database whose behavior you want to change.

# **Access Errors**

This section contains error messages pertaining to access errors.

### **Severity Level**

21

### **Error Message Text**

```
Descriptor for system table '%ld' in database '%d' not found in the descriptor hash table.
```

### Explanation

### Release 10.0 and Later

This error occurs when the SQL Server Threshold Manager could not open the *systhresholds* system table in a database because the descriptor for *systhresholds* was not found. SQL Server expects every database to have a *systhresholds* table. The failure to open *systhresholds* is probably due to corruption in the database named in the error message.

► Note

Error 601 was not raised in SQL Server releases 4.8 and 4.9.x.

**Release 4.2 and Earlier** 

This error occurs when the SQL Server could not open a system table in a database because the descriptor for that system table was not found. The failure to open the system table is probably due to corruption, either in memory or in the database itself.

# Action

1. Determine the name of the database from the database ID in the error message:

```
1> use master
2> go
1> select * from sysdatabases where dbid = ID
2> go
```

where *ID* is the database ID from the error message.

2. Determine whether the system table object ID displayed in the error message exists:

```
1> use database_name
2> go
1> select * from sysobjects where id = object_ID
2> go
```

where *database\_name* is the name from step 1 and *object\_ID* is the system table ID listed in the error message.

- 3. If the ID exists for the table displayed in the message, then the 601 error may be due to corruption in the memory structure used to hold the descriptor. Shutting down and restarting SQL Server should clear the problem. If this does not work, go on to step 4.
- 4. If the ID does not exist, there is probably corruption in your database. Do the following:
  - 1. Run dbcc checkalloc, dbcc checkcatalog, and dbcc checkdb for the database.
  - 2. Determine whether hardware problems exist by checking your operating system error log.
  - 3. Call Sybase Technical Support. They may be able to help you recover your database, but you will probably have to recover your database from backups.

#### **Additional Information**

Before contacting Technical Support, be prepared to provide:

- SQL Server release and EBF Rollup level
- Complete text of all error messages
- SQL Server error log output
- Operating system error log output
- dbcc checkalloc, dbcc checkcatalog, and dbcc checkdb output for the database listed in the error message

**Severity Level** 

19

### **Error Message Text**

There are not enough system session descriptors available to run this query. The maximum number available to a process is %d. Split query and rerun.

#### Explanation

This error occurs when SQL Server runs out of **system** session descriptors when running a query. Error 603 can occur on commands which create, drop, and update objects, as these commands often have many dependencies on system tables which SQL Server must handle to complete the command.

Session descriptors are the data structures used to manage access to SQL Server objects. In SQL Server 10.0.1 and earlier versions, session descriptors came in two forms: **user** session descriptors (including worktable session descriptors) and **system** session descriptors. SQL Server 10.0.2 included a new type, **worktable** session descriptors (separate from user session descriptors). All three are described below.

Session descriptors are used in the following situations:

- When your query is compiled or executed, a **user** session descriptor is reserved and assigned to each table involved in the query. Currently, a maximum of 16 user session descriptors are possible, limiting the number of tables involved in a query to that number. If a single table is referenced more than once in the from clause (a "self join"), a user session descriptor is reserved for each reference. User session descriptors correspond to the open objects configuration parameter.
- **System** session descriptors are used for system tables that are not named in the query. For example, the following query:

```
select x from t where y = 3
```

prompts SQL Server to open *sysobjects* and *syscolumns* to look up the table *t* and its columns *x* and *y*. A maximum of six system session descriptors are available.

• A worktable session descriptor is used to manage references to worktables. A worktable is required for order by, group by, distinct, and some other operators. Also, the query optimizer may choose a plan involving reformatting which copies rows from a table into a worktable, creates an index, then performs joins between the worktable and other tables in the query. There are 12 worktable session descriptors that can be used at any one time.

#### ► Note

In SQL Server releases previous to 10.0.2, descriptors for worktables were user session descriptors.

You cannot change the number of session descriptors available to a query.

#### Action

When you get Error 603, you do indeed have to simplify your query or command. To recover from this error, split up your query or simplify the command and run it again.

For example, if you are entering a **drop procedure** command and that procedure contains other procedures or triggers, you can simplify your command by dropping the other procedures or triggers manually before running the **drop procedure** command.

#### **Severity Level**

21

#### **Error Message Text**

```
Attempt to fetch logical page %ld in database
'%.*s' belongs to object id '%ld', not to object
'%.*s'.
```

#### Explanation

This serious error occurs when SQL Server discovers page allocation corruption. SQL Server tries to access a particular object but discovers a page in the page chain for that object whose object ID is different than that of the object being accessed. There is probably either a damaged page chain or an invalid entry in the system tables for that object.

Usually this error is detected after the corruption has been written to the database on disk (hard 605), but it can also occur entirely in cache without the damage ever being written to disk (transient 605), and is not associated with data corruption. See the next section for more information about the different kinds of 605 errors and their causes.

```
► Note
```

When an error is transient, it means it is reported even though no error condition exists or it exists only in memory. Generally, a transient error is cleared by a reboot of SQL Server, whereas a hard error does not go away with a SQL Server reboot.

#### How Much Corruption?

Assuming the error is not a transient 605, corruption usually exists in the second object specified in the 605 error text. If the dbcc checkdb and dbcc checkalloc commands do not report additional errors, the first object mentioned is not corrupt. Because the 605 error can mask the existence of other errors, you must run the dbcc checks to determine the extent of the damage. Instead of listing two object names, this error might contain other information:

- The error displays a number greater than zero. An attempt was made to refer to an object ID that does not exist in the system table *sysobjects*.
- If the error states that a page belongs to object "ALLOCATION," some of the allocation structures used by the database may be corrupted.

### Hardware and Software Causes for 605 Errors

Hardware Cause	Additional Information	Transient or Hard
Overlapping partitions on disk (UNIX only). <sup>a</sup>	Often this occurs when SYBASE is installed on an incorrect partition (a, b, and so on).	Hard
Problem with device driver		Hard
Problem with controller		Hard
Bad disk	"Data retry" messages in /var/adm/messages (UNIX).	Hard

#### Table 2-1: Hardware causes of Error 605

a. For more information about overlapping partitions, refer to "Correct Use of Raw Partitions" in the *SQL Server Troubleshooting Guide*.

#### Table 2-2: Software causes of Error 605

Software Causes	Additional Information	Transient or Hard
Loading a <i>master</i> database on a master device that does not have rows in <i>sysusages</i> for <i>dbid</i> 1,2,3 identical to the device it was dumped from.		Hard
Loading a <i>master</i> database that has <i>syscharsets</i> in a location that is different from the default.	Caused by changing SQL Server sort order after operating it for some time.	Hard

Software Causes	Additional Information	Transient or Hard
Operating System notifies SQL Server that an I/O operation is complete when it is not.	SQL Server connection often dies.	Transient
Any time buffer information about a page is different from the disk information about the page (and the disk has the correct information).	Running <b>dbcc checkdb</b> and <b>dbcc checkalloc</b> should produce no more 605 errors.	
	Corruption is not on disk. First, checkpoint the other databases. Then, perform a shutdownwith nowait and restart SQL Server.	

#### Table 2-2: Software causes of Error 605 (continued)

#### Action

- 1. Run the dbcc checktable command on the second object specified in the error message.
- 2. To help determine the full extent of the corruption, run the dbcc checkdb and dbcc checkalloc commands as soon as feasible.
- 3. Check the SQL Server error log for other errors which often accompany a 605 error. If the 605 error is not transient, the problem is severe and you will probably need to restore from known clean backups.
- 4. Either of the following situations may also apply to your SQL Server:
  - If this error occurs after rebuilding the master device or reconfiguring the configuration block to the defaults, then the sort order on the new device may not match the sort order in the dump. If the sort orders do not match, call Technical Support for further assistance.
  - To check if the error occurred as the result of hardware failure, examine your operating system error log and correct hardware problems.

If you have dial-in facilities, Technical Support may be able to help you recover from this error without having to restore from backups. However, this recovery procedure often results in a loss of data, and therefore is often unacceptable, especially to production systems.

# **Additional Information**

Check the operating system error log for hardware failures.

Before contacting Technical Support, be prepared to fax or provide:

- SQL Server release and EBF Rollup level
- Complete text of all error messages
- SQL Server error log output
- Operating system error log output
- dbcc checkdb, dbcc checkalloc, and dbcc page output

# **Severity Level**

21

# **Error Message Text**

```
Attempt made to end a transaction that is idle or in the middle of an update.
```

# Explanation

This error occurs when a termination request event occurs during the processing of a transaction. Error 611 can be caused by:

- Massive processing using group by or order by
- A stored procedure that references other stored procedures (and is in the midst of calling those procedures) is dropped
- A SQL Server problem

# Action

To determine what might have caused the above error message, check for the following:

- **dbcancel** calls from within DB-Library exit processing routines that fail to test all error conditions
- Users typing Ctrl-c during heavy use of temporary tables

Error 611 may produce a stack trace that gives a portion of the SQL that caused the error. Obtain the entire stack trace, along with any errors that might have occurred prior to or following the 611 error, to determine the internal state of the process at the time of the failure.

#### **Severity Level**

21

#### Error Message Text

```
A row on page %ld was accessed that has an illegal length of %d in database '%.*s'.
```

#### Explanation

This error occurs when SQL Server accesses a data or index row whose length is smaller than the minimum row size or greater than the maximum row size.

The minimum length of a row for each object is stored in the *minlen* column of *sysindexes* and in each data or index page header. The maximum size allowed for a data row or index row is 1962 bytes. (On Stratus platforms, the row size can be up to 4010 bytes because Stratus platforms have a larger page size than other platforms.)

This error can occur under the following conditions:

- During normal processing, when SQL Server tries to access the row specified by the error message.
- During database recovery (database recovery occurs during SQL Server start-up or when a load database or load transaction command is processed).

Error 614 can be caused by data corruption during normal processing (for example, an operating system panic occurs, causing interruption in disk writes when using UNIX files for SYBASE database devices). This may be due to a problem with SQL Server, the operating system, or hardware.

#### Action

Error 614 is usually the result of a more serious underlying problem, and recovering from this error depends on when the error occurred. Determine whether the error occurred during normal processing or during database recovery, then follow the appropriate set of instructions in this section.

# If the Error Occurred During Normal Processing

- 1. Use the procedure in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond to the page number from the error message text.
- 2. If the object encountering the error is **not** a system table (a system table's object ID is less than 100), continue with step 3.

If the object encountering the error is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption, but it may be necessary to restore the database from clean backups.

3. For user tables, if the index ID is 0, continue with step 4.

If the index ID is **not** 0, translate it into an index name:

```
1> use database_name
2> go
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the **sp\_helpindex** procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

- 4. If the index ID is 0, do one of the following:
  - Restore the database from clean backups.
  - Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

#### ♦ WARNING!

Some data might be lost on this page if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

Before dumping your database, make sure it works correctly. Run the following commands prior to each dump:

- 1. dbcc checkdb.
- 2. dbcc checkalloc or dbcc checkalloc with the fix option. (See "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about how to run these commands in multi-user mode and how to prevent spurious allocation errors from dbcc commands.)

#### If the Error Occurred During Database Recovery

When this error occurs during recovery, the database is marked suspect and is not accessible. Usually, you must load the database from backup. To do this, follow the instructions below:

- 1. Drop the database. If the drop fails, follow the instructions in "How to Drop a Database When drop database Fails" in the *SQL Server Troubleshooting Guide*.
- 2. Create a database for load. Make sure the database you create has sizes as least as large as those in *sysusages* for the original database (and that all other *sysusages* values match the original values). See the entry for create database for load in the *SQL Server Reference Manual* for releases 4.8 and later, or the entry for create database for releases earlier than 4.8.

#### ► Note

In SQL Server releases prior to 10.0, the SQL Server Reference Manual was called the Commands Reference Manual.

3. Load the database from backup. (See load database in the *SQL Server Reference Manual*.)

If loading from backups is not feasible, call Sybase Technical Support. Have the following documentation ready:

- SQL Server error log
- Text of all error messages
- select \* from master..sysusages output

Refer to the *System Administration Guide* for information about create database for load and load database.

# Preventing This Error on Recovery

To prevent Error 614 occurring on recovery, checkpoint each database that is being used before shutting down SQL Server.

### **Additional Information**

See the *System Administration Guide* or *SQL Server Reference Manual* for complete information about how to safely create, dump, load, and re-create databases.

#### **Severity Level**

21

#### **Error Message Text**

```
Attempt to retrieve row from page via RID failed because logical page %ld is not a data page. %S_RID. %S_PAGE.
```

#### Explanation

This error occurs when SQL Server tries to retrieve a row from a data page by specifying the row ID (RID) and the retrieval fails because the requested page was not a data page. This error is usually caused by a SQL Server problem.

After loading a database on top of an existing database, Error 623 may appear during recovery or while doing a checkpoint.

This error may or may not be transient (transient errors disappear after SQL Server is restarted). In the worst case, Error 623 may indicate corruption in your database. Perform the action below as soon as possible to determine how serious the error is and to avoid further corruption if corruption has occurred.

#### Action

Determine whether this error is transient by performing the following steps:

- 1. See "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* for instructions on how to use the dbcc page command to determine the table involved. Save this information and use it in the appropriate section below.
- 2. Using the object ID found in step 1, issue the following query to determine the table name:

```
1> select object_name(object_ID)
2> go
```

3. Using the table name displayed in the output from step 2, issue the following query to determine whether corruption has occurred:

```
1> dbcc checktable(table_name)
2> go
```

If no errors occur when you run dbcc checktable, the 623 error was probably transient. Shut down and restart SQL Server.

## If the Error is Not Transient

If errors **do** occur when you run **dbcc checktable**, refer to the writeups for those errors in this manual for instructions on how to recover from them or restore your database from a clean backup. To restore from backup, follow the instructions below:

- 1. Drop the database. If the drop fails, follow the instructions in "How to Drop a Database When drop database Fails" in the *SQL Server Troubleshooting Guide*.
- 2. Create a database for load. Make sure the database you create has sizes as least as large as those in *sysusages* for the original database and that all other *sysusages* values match the original values. See the entry for create database for load in the *SQL Server Reference Manual* for releases 4.8 and later, or the entry for create database for releases for releases earlier than 4.8.
- 3. Load the database from backup. (See load database in the *SQL Server Reference Manual*.)

If loading from backups is not feasible, call Sybase Technical Support. Have the following information ready:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all error messages
- select \* from master..sysusages output

## Additional Information

See the *System Administration Guide* or *SQL Server Reference Manual* for complete information about how to safely create, dump, load, and re-create databases.

#### **Severity Level**

24

### **Error Message Text**

```
Attempt to retrieve row from page via RID failed because the requested RID has a higher number than the last RID on the page. %S_RID.%S_PAGE.
```

#### Explanation

This error occurs when SQL Server tries to retrieve a row from a data page by specifying the row ID (RID), but the retrieval fails because the requested row ID was a higher number than the last row ID on the page. This can happen:

- During normal processing, if the leaf page of a corrupt nonclustered index points to an incorrect or nonexistent row ID on a data page.
- During database recovery at SQL Server start-up (database recovery also occurs when a load database command is processed). Recovery fails because the data structure which contains the last checkpoint record in the transaction log is pointing to an incorrect or nonexistent record in the log.

#### Action

Recovery from this error depends on when the error occurred. Determine whether the error occurred during normal processing or during database recovery and then follow the appropriate instructions in this section:

- "If the Error Occurred During Normal Processing" on page 2-47
- "Index Unknown, Query Known" on page 2-48
- "Index and Query Both Unknown" on page 2-49
- "If the Error Occurred During Database Recovery" on page 2-49

#### If the Error Occurred During Normal Processing

See "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* for instructions about how to use the dbcc page command to determine the table involved. Save this information and use it in the appropriate section below. The specific action you take depends on whether or not you know which index or query caused the problem. In general, dropping and re-creating the index should resolve the index corruption.

Index Unknown, Query Known

If you do not know which index is causing the problem, but you do know which query encounters the problem, use this section to correct it. If you do not know the index or query, go to "Index and Query Both Unknown" on page 2-49.

1. Determine which index should be dropped by reading the showplan output for the query that encounters the error and determining which nonclustered index the query is using to access the table in question.

Turn on showplan and use the no execute mode:

1> set showplan on 2> go 1> set noexec on 2> go

Enter the query that was causing the problem. For example:

1> select title from titles where title > "Cooking"
2> go
STEP 1
The type of query is SELECT
FROM TABLE
titles
Nested iteration
Index: titleind

Turn showplan and the no execute mode off again:

1> set noexec off
2> go
1> set showplan off
2> go

2. If the index identified in step 1 is on a system table (object ID is less than 100), refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on repairing the system table index. Otherwise, drop and re-create the index identified in step 1 (in this example "titleind").

2-48

#### Index and Query Both Unknown

If you do not know either the index or the query, rebuild all nonclustered indexes on the table using the instructions in this section.

1. Examine the dbcc page output you obtained earlier.

If the object is a system table (the object ID is less than 100), refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index. Then go to step 3.

- 2. For a user table, use sp\_helpindex to list all indexes on the table, and then rebuild all the nonclustered indexes using one of two methods:
  - Drop and re-create each nonclustered index on the table.

Or:

- If a clustered index also exists on the table, drop and re-create it; this causes all nonclustered indexes to be rebuilt. If your table is large, you may not have the space to do this (a rough rule of thumb is 120 percent to 150 percent of your table size must be available).
- 3. Run dbcc checktable on the table to confirm that the problem has been resolved.

#### If the Error Occurred During Database Recovery

When this error occurs during recovery, the database is marked suspect and is not accessible. Usually, you must load the database from backup. Follow the instructions below, depending on the release level of your SQL Server (if you have dial-in facilities, Sybase Technical Support may be able to correct the problem so that recovery of this database can proceed):

- 1. Drop the suspect database using the instructions in "How to Drop a Database When drop database Fails" in the *SQL Server Troubleshooting Guide*.
- 2. Create a database for load. Make sure the database you create has sizes as least as large as those in *sysusages* for the original database (and that all other *sysusages* values match the original values). See the entry for create database for load in the *SQL Server Reference Manual* for releases 4.8 and later, or the entry for create database for releases for releases earlier than 4.8.

► Note

In SQL Server releases prior to 10.0, the *SQL Server Reference Manual* was called the *Commands Reference Manual*.

- 3. Load the database from backup (see the entry for load database in the *SQL Server Reference Manual*).
- 4. If your SQL Server is release 10.0 or later, skip this step.

In step 2, after the database has loaded, user segment information exists in *syssegments* but not in *sysusages*. If your database includes user segments, use sp\_extendsegment for all user-defined segments in the database so they will be known to *sysusages*. Refer to the *System Administration Guide* and the *SQL Server Reference Manual* for information about sp\_extendsegment.

#### **Severity Level**

21

#### **Error Message Text**

```
Could not retrieve row from logical page %ld via
RID because the entry in the offset table for that
RID is less than or equal to 0.
```

#### **Explanation**

A SQL Server attempt to retrieve a row from a data page by specifying the row ID (RID) failed because the requested row ID had an illegal value.

Every row in a database has a unique row ID. The two parts of a row ID are a logical page number and a row number. The row number part of row IDs are stored at the end of every data page and indicate where a certain row is located on that page.

Routines that use row IDs include the dbcc checkdb, dbcc checktable, and dbcc checkcatalog commands. They verify the consistency of indexes and use the row IDs stored in nonclustered indexes to quickly access the data rows. If a row number has an illegal value (smaller than or equal to zero), Error 625 is reported.

This error can occur under the following conditions:

- The data page of a nonclustered index is corrupt
- Hardware failure

#### Action

Examine your operating system error log file and the SQL Server error log to determine if hardware errors may have corrupted your database devices.

See "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* for instructions about how to use the dbcc page command to determine the table and the index involved. Save this information and use it in the appropriate section that follows.

The specific action you take depends on whether or not you know which index or query caused the problem. In general, dropping and re-creating the index will resolve the index corruption. Index Unknown, Query Known

If you do not know which index is causing the problem, but you do know which query encounters the problem, use this section to correct it. If you do not know the index or query, use the next section.

1. Determine which index should be dropped by reading the showplan output for the query that encounters the error and determining which nonclustered index the query is using to access the table in question.

Turn on showplan and use the no execute mode:

1> set showplan on 2> go 1> set noexec on 2> go

Enter the query that was causing the problem. For example:

```
1> select title from titles where title > "Cooking"
2> go
STEP 1
```

```
The type of query is SELECT
FROM TABLE
titles
Nested iteration
Index: titleind
```

Turn showplan and the no execute mode off again:

```
1> set noexec off
2> go
1> set showplan off
2> go
```

- 2. If the index identified in step 1 is on a system table (object ID is less than 100), refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.
- 3. Otherwise, drop and re-create the index identified in step 1 (in this example "titleind").

If this did not clear the error, create a clustered index on the table or drop and re-create the existing clustered index. Note that creating a clustered index requires an amount of space equal to 120 percent to 150 percent of the size of the table where you want to create the index.

#### **Both Index and Query Unknown**

If you do not know either the index or the query, rebuild all nonclustered indexes on the table using the instructions in this section.

- 1. Look at the dbcc page output you obtained earlier.
- 2. If the object is a system table (object ID is less than 100), refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on repairing the system table index. Then go to step 4.
- 3. For a user table (object ID is 100 or greater), use sp\_helpindex to list all indexes on the table, and then rebuild all the nonclustered indexes using one of two methods:
  - Drop and re-create each nonclustered index on the table.

Or:

- If a clustered index also exists on the table, drop and re-create it, which causes all nonclustered indexes to be automatically rebuilt. If your table is large, you may not have the space to do this (a rough rule of thumb is 120 percent to 150 percent of your table size must be available).

If the above methods did not clear the error, create a clustered index on the table (if space is available). You may then drop the clustered index.

4. Run dbcc checktable on the table to confirm that the problem has been resolved.

#### Additional Information

For more information, see the entry for create database for load in the *SQL Server Reference Manual* for releases 4.8 and above or the create database entry for releases below 4.8.

If you need to call Technical Support for assistance, have the following documentation ready:

- SQL Server version and EBF Rollup level
- SQL Server error log
- Text of all error messages
- select \* from master..sysusages output

#### **Severity Level**

21

#### **Error Message Text**

```
Fatal attempt to delete clustered index entry for page %ld - index row contains page %ld.
```

#### Explanation

This error occurs when SQL Server fails to delete a clustered index entry because the index entry did not point to the expected page.

In the error message text, the first page number refers to the data page and the second page number refers to the node-level index page that points to the data page.

Error 629 can occur when you attempt to delete a row (for example, when a table that has a clustered index is dropped, the row in *sysindexes* is deleted).

The error is caused by data corruption that occurred during SQL Server processing (for example, an operating system panic occurs, causing interruption in disk writes when using UNIX files for SYBASE database devices). This may be due to a problem with SQL Server, the operating system, or hardware.

#### Action

- 1. Use the procedure in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond to the first page number in the error message text.
- 2. If the object encountering the error is **not** a system table (a system table's object ID is less than 100), continue with step 3.

If the object with the error is a system table, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index. Then go to step 4.

3. Translate the index ID into an index name:

```
1> use database_name
2> go
```

```
1> select name from sysindexes
2> where id = object_ID and indid = 1
3> go
```

To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Drop the index.

Re-create the index. This clears the corruption in most cases.

4. Run dbcc checktable on the table to verify that the corruption is gone. If corruption still exists, call Sybase Technical Support.

#### Additional Information

See the *System Administration Guide* and *SQL Server Reference Manual* for information about dropping and re-creating indexes.

#### **Severity Level**

21

### **Error Message Text**

```
The length of %d passed to delete row routine for
the row at offset %d is incorrect on the following
page: %S_PAGE.
```

#### Explanation

This error occurs when SQL Server attempts to delete a row (via a direct delete or inherently through updating) from an index or data page by specifying the row offset and the row length, and the action fails because the specified values of the offset or row length did not match the actual values.

Error 631 can happen under the following conditions:

- During normal processing, when SQL Server tries to delete the row specified by the error message.
- During database recovery (database recovery occurs during SQL Server start-up or when a load database or load transaction command is processed).

Some potential causes of Error 631 are:

- Data corruption during normal processing (for example, an operating system panic occurs, causing interruption in disk writes when using UNIX files for SYBASE database devices). This may be due to a problem with SQL Server, the operating system, or hardware.
- Hardware failure during loading or dumping

#### Action

Error 631 is probably the result of a more serious underlying problem, and recovering from this error depends on when the error occurred. Follow the instructions in this section, selecting the correct set depending on whether the error occurred during normal processing or during database recovery.

2-56

### If the Error Occurred During Normal Processing

- 1. Use the procedure in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond to the page number from the error message text.
- 2. If the object encountering the error is **not** a system table (a system table's object ID is less than 100), continue with step 3.

If the object with the error is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption but you may have to restore from clean backups.

3. For user tables, if the index ID is 0 or 255, continue with step 4.

If the index ID is **not** 0 or 255, translate it into an index name:

```
1> use database_name
2> go
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

4. If the index ID is 255, delete the bad data row.

If the index ID is 0, do one of the following:

- Restore the database from clean backups.
- Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

## ♦ WARNING!

Some data might be lost on this page if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

Before dumping your database, make sure it works correctly. Run the following commands prior to each dump:

- 1. dbcc checkdb.
- 2. dbcc checkalloc or dbcc checkalloc with the fix option. (See "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about how to run these commands in multi-user mode and how to prevent spurious allocation errors from dbcc commands.)

### If the Error Occurred During Database Recovery

When this error occurs during recovery, the database is marked suspect and is not accessible. Usually, you must load the database from backup. To do this, follow the instructions below:

- 1. Drop the database. If the drop fails, follow the instructions in "How to Drop a Database When drop database Fails" in the *SQL Server Troubleshooting Guide*.
- 2. Create a database for load. Make sure the database you create has sizes as least as large as those in *sysusages* for the original database (and that all other *sysusages* values match the original values). See the entry for create database for load in the *SQL Server Reference Manual* for releases 4.8 and later, or the entry for create database for releases for releases earlier than 4.8.

#### ► Note

In SQL Server releases prior to 10.0, the *SQL Server Reference Manual* was called the *Commands Reference Manual*.

3. Load the database from backup. (See the load database entry in the *SQL Server Reference Manual*.)

If loading from backups is not feasible, call Sybase Technical Support. Have the following documentation ready:

- SQL Server error log
- Text of all error messages
- select \* from master..sysusages output

Refer to the *System Administration Guide* for information about create database for load and load database.

## Preventing This Error on Recovery

One of the following items may help to prevent Error 631 from occurring on recovery:

• Use shutdown to shut down SQL Server instead of shutdown with nowait after a period of heavy update activity in your databases.

Or:

• checkpoint each database that is being used before shutting down SQL Server.

Examine both your operating system error log and the SQL Server error log to determine if hardware errors may have affected your database devices. Look for any kernel messages reporting I/O errors and check the hardware error log or diagnostics utilities for I/O errors.

#### **Additional Information**

See the *System Administration Guide* or *SQL Server Reference Manual* for complete information about how to safely create, dump, load, and re-create databases.

## **Severity Level**

20

## **Error Message Text**

```
Memmove() was called with a length of d - maximum allowed length is 2048.
```

### Explanation

This error occurs when SQL Server is moving data for the result set of a query or command and the size of the data SQL Server moves exceeds 2048 bytes. Error 632 is caused by a SQL Server problem or data corruption.

### Action

Run dbcc checkdb on the database that was being accessed when the error occurred and then call Sybase Technical Support.

### **Additional Information**

Before contacting Technical Support, be prepared to provide:

- SQL Server release and EBF Rollup level
- Complete text of all error messages
- SQL Server error log output
- Operating system error log output
- dbcc checkdb output for the database being accessed when the error occurred

#### Severity Level

21

#### **Error Message Text**

#### Release 10.0.2 and Later

Index row entry for index id %d of table '%.\*s' in database '%.\*s' is missing. Drop and re-create the index. (index page %ld, row %d, data page %ld)

#### Release 10.0.1 and Earlier

The non\_clustered leaf row entry for page %ld row %d was not found in index page %ld indexid %d database '%.\*s'.

#### Explanation

This error occurs when the nonclustered index indicated by "indexid" is corrupt. This corruption is detected when a process tries to delete a nonexistent row.

#### Action

To recover from this error, use the following steps to drop and re-create the index.

- Record the value of "index page" and "index id" specified in the 644 error text. Then follow the instructions in "How to Find an Object Name from a Page Number" in the SQL Server Troubleshooting Guide to identify which table and index correspond to the index page number. Also note the object ID.
- 2. If the object with the error is a system table (its object ID is less than 100), refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index. Then go to step 4.
- 3. If the object ID is greater than 100, drop and re-create the index, using the table name and the index name obtained in the step 2. This clears the corruption in most cases.
- 4. To verify that all problems have been resolved on this table, run dbcc checktable and dbcc tablealloc on the affected table. If you have

a pre-4.9.1 SQL Server, dbcc tablealloc is not available, and you should run dbcc checkalloc instead.

## **Additional Information**

If problems persist, this procedure may not be sufficient to clean up the index corruption, and you should contact Sybase Technical Support. Have the output from the dbcc checkalloc or dbcc tablealloc and dbcc checkdb commands available for their review.

## **Memory Manager Errors**

This section contains error messages pertaining to the SQL Server Memory Manager.

#### **Severity Level**

19

#### **Error Message Text**

#### Release 10.0.2 and Later

There is not enough procedure cache to run this procedure, trigger, or SQL batch. Retry later, or ask your SA to reconfigure SQL Server with more procedure cache.

#### Release 10.0.1 and Earlier

There is insufficient system memory to run this query.

#### **Explanation**

This error occurs when there is not enough procedure cache available to execute a stored procedure, trigger, or batch of one or more SQL statements. This can happen when loading a query plan into procedure cache or when resolving or compiling a procedure, trigger or batch. See "Procedure Re-resolution" in the *SQL Server Troubleshooting Guide* for a description of resolution.

The amount of available procedure cache is based on SQL Server use, so this error may occur intermittently as procedure cache is used and released. Also, one or more procedures can repeatedly fail with this error while others complete normally. This is because query plans vary greatly in size. Therefore, while there may not be sufficient space to load a large query plan, several smaller plans may fit.

#### Action

Correct this error by increasing the size of procedure cache. You can increase the amount of total procedure cache in three ways:

- Use the memory configuration parameter to increase the total amount of memory requested by SQL Server.
- Use the procedure cache configuration parameter to increase the percentage of total cache space dedicated to procedure cache.
- Decrease the amount of memory required for other resources, for example, unneeded user connections.

Increasing the memory configuration parameter for SQL Server is the most straightforward method. It increases both the procedure and data caches, although it can waste memory.

► Note

On some operating systems, especially VMS and most UNIX systems, be sure to make any necessary operating system memory resource adjustments. For example, verify the current kernel value for the maximum size of a shared memory segment, usually *SHMMAX*, and make sure you have adequate memory and swap space on the system for the additional memory configured for the Server. See your operating system documentation or the *SYBASE SQL Server Installation Guide* or *System Administration Guide Supplement* for more information.

If your memory resources are limited, you can increase the procedure cache configuration parameter for the SQL Server without changing the value of the memory configuration parameter. This shifts space from the data cache to the procedure cache and may result in performance degradation or other problems if not enough data cache remains.

You can also increase the size of the procedure cache without substantially changing the amount of data cache. This requires some combination of increasing both the **memory** and **procedure cache** configuration parameters in such a way that the majority of the new memory goes to procedure cache. Although this method requires more planning, it allows you to control where the additional memory goes. For more information refer to "SQL Server Memory Issues" in the *SQL Server Troubleshooting Guide*.

#### **Severity Level**

20

#### **Error Message Text**

```
Memory request for %d bytes exceeds the size of single page of %d bytes.
```

#### Explanation

This error occurs when you exceed any of the following limits:

• 128 search conditions or join operations in a SQL statement.

A search condition sets the conditions in a where or having clause. For more information about and examples of search conditions, see the *SQL Server Reference Manual* entry for search conditions.

A join operator compares two or more tables or views. For more information about and examples of join operators, see the *SQL Server Reference Manual* entry for join.

- 128 columns in a Data Workbench update or delete table. For more information about the troubleshooting procedure for Error 702 in the Data Workbench environment, see the *Tools and Connectivity Troubleshooting Guide*.
- The total width of the columns in a temporary table during a union exceeds the allowed row width of 1962 bytes (4010 bytes for Stratus platforms).

After displaying Error 702, SQL Server terminates the current process.

#### Action

If your query exceeds the limit of 128 search conditions or join operations, rewrite the query so that the limit of 128 statements is not exceeded.

If the total width of the columns in a temporary table during a union exceeds the allowed row width of 1962 bytes (4010 for Stratus), rewrite your query so that the limit of 1962 bytes is not exceeded.

If this error occurred while in a Data Workbench session, see the *Tools* and *Connectivity Troubleshooting Guide*.

## **Additional Information**

For more information, see joins and search conditions in the *SQL Server Reference Manual*.

**Severity Level** 

17

#### **Error Message Text**

#### Release 10.0.2 and Later

You cannot run this procedure, trigger, or SQL batch because it requires more than %ld pages of memory. Break it up into shorter queries, if possible.

#### Release 10.0.1 and Earlier

Memory request failed because more than %d pages are required to run the query in its present form. The query should be broken up into shorter queries if possible.

#### **Explanation**

This error occurs when a stored procedure or trigger cannot be executed because it requires more memory than is allowed for execution. Prior to SQL Server release 10.0, the maximum amount of memory allowed was 64 pages, regardless of the page size used by SQL Server.

► Note

A page is 2K, except on Stratus, where pages are 4K.

Two different stages of execution can trigger this error: "resolution," in which the query tree is built, or "compilation," in which the query plan is generated. If a query has been executed successfully but later fails with this error, it means that the query tree used to be less than 64 pages but has since grown. If a query encounters this error every time it is executed, then it is too complex to be executed in its present form without exceeding the memory limit. See "Storing and Executing Procedures" in the *SQL Server Troubleshooting Guide* for more information about query trees and query plans.

#### Action

► Note

As of SQL Server release 10.0, this error has the same meaning as Error 701. Refer to "Error 701" on page 2-64 for information about how to handle this error on 10.0 and later releases.

If the procedure or trigger causes this error every time you try to execute it, divide it into smaller pieces. This division process varies greatly with the type of SQL statement.

If the object has successfully executed before, the 703 error probably occurred because the query tree grew beyond the 64 page limit. This growth occurs each time the query tree is re-resolved. Once you have encountered the 703 error in this situation, you can drop and re-create the procedure or trigger in order to shrink the query tree to a legal size, or you can break the object up into smaller modules. Restarting SQL Server has no effect on the size of the query tree because query trees are stored on disk.

If this error occurs frequently on an object, you can either periodically drop and re-create the object as part of regular database maintenance, or break it into smaller modules.

#### **Severity Level**

20

#### **Error Message Text**

```
Process %d tried to remove PROC_HDR 0x\%lx that it does not hold in Pss.
```

#### Explanation

The Memory Manager allocates, deallocates, and manages memory for SQL Server. It manages an array of structures in the process header, each of which represents a physical page of memory in procedure cache and the current byte allocation of that page.

As part of its memory deallocation process, SQL Server stops tracking process headers when they are no longer needed and attempts to remove them from procedure cache. When SQL Server attempts to deallocate a process header that it finds but does not own, Error 706 occurs.

Error 706 is caused by memory corruption or a SQL Server problem.

#### Action

Since process headers are stored in procedure cache, restarting SQL Server should clear Error 706. If it does not clear the 706 error or if the 706 error occurs again, contact Sybase Technical Support.

#### **Severity Level**

20

#### **Error Message Text**

```
System error detected during attempt to free
memory at address 0x%lx. Please consult the SQL
Server error log for more details.
```

#### Explanation

The Memory Manager allocates, deallocates, and manages memory for SQL Server. It manages an array of structures in the process header, each of which represents a physical page of memory in procedure cache and the current byte allocation of that page.

As part of its memory deallocation process, SQL Server tries to release the pages of memory allocated to a process header when they are no longer needed. When SQL Server is unable to release that section of memory, Error 707 occurs.

Error 707 is caused by memory corruption or a SQL Server problem.

#### Action

Since process headers are stored in procedure cache, restarting SQL Server should clear Error 707. If it does not clear the 707 error or if the 707 error occurs again, contact Sybase Technical Support.

## **Buffer Manager Errors**

This section contains error messages pertaining to the Buffer Manager.

#### Severity Level

20

#### **Error Message Text**

Unable to place buffer 0x%lx holding logical page %ld in sdes for object '%.\*s' - either there is no room in sdes or buffer already in requested slot.

► Note

"sdes" is an abbreviation for "session descriptors." Refer to "Error 603" on page 2-34 for information about session descriptors.

#### Explanation

Whenever a connection to SQL Server attempts to open a table, it gets a session descriptor. A session descriptor is an internal data structure in SQL Server which contains information about a table. All open objects in SQL Server require a descriptor structure.

To ensure that a buffer read by the session remains in the buffer cache and does not age out, SQL Server "keeps" the buffer. The session descriptor contains eight slots where buffers that are "kept" by the session are stored.

Error 803 occurs during recovery or regular SQL Server operation when one of the following is true:

- SQL Server runs out of session descriptors
- A buffer was already in the session descriptor's requested slot

#### ► Note

The utility **sybload** also has an Error 803 which can occur during installation or upgrade. The last part of this writeup details the **sybload** 803 error.

#### Action

When an 803 error occurs:

• Look for other errors in the SQL Server error log to indicate the specific source of the problem (for example, not enough locks, not enough open objects) and clear those errors first.

• Restart SQL Server so that the configuration change will take effect. Restarting will also clear the buffer cache and the 803 error.

If problems persist, contact Technical Support for assistance. Before contacting Technical Support, be prepared to fax or provide the complete text of all error messages.

Examine your operating system error log file as well as the SYBASE error log to determine if hardware errors may have caused the problem.

## sybload Error 803

## **Error Message Text**

Please inform Sybase Customer Service that there may be a problem with this tape and cite error condition 803.

sybload Error 803 can occur when unloading software because of one of the following:

- When loading SYBASE software from tape, the operator made one of the following errors:
  - The nonrewinding device was not specified when loading SYBASE software from media with sybload.
  - When loading the **sybload** utility for a remote installation, the operator typed the commands for a local installation instead of those provided for a remote installation.

A local installation is one in which the SYBASE installation directory and tape drive are on the same machine. Remote installations are the ones where the SYBASE installation directory and tape drive are on different machines.

• The version of the SQL Server to be installed requires a different version of the operating system.

## Action for sybload Error 803

Use appropriate procedures from the following:

• Check that the SQL Server platform and version you are trying to install or upgrade is compatible with your operating system version. For more information, see the *SYBASE SQL Server Installation Guide* or *Release Bulletin*.

- Check that you specify a nonrewinding device when instructed to do so in the *SYBASE SQL Server Installation Guide* or *Release Bulletin*.
- If you are loading SYBASE software for a remote installation, check that you typed in the correct commands for loading the sybload utility from tape. Remote load commands are different from the commands for a local installation. Refer to the *SYBASE SQL Server Installation Guide* for your platform for details.

#### **Severity Level**

20

#### **Error Message Text**

```
Unable to find buffer 0x%lx holding logical page %ld in sdes 0x%lx kept buffer pool for object '%.*s'.
```

#### ► Note

"sdes" is an abbreviation for "session descriptors." Refer to "Error 603" on page 2-34 for information about session descriptors.

#### Explanation

This error occurs when SQL Server is unable to find the pointer to a buffer header in a session descriptor.

A session descriptor is an internal data structure in SQL Server which contains information about a table. All open objects in SQL Server require a descriptor structure. Whenever a connection to SQL Server attempts to open a table, it gets a session descriptor.

To ensure that a buffer read by the session remains in the buffer cache and does not age out, SQL Server "keeps" the buffer. The session descriptor contains eight slots where buffers that are "kept" by the session are stored. If a buffer that SQL Server is looking for in a session descriptor cannot be found, Error 804 will occur.

Error 804 occurs during normal processing, such as queries and updates.

This error can be transient.

#### Action

When an 804 error occurs, restart SQL Server as soon as possible.

If additional 804 errors occur, call Sybase Technical Support. Before contacting Technical Support, be prepared to fax or provide the complete text of all error messages and your SQL Server version number.

#### **Severity Level**

21

#### **Error Message Text**

```
Could not find virtual page for logical page %ld in database '%S_DBID'.
```

#### Explanation

This error occurs when SQL Server fails to convert a logical page number to a virtual page number. Depending on what caused the error, it can be serious or transient.

A virtual page is the page within a Sybase device whereas a logical page is the page in a SQL Server database. There is a one-to-one correspondence between the two types of pages.

If Error 806 appears with a stack trace and *exec\_dbcc* appears in the stack trace, it means you used **dbcc** page with an invalid parameter. This is **not** a serious problem.

If Error 806 occurs on recovery, it may be transient or serious (see information under "Action" for specifics).

Error 806 can occur during normal processing, such as creating an index or running a stored procedure. In this case, the error is probably caused by corruption or a problem with SQL Server and it is a serious error.

#### Action

If Error 806 specifies *tempdb* in the message output, restart SQL Server. Since *tempdb* is rebuilt each time SQL Server is restarted, this may clear the error. If the error occurs again (on *tempdb*), call Sybase Technical Support.

#### **During Recovery**

If Error 806 occurs on recovery, the database will be marked suspect. If the error is transient, resetting the suspect status will solve the problem. To determine whether this is the case, reset the suspect status one of two ways:

- Execute the sp\_resetstatus procedure supplied in "How to Reset a Database's "suspect" Status" in the *SQL Server Troubleshooting Guide*. This is the safest method. After you execute sp\_resetstatus, go to step 3 in the following set of steps.
- Alternatively, you can do the following:
- 1. Use the following procedure on the suspect database:

```
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
1> use master
2> go
1> begin transaction
2> go
1> update sysdatabases
2> set status = status & ~256
3> where name="database_name"
4> go
```

If only one row is affected by the update transaction, continue with these instructions. If more than one row is affected by the update transaction, roll back the transaction and find out why other rows are being affected.

2. If the above commands affect only one row, use the commands below to commit the transaction, disable updates to the system tables, issue a checkpoint, and shut down SQL Server:

```
1> commit transaction
2> go
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
1> checkpoint
2> go
1> shutdown
2> go
Start SOL Server
```

3. Start SQL Server.

4. Run dbcc checkdb and dbcc checkalloc on the database listed in the error message output.

If Error 806 occurs again, call Sybase Technical Support. Technical Support may be able to help you recover from this error if the corruption is not too widespread. However, recovery from clean backups might be necessary.

#### **During Normal Processing**

If Error 806 occurs during normal processing, determine which object is causing the error. Use the page number in the error message and refer to "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide*.

Run dbcc checkdb and dbcc checkalloc on the database listed in the error message output.

Call Sybase Technical Support. Technical Support may be able to help you recover from this error if the corruption is not too widespread. However, recovery from clean backups might be necessary.

#### **Additional Information**

Refer to the *System Administration Guide* for information about dbcc commands.

Before contacting Technical Support, be prepared to fax or provide:

- SQL Server release and EBF Rollup level
- Complete text of all error messages
- SQL Server error log output
- Operating system error log output
- dbcc checkdb, dbcc checkalloc, and dbcc page output

### **Severity Level**

20

### **Error Message Text**

Logical page %ld in database %d is already hashed.

### Explanation

This error is serious, especially if it occurs on a table's data page. It means that a page is currently in the data cache and is in use by a table or index but is not marked as allocated. An attempt to allocate it causes Error 813.

The same page could be allocated again after it is removed from the data cache, resulting in a loss of whatever data resides on the page.

## ♦ WARNING!

Pages encountering Error 813 will not be included in a database dump. This is because database dumps are performed by reading allocation pagess and not by traversing page chains. Therefore, this error should be corrected before dumping the database.

After the page is removed from the data cache, further attempts to access this page may raise 2500 series errors (allocation errors) when running dbcc checkalloc, dbcc tablealloc, or dbcc indexalloc.

#### Action

To clear the error, do the following:

- 1. Execute the checkpoint command on the database specified in the error message. This will cause all modified pages in the data cache to be flushed to disk.
- 2. Run dbcc checkalloc or dbcc checkalloc with the fix option on the database. As a result, you might get allocation errors because the page displayed in the error message will be linked but not allocated. Refer to the *SQL Server Reference Manual* for information about running these commands.

- 3. If you do get allocation errors:
  - Check your hardware error log or diagnostics utilities for I/O errors.
  - Check your operating system error log file and the SQL Server error log to see if hardware errors may have caused the corruption. (Look for messages reporting I/O errors.)
  - Refer to the writeups in this manual.
- 4. If you still have 813 errors after fixing any other errors reported during step 2, restart SQL Server (using shutdown). This will probably clear Error 813.

If errors still occur, call Sybase Technical Support. They might be able to help you recover from this error if the corruption is not too widespread. However, recovery from backups might be necessary.

# Using dump transaction with no\_log

Using the dump transaction with no\_log command can result in an 813 error. Therefore, do not use dump transaction with no\_log unless it is absolutely necessary (when dump transaction with truncate\_only will not truncate the log). Try using dump transaction with truncate\_only first and use the no\_log option only as a last resort. Refer to "Error 1105" on page 2-120 for details.

Check for any occurrences of dump transaction with no\_log in any of your scripts or procedures and replace them with dump transaction with truncate\_only.

# **Additional Information**

For more information on the 2500 series errors, refer to the information in this section and in "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide*.

### **Severity Level**

20

# **Error Message Text**

```
Attempt to unhash buffer at 0x%lx with a buffer pageno of %ld and database id %ld with HASHED status set failed - buffer was not found. %S_PAGE.
```

### Explanation

Every data or index page in the data cache has a corresponding buffer pointing to it, controlling its stay in the cache. As these pages are read and modified, they are scheduled to be written to disk and removed from the data cache. Error 821 occurs when SQL Server attempts to remove a nonexistent or corrupted page from the data cache.

Some potential causes of this error are:

- Overlapping partitions (see "Correct Use of Raw Partitions" in the *SQL Server Troubleshooting Guide*).
- Hardware failure.
- A SQL Server problem (such as can occur when using the dump transaction with no\_log command after a period of heavy user activity in the database).

This error can occur during database recovery as well as during normal operation of SQL Server.

### Action

- 1. Check the SQL Server error log to see if there are other indications of hardware problems, such as kernel messages reporting I/O errors. If problems exist, solve them.
- 2. Check the operating system error log or run diagnostic utilities to check for I/O errors. If problems exist, solve them.
- 3. Once the actual cause of corruption has been removed, shut down and restart SQL Server to clear the 821 error. If restarting is not feasible and if you have dial-in facilities, Sybase Technical Support may be able to help resolve the immediate problem.

# Using *dump transaction with no\_log*

Using the dump transaction with no\_log command can result in an 821 error. Therefore, do not use dump transaction with no\_log unless it is absolutely necessary (when dump transaction with truncate\_only will not truncate the log). Try using dump transaction with truncate\_only first and use the no\_log option only as a last resort. Refer to "Error 1105" on page 2-120 for details.

Check for any occurrences of dump transaction with no\_log in any of your scripts or procedures and replace them with dump transaction with truncate\_only.

# Additional Information

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Operating system error log
- Output of dbcc checkalloc and dbcc checkdb
- Text of all error messages

**Severity Level** 

20

**Error Message Text** 

Release 10.0 and Later

```
Could not start I/O for request BLKIO flags = 0x%lx, size = %ld, errcode = 0x%lx, %S_BUF.
```

### **Release 4.9.x and Earlier**

Could not start I/O for request %S\_BLKIOPTR

### **Explanation**

This error occurs when a read or write operation was requested and SQL Server encountered an error while attempting to initiate the request. Error 822 can occur for any of the following reasons:

- A database device is offline
- A database device or file has been removed or renamed
- A database device or file is inaccessible to SQL Server for any other reason, such as insufficient permissions

The information in the error message refers to an internal SYBASE structure and does not help determine which database device or file is involved. Other error messages appearing in the SQL Server error log just before the 822 error usually indicate the device involved. Refer to "Additional Information" on page 2-86 for examples of this type of error message.

### Action

- 1. Examine the availability and condition of the device involved using your standard operating system procedures, and make sure the device is accessible.
- 2. SQL Server will not be able to recover the database upon restarting because an 822 error will mark the database suspect. If you know that the database was marked suspect because the device was unavailable, reset the status of the database to allow recovery to continue.

#### ♦ WARNING!

Do not use these procedures if other errors in the error log near the 822 error message suggest that the database may have been marked suspect for some other reason.

Reset the suspect status one of two ways:

- Execute the sp\_resetstatus procedure supplied in "How to Reset a Database's "suspect" Status" in the *SQL Server Troubleshooting Guide*. This is the safest method. After you execute sp\_resetstatus, go to step 3.

Or:

- Use the following procedure on the suspect database:

```
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
1> use master
2> go
1> begin transaction
2> go
1> update sysdatabases
2> set status = status & ~256
3> where name="database_name"
4> go
```

If only one row is affected by the update transaction, continue with these instructions. If more than one row is affected by the update transaction, roll back the transaction and find out why other rows are being affected.

If the above commands affect only one row, use the commands below to commit the transaction, disable updates to the system tables, issue a checkpoint, and shut down SQL Server:

```
1> commit transaction
2> go
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
```

- 1> checkpoint
  2> go
  1> shutdown
- 2> go
- 3. Start SQL Server. If the database is again marked suspect, examine the SQL Server error log for indications of other errors.

#### **Additional Information**

Here are two examples of the 822 error and associated messages from the SQL Server error log. Examining the error messages just before the error usually helps determine the type of problem.

**Example 1: Permission Problems** 

In this example, the permissions on *disk1* (virtual device 6) were incorrect, so SQL Server was unable to access this device:

```
kernel: initializing virtual device 6, "/work/disk1"
kernel: dopen: open "/work/disk1", Permission denied
kernel: dinit: failed to open primary device /work/disk1 for vdn6
kernel: udstartio: vdn 6 has not been set up
server: Error: 822, Severity: 20, State: 3
server: Could not start I/O for request BLKIO ....
```

To correct an error like this:

- 1. Change the ownership to user "sybase". Make sure that this user has read/write permission on the device. In our example, check the user's permission on */work/disk1*.
- 2. Become the "sybase" user.
- 3. Start SQL Server.

#### Example 2: File Missing

In this example, the UNIX file corresponding to *disk1* (virtual device 6) did not exist when SQL Server started, causing access to the device to fail:

```
kernel: initializing virtual device 6, "/work/disk1"
kernel: dopen: open "/work/disk1", No such file or directory
kernel: dinit: failed to open primary device /work/disk1 for vdn6
kernel: udstartio: vdn 6 has not been set up
server: Error: 822, Severity: 20, State: 3
server: Could not start I/O for request BLKIO ....
```

To correct an error like this, make sure that the virtual device in the error message exists, and correct whatever error caused SQL Server

not to find it. (For example, it may have been renamed or moved while SQL Server was shut down.)

If the virtual device no longer exists, you will have to restore from known clean backups. Merely creating an empty device with the right name will not solve this problem.

#### **Severity Level**

24

#### **Error Message Text**

I/O error detected during %S\_MSG for %S\_BUF.

#### Explanation

This error occurs when SQL Server encounters an I/O error on a read or write request made to a SYBASE device. It usually means you have disk problems. The parameters in the error message refer to internal SYBASE structures and do not often help determine which device is involved. However, additional kernel messages in SQL Server's error log recorded before the 823 error should indicate which device is involved.

#### Action

Check the accessibility and condition of the device in question. Once you have identified and corrected the problem, execute the dbcc checkdb and dbcc checkalloc commands to ensure that no other damage was caused by the bad device.

#### SunOS 4.x (Sun Solaris 1.x)

The 823 error can also appear if the Sun operating system is not configured correctly for asynchronous I/O.

SQL Servers running on Sun machines can use asynchronous I/O on raw partitions without any special kernel enhancements such as the MtXinu DBM or Sun DBE. A SQL Server running on a SunOS 4.1 or later operating system attempts to do asynchronous I/O on raw partitions. Successful use of asynchronous I/O requires that the UNIX kernel be configured to support it. Two options must be included in the Sun operating system kernel configuration file when the kernel is built. These are:

You can find examples of how to set these options in the original GENERIC configuration file usually found in /sys/{sun3, sun4}/conf. The exact location of these files may vary. Do not comment out these options.

An improperly configured kernel causes SQL Server to fail the first time it attempts to do I/O to a raw partition. For example, if your master device is on a raw partition, and you attempt to start SQL Server with that master device, SQL Server displays the following message:

```
kernel: read error on virtual disk 0 block 28
kernel: invalid argument
server: Failure to open master database for the first time
server: Error 823, Severity 24, State 1
```

If the master device is located on a UNIX file for which standard disk I/O rather than asynchronous I/O is used, SQL Server starts even if the Sun kernel is not configured for asynchronous I/O. However, if you attempt a disk init command on a raw partition, it fails with a 5115 error. See "Error 5115" on page 2-380 for more information.

### **AIX Systems Only**

Error 823 may occur if the disk device is run under the Logical Volume Manager (LVM) and vstart is less than 2. vstart needs to be equal to 2 because the first AIX block (1 AIX block equals 2 SYBASE pages) has to be available for the LVM control block (LVCB).

**Severity Level** 

17

**Error Message Text** 

Release 10.0.1 and Later

Device '%.\*s' (with physical name '%.\*s', and virtual device number %d) has not been correctly activated at startup time. Please contact a user with System Administrator (SA) role.

### Release 10.0

Device '%.\*s' (with physical name '%.\*s', and virtual device number %d) has not been correctly activated at startup time. Please contact a user with System Administrator (SA) authorization.

#### Release 4.9.x and Earlier

```
Device '%.*s' (with physical name '%.*s', and
virtual device number %d) has not been correctly
activated at startup time. Please contact System
Administrator for assistance.
```

#### Explanation

This error can occur when SQL Server is unable to access a device during startup.

This error can occur for any of the following reasons:

- A database device is offline
- A database device or file has been removed or renamed
- A database device or file is inaccessible to SQL Server for any other reason, such as insufficient permissions

#### Action

- 1. Examine the availability and condition of the device involved using your standard operating system procedures, and make sure the device is accessible.
- 2. SQL Server will not be able to recover any databases that have space on the device upon restarting because an 840 error will mark the databases suspect. If you know that the databases were marked suspect because the device was unavailable, resolve the problem with the device and reset the status of the databases to allow recovery to continue.

#### ♦ WARNING!

Do not use these procedures if other errors in the error log near the 840 error message suggest that the databases may have been marked suspect for some other reason.

Reset the suspect status one of two ways:

- Execute the sp\_resetstatus procedure supplied in "How to Reset a Database's "suspect" Status" in the *SQL Server Troubleshooting Guide*. This is the safest method. After you execute sp\_resetstatus, go to step 3.

#### Or:

- Use the following procedure on the suspect database:

```
1> use master
2> go
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
1> begin transaction
2> go
1> update sysdatabases
2> set status = status & ~256
3> where name="database_name"
4> go
```

If only one row is affected by the update transaction, continue with these instructions. If more than one row is affected by the update transaction, roll back the transaction and find out why other rows are being affected. If the above commands affect only one row, use the commands below to commit the transaction, disable updates to the system tables, issue a checkpoint, and shut down SQL Server:

```
1> commit transaction
2> go
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
1> shutdown
2> go
```

- 3. Start SQL Server.
- 4. Execute the dbcc checkdb and dbcc checkalloc commands for the affected databases to ensure that no other damage was caused by the device being unavailable. If other errors occur, refer to the writeups for those errors in this manual. If errors still persist, call Sybase Technical Support.

# **Open Database Manager Errors**

This section contains error messages pertaining to the Open Database Manager.

# **Severity Level**

23

# **Error Message Text**

Unable to find row in sysindexes for clustered index on system catalog %d in database %d. This index should exist in all databases. Run DBCC CHECKTABLE on sysindexes in the database.

### Explanation

This error occurs when SQL Server is unable to access the *sysindexes* table for a database.

Error 903 can occur under the following circumstances when SQL Server accesses *sysindexes*:

- Actions that involve opening a database since those actions access the *sysindexes* table for that database (for example, database recovery)
- Actions that need to access columns in *sysindexes*. Such actions include:
  - order\_by clause compilation.
  - Running the update statistics command.
  - Threshold Manager space calculations. Refer to "Thresholds and Transaction Log Management" in the *SQL Server Troubleshooting Guide* for information about thresholds.

Possible causes of Error 903 are:

- A problem has occurred with the device on which *sysindexes* resides and *sysindexes* has become corrupted.
- The partition on which the database resides has been mounted as a file system or overlapping partitions exist. Refer to "Correct Use of Raw Partitions" in the *SQL Server Troubleshooting Guide*.
- A SQL Server problem.

#### Action

Determine whether the device on which the database resides is damaged and solve any problems that exist.

If the database named in the error message is accessible, run dbcc checkdb and dbcc checkcatalog to determine whether the database is corrupted. If the database is corrupted, restore it from backup.

If you cannot find any device problems and the database is not corrupted, call Sybase Technical Support.

#### **Additional Information**

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- select \* from database\_name..sysindexes where id=object\_id("sysindexes") output
- dbcc checkdb and dbcc checkcatalog output
- Text of all the error messages

# **Severity Level**

17

# **Error Message Text**

```
Unable to allocate a DBTABLE descriptor to open
database '%S_DBID'. Another database must be
closed or dropped before opening this one.
```

### Explanation

As part of open database processing, the Database Table Manager controls and synchronizes access to database tables. When SQL Server tries to open a database and there are not enough database descriptors available, Error 905 occurs. (The total number of database descriptors available is controlled by the **open databases** configuration parameter.)

The action in the text of the 905 message only applies to environments where:

- You cannot have any more open databases (for instance, because of memory limitations), or
- You cannot increase the open databases configuration parameter until a later date.

### Action

When Error 905 occurs, you can do one of the following to solve the problem:

• In the short term, you can close or drop another database. This is not a long-term solution. You can use this option until a SQL Server restart can be scheduled, as a restart is necessary for the long-term solution.

To close a database, make sure no users are accessing the database or put the database into single-user mode. This does not guarantee that the database will not be accessed by SQL Server (for example, for an automatic checkpoint) and so the 905 error may still occur. • As a long-term solution, use sp\_configure to increase the open databases configuration parameter. First, determine your current value:

```
1> sp_configure "open databases"
2> go
```

name	minimum	maximum	config_value	run_value
open databases	5	2147483647	0	12

Determine what your new value should be. One way to do this is to determine the number of databases you have now and then add a padding factor for future growth.

For example:

```
1> select count(*) from sysdatabases
2> go
------
```

12

Then set the new value:

```
1> sp_configure "open databases", 15
2> go
1> reconfigure
2> go
```

where "15" is the new value (12 existing databases plus a padding factor of 3 for future growth).

Shut down and restart SQL Server to put the new value in effect.

► Note

If you increase the value of the *open databases* configuration parameter, make sure you reconfigure SQL Server memory appropriately. In SQL Server release 10.0 and later, 17,408 bytes are required for each additional open database. In SQL Server release 4.9 x, 7,500 bytes are required for each additional open database. Refer to "SQL Server Memory Issues" in the *SQL Server Troubleshooting Guide* for information about SQL Server memory use.

# **Severity Level**

23

# **Error Message Text**

Could not locate row in sysobjects for system catalog %d in database %d. This system catalog should exist in all databases. Run DBCC CHECKTABLE on sysindexes in the database.

# Explanation

SQL Server expects to find certain information in static locations within a database or on the Server: page 1 of each database is the first page for *sysobjects*; page 24 of each database is the first page for *sysindexes*.

As part of open database processing, the Descriptor Manager searches page 1 for entries for *sysobjects* and page 24 for entries for *sysindexes*. If the entries are not found on the expected page, Error 906 will occur.

Error 906 occurs during open database actions such as recovery and create database.

Possible causes of Error 906 are:

- The partition on which the database resides has been mounted as a file system or overlapping partitions exist. Refer to "Correct Use of Raw Partitions" in the *SQL Server Troubleshooting Guide*.
- A database device has been created which exceeds the SQL Server maximum size limit (2GB).

# Action

Error 906 means that, for the database specified in the message, either page 1 or page 24 is corrupted. Restore that database from backup.

#### **Severity Level**

16

### **Error Message Text**

Attempt to locate entry in sysdatabases for database '%.\*s' by name failed - no entry found under that name. Make sure that name is entered properly.

#### Explanation

This error occurs during a use command when SQL Server tries to access a database that is not listed in *sysdatabases*. For example:

A user tries to access a database not listed in sysdatabases:

1> use database\_name
2> go

• A user tries to connect to SQL Server when an invalid database is indicated by the *defaultdb* column of *master..syslogins*. The user is connected only to *master*, or not connected at all, and Error 911 occurs.

#### Action

1. Obtain a list of all databases in *sysdatabases* by executing sp\_helpdb or the following:

1> select name from master..sysdatabases
2> go

The list returned contains the only valid databases recognized by SQL Server and available for use.

- 2. Either create the missing database or change the reference to a database from the list created in step 1:
  - If a user-issued use command created the error, either create a database following regular procedures (see the *System Administration Guide* or the *SQL Server Reference Manual*) or change the use command to refer to one of the databases from the list created in step 1.
  - If the defaultdb option of sp\_addlogin or sp\_defaultdb refers to an invalid database, the System Administrator ("sa") or user needs to specify a new, valid, default database. To specify a

new default database, choose a database name from the list in step 1 and use the following procedure:

1> sp\_defaultdb user\_name, database\_name
2> go

Although you can successfully change your own default database at any time with sp\_defaultdb, you can access the database only if the database owner ("dbo") has added you as a user in the database.

For more information about sp\_addlogin and sp\_defaultdb see the *SQL Server Reference Manual.* 

#### **Severity Level**

22

#### **Error Message Text**

```
Could not find row in sysdatabases with database id %d. Run DBCC CHECKTABLE on sysdatabases.
```

#### Explanation

This error occurs when SQL Server cannot find an entry in *sysdatabases* when looking for a specific database ID.

Possible causes of Error 913 are:

- Accessing a stored procedure or view that refers to a table in a database that has been dropped.
- A nonclustered index on the *sysdatabases* table in the *master* database has been corrupted.

This is a serious error if the index is corrupt or recovery fails.

#### Action

When a database has been dropped and re-created, it can end up with a different database ID. If a stored procedure still references the old database ID, recompile the stored procedure so that it will reference the new database ID. Refer to "Building the Query Plan: Compilation" in the *SQL Server Troubleshooting Guide* for information about recompiling stored procedures.

If the cause of the error is a corrupted nonclustered index, either restore *master* from backup or call Sybase Technical Support to determine whether they can help you repair the index. Refer to "System Database Recovery" in the *SQL Server Troubleshooting Guide* for information about restoring the master database.

#### **Severity Level**

14

### **Error Message Text**

```
Server user id %d is not a valid user in database '%.*s'
```

### Explanation

This error occurs when a user tries to open a database and SQL Server finds that the user ID is not valid.

SQL Server performs the following checks to determine the validity of the user:

- 1. Looks for the *suid* (SQL Server user ID, copied from *syslogins*) of the process in *sysusers* to find the corresponding *uid* (user ID, unique in this database) and *gid* (group ID to which this user belongs).
- 2. Looks for a "guest" row in *sysusers* (the *suid* for "guest" is -1).
- 3. Looks for the user ID in sysalternates.
- 4. In SQL Server release 10.0 and later, checks *sysusers* to determine whether the user has "sso" role if the user does have an *suid* in the database.

### Action

1. Check whether the SQL Server user ID exists in syslogins:

```
1> use master
2> go
1> select * from syslogins
2> where suid = Server_user_ID
3> go
```

where *Server\_user\_ID* is the SQL Server user ID displayed in the error message.

2. Check whether there is a "guest" row in *sysusers* of the database being opened:

1> use database\_name
2> go

```
1> select * from sysusers
2> where suid = -1
3> go
```

where *database\_name* is the name of the database in the error message.

3. Check whether the user ID exists in *sysalternates* of the database being opened:

```
1> use database_name
2> go
1> select * from sysalternates
2> where suid = Server_user_ID
3> go
```

4. For SQL Server release 10.0 and later, if the user does have an *suid* in the database, check *sysusers* to determine whether the user has "sso" role. Use the following commands to list all users with "sso" role:

```
1> use master
2> go
1> select l.name, s.name from sysloginroles slr,
2> syslogins l,syssrvroles s
3> where slr.srid = 1 and slr.suid = l.suid and
4> slr.srid = s.srid
5> go
```

5. If the database is being accessed due to a referential constraint, make sure the user ID exists in each database.

If Error 916 occurs after the above actions have been taken, call Sybase Technical Support.

#### **Additional Information**

Refer to the *SQL Server Reference Manual* for information about the sp\_adduser command.

Have the following information ready when you call Sybase Technical Support:

- SQL Server release and EBF rollup level
- Text of all error messages
- Output from syslogins
- Output from *sysusers* and *sysalternates* for the database being opened

#### **Severity Level**

14

### **Error Message Text**

```
Database '%.*s' has not been recovered yet - please wait and try again.
```

### Explanation

This error occurs during the database recovery phase of SQL Server startup when you try to access a database for which recovery has not yet been completed. It can also occur if the database was marked suspect due to an earlier problem.

If the database has been marked suspect by an earlier problem, this is a serious error and must be corrected if you want to access your database again.

# Action

The time it takes to recover a database depends on:

- The number and size of transactions that must be committed or rolled back
- The amount of memory

Determine whether the recovery phase of the database has completed:

- Examine your SQL Server error log
- Enter two select \* from sysprocesses where spid=1 commands about 20 seconds apart. Find the *spid* of the process doing the recovery (the value of *spid* for the process doing database recovery is always "1" during SQL Server startup) and see if *physical\_io* is increasing over time. If it is, recovery is still in progress.

If the database is still being recovered, wait until recovery is complete and then try to access the database.

If you are still unable to access the database and the database has been marked suspect because of an earlier problem, determine what the earlier problem was by looking at the SQL Server error log. Follow the instructions in this manual for recovering from that error. If no instructions exist, call Sybase Technical Support.

# **Additional Information**

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log, including boot sequence and all errors
- Operating system error log
- select \* from master..sysusages output
- select dbid, name, status from sysdatabases output
- Text of all the error messages

#### **Severity Level**

14

### **Error Message Text**

Database '%.\*s' is already open and can only have one user at a time.

### Explanation

This error occurs when an attempt is made to access a database that has been set to single-user mode and is being accessed by another user. Only one user at a time can access a database that is set to single-user mode.

#### Action

1. Verify that the database being accessed is in single-user mode:

```
1> sp_helpdb database_name
2> go
```

If the database is in single-user mode you will see "single user" in the *status* column along with any other options that are set for the database.

- 2. Execute sp\_who to see who is accessing the database. Look for the database name under the *dbname* column and the login name in the *loginame* column.
- 3. Contact the individual and arrange for database access or contact your System Administrator ("sa"). The "sa" or database owner ("dbo") can set the database to multi-user access when the database is not in use by executing the following from the *master* database:

```
1> sp_dboption database_name, single, false
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

For further information on sp\_dboption, see the *SQL Server Reference Manual*.

#### **Severity Level**

19

#### **Error Message Text**

Maximum number of used databases for each query has been exceeded. The maximum allowed is %d.

#### Explanation

SQL Server has an eight-level-deep stack of open databases for each process. Every time a database is opened, it gets pushed onto the stack, and when it is closed, it is removed from the stack.

There is usually one open database for each process. If you run a query that uses a table in a different database, SQL Server opens that database just long enough to use the table and then closes the database.

Sometimes, however, a database must remain open longer. For example, if you execute a stored procedure in a different database than the one you are currently using, SQL Server will open that database and keep it open until the stored procedure finishes.

Only eight databases can be open at once. A single database opened multiple times is added to the stack each time it is opened. This can result in Error 925, even though only one database has been used. If you have stored procedures that call other procedures in other databases, you can exceed this limit.

In addition, if a SQL Server problem causes an open database to remain on the stack even though it should have been closed, Error 925 occurs.

#### ► Note

The use *database\_name* command closes the database that was previously open, so Error 925 would **not** be generated by eight use *database\_name* commands in a row in Transact-SQL.

An example of a set of queries that exceeds the eight-level limit follows:

```
1> use db9
2> go
```

```
1> create proc pr9
2> as
3> select * from table1
4> go
1> use db8
2> go
1> create proc pr8
2> as
3> exec db9..pr9
4> go
1> use db7
2> go
1> create proc pr7
2> as
3> exec db8..pr8
4> go
1> use db6
2> go
1> create proc pr6
2> as
3> exec db7..pr7
4> go
1> use db5
2> go
1> create proc pr5
2> as
3> exec db6..pr6
4> go
1> use db4
2> go
1> create proc pr4
2> as
3> exec db5..pr5
4> go
1 > use db3
2> go
1> create proc pr3
2> as
3> exec db4..pr4
4> go
1> use db2
2> go
```

```
1> create proc pr2
2> as
3> exec db3..pr3
4> go
1> use db1
2> go
1> create proc pr1
2> as
3> exec db2..pr2
4> go
```

#### Action

Be aware of the eight-level open database limit when writing queries.

If you believe Error 925 is occurring for a reason other than a query that has more than eight database openings, restarting SQL Server may clear the problem. If the problem persists, call Sybase Technical Support.

**Severity Level** 

14

# **Error Message Text**

# Release 10.0 and Later

Database '%.\*s' cannot be opened. An earlier attempt at recovery marked it 'suspect'. Check the SQL Server errorlog for information as to the cause.

# **Release 4.9.x and Earlier**

Database '%.\*s' cannot be opened - it has been marked SUSPECT by recovery. The SA can drop the database with dbcc.

### Explanation

This error occurs when you attempt to reference a database that has been marked suspect in one of the following circumstances:

- During start-up of SQL Server
- By the System Administrator as a result of certain critical errors

This is a serious error and must be corrected if you want to access your database again. This error may be caused by hardware failure.

### Action

Since the 926 error is the result of an earlier error or action, the recommended action is to determine what caused the database to be marked suspect.

In order to determine the cause, check the SQL Server error log for error messages for the database in question and try to eliminate those first by using the troubleshooting procedures in this manual.

Depending on why the database was marked suspect, you may choose to remove its suspect flag if you are certain that the critical error which caused the database to be marked suspect has been resolved (for example, if one of the database devices was not available when SQL Server was booted and you are sure that the device is available now). If you cannot find any procedures recommended for your specific errors, call Sybase Technical Support for assistance.

If the specified database does not contain important data or if you have a known, clean backup of it, you may choose to drop it first, re-create it, and then load the clean database dump into it. Refer to "How to Drop a Database When drop database Fails" in the *SQL Server Troubleshooting Guide* for information on how to drop a database that has been marked suspect.

Before loading the database dump into the newly created database, make sure that the new database and the dumped database have the same data and log mapping, and the same user segment definitions. Refer to "Error 2558" on page 2-274 for information about how to do this.

#### **Additional Information**

For more information on how to remove the suspect flag from a database, refer to sp\_resetstatus in "How to Reset a Database's "suspect" Status" in the *SQL Server Troubleshooting Guide*.

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log, including boot sequence and all errors
- Operating system error log
- select \* from master..sysusages output
- select dbid, name, status from sysdatabases output
- Text of all the error messages

**Severity Level** 

14

### **Error Message Text**

### Release 10.0 and Later

Database '%.\*s' cannot be opened because either an earlier system termination left LOAD DATABASE incomplete or the database is created with 'for load' option. Load the database or contact a user with System Administrator (SA) role.

# **Release 4.9.x and Earlier**

Database '%.\*s' cannot be opened because an earlier system termination left LOAD DATABASE incomplete. Reload the database or notify the System Administrator.

### **Explanation**

This error occurs when you attempt to reference a database:

- whose load database process has been terminated before completion, or
- that was created with the for load option and that has not been loaded yet.

### Action

Eliminate the 930 error by determining which of the above situations applies to you and then following the appropriate set of steps below.

**Previous Load Database Failed** 

Since the 930 error is, in this case, the result of an earlier error, the recommended action is to determine what caused the load database process to be terminated before completion.

In order to determine this, check the SQL Server (all releases) and Backup Server (release 10.0 and later) error logs for error messages about the database in question, and try to eliminate those first by using the troubleshooting procedures in this manual. Examine your operating system error log file and the SQL Server error log to determine if hardware errors may have caused the corruption.

To proceed, follow these steps:

- 1. Drop the database using the drop database command.
- 2. Create a database for loading. Make sure the database you create has sizes as least as large as those in *sysusages* for the original database (and that all other *sysusages* values match the original values). See the entry for create database for load in the *SQL Server Reference Manual* for releases 4.8 and later, or the entry for create database for releases for releases earlier than 4.8.
- 3. Do the load again, using load database.

If you cannot find any procedures recommended for your specific errors in this manual, call Sybase Technical Support for assistance.

### Incorrect Use of create database for load

Load the database before you try to reference it.

If you do not have enough space allocated to the specified database and get Error 3105 when you attempt to load the database, use one of the following methods to increase the size of the database to be loaded:

• Drop the newly created database first and then re-create it with the size specified in the Error 3105 message. If you cannot drop the database with **drop database** because the database has been marked suspect, and you are getting Error 926, see "How to Drop a Database When drop database Fails" in the *SQL Server Troubleshooting Guide*.

Or:

• Use the command alter database db\_name for load.

To avoid (pre-10.0 release) 2558 errors or a remapped database (10.0 and later), the database being loaded into must be identical to the dumped database in terms of data, log, and segment mapping. See "Error 2558" on page 2-274 for more information.

If the specified database is the default database for a certain user, that user will get Errors 930 and 4001 and will be put in the *master* database when they connect to the SQL Server. Use the following commands if you want to change that user's default database:

1> sp\_defaultdb user\_loginame, new\_defaultdb
2> go

# **Additional Information**

For more information, see:

- The create database and alter database entries in the *System Administration Guide*
- "Error 2558" on page 2-274 of this manual
- "How to Drop a Database When drop database Fails" in the SQL Server Troubleshooting Guide

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Operating system error log
- select \* from master..sysusages output
- select \* from master..sysdatabases output
- Text of all the error messages

#### **Severity Level**

22

#### **Error Message Text**

```
Logical page %1d of the log encountered while
retrieving highest timestamp in database '%.*s' is
not the last page of the log and we are not
currently recovering that database.
```

#### Explanation

This error occurs during recovery when SQL Server fails to initialize a database timestamp to prepare a database for use. The failure occurs because the timestamp at the end of the transaction log is not consistent with the root page for *syslogs*. This error can occur if SQL Server goes down without completing the updates to database structure.

This is a serious error but you can usually recover from it.

#### Action

Follow these steps to recover from Error 933:

- 1. The affected database will be marked suspect. Reset the suspect status in one of the following two ways:
  - Execute the sp\_resetstatus procedure supplied in "How to Reset a Database's "suspect" Status" in the *SQL Server Troubleshooting Guide*. This is the safest method. After you execute sp\_resetstatus, go to step 2.

Or:

- Use the following procedure on the suspect database:

```
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
1> use master
2> go
1> begin transaction
2> go
```

```
1> update sysdatabases
2> set status = status & ~256
3> where name="database_name"
4> go
```

If only one row is affected by the update transaction, continue with these instructions. If more than one row is affected by the update transaction, roll back the transaction and find out why other rows are being affected.

If the above commands affect only one row, use the commands below to commit the transaction, disable updates to the system tables, issue a checkpoint, and shut down SQL Server:

```
1> commit transaction
2> go
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
1> checkpoint
2> go
1> shutdown
2> go
```

- 2. Start SQL Server.
- 3. Execute the dbcc checkdb, dbcc checkalloc, and dbcc checkcatalog commands for the affected database to ensure that no other errors occur. If no other errors occur, the error was transient. If the 933 or other errors occur, call Sybase Technical Support or restore the database from backups.

#### **Severity Level**

21

### **Error Message Text**

```
Function %s was asked to locate page %ld in database %ld. The page number is invalid for the database.
```

#### Explanation

When the SQL Server Open Database Manager requests a page (for allocating extents and a number of other operations), it receives a page number and a pointer to a disk map array. It then determines the index of the map element that contains the target page and returns that index. When SQL Server cannot find the requested page number in the database disk map array, Error 945 occurs.

Possible causes of Error 945 are:

- Memory corruption
- Bad allocation pages pointers (sysusages table corruption)

#### Action

- 1. Shut down and restart SQL Server. If the corruption is in memory, this should clear the 945 error. If Error 945 persists, go on to the step 2.
- Check your record of *sysusages* information for the database in the error message against what is currently in *master..sysusages*. (If you do not have a record of your *sysusages* information, try sp\_help *database\_name* to get a subset of the *sysusages* information.) To determine the *sysusages* information currently held by SQL Server, type:

1> select \* from sysusages where dbid = database\_ID
2> go

where *database\_ID* is the number from the error message output.

3. If the information you have on record for that database does not match what is currently in *sysusages*, refer to "Restoring Device and Database Information in the System Catalog" in the *SQL Server Troubleshooting Guide* for information about restoring a corrupted *sysusages* table.

4. If the information you have on record for that database matches what is currently in *sysusages*, the database specified in the error message is corrupt. Restore the database from known, clean backups.

# Page Manager Errors

This section contains error messages pertaining to the Page Manager.

### **Severity Level**

17, State 1, 2, 3, or 4

### **Error Message Text**

Can't allocate space for object '%.\*s' in database '%.\*s' because the '%.\*s' segment is full. If you ran out of space in syslogs, dump the transaction log. Otherwise, use ALTER DATABASE or sp\_extendsegment to increase the size of the segment.

#### Explanation

This error occurs when SQL Server is unable to allocate space for a database. The state of the error indicates what type of space could not be allocated:

SQL Server Release	Type of Space That Cannot Be Allocated			
	State 1	State 2	State 3	State 4
10.0	Data space	Extent allocation space	Log space	Log space
4.8 through 4.9.1	Data space	Extent allocation space	Log space	Not applicable
Earlier than 4.8	Data space	Log space	Not applicable	Not applicable

Table 2-3: Allocation error: type of space

1105 errors are written to the error log only when they occur during the checkpoint process. When a user process encounters this error, the message is returned to the client application without being written to SQL Server's error log.

To correct the error, you must know:

- The SQL Server release level.
- Whether the error occurred during run time or during recovery:
  - Run time is a period during which SQL Server performs normal daily activities, such as receiving a query from a client application, processing a transaction, or returning results from a transaction.

- A SQL Server is in recovery when doing activities such as restarting, performing the processing that occurs after a database dump is loaded, and performing the processing that occurs after a transaction log is loaded.
- Which state the error is: 1, 2, 3, or 4.

Use the "Action" section below which corresponds to your combination of these factors to correct this error.

#### Action

The "Action" sections for this writeup are organized as follows:

- "Action for SQL Server Release 10.0 or Later" on page 2-122
  - "Run-Time 1105 Errors: State 4" on page 2-122
- "Action for SQL Server Release 4.8 or Later" on page 2-122
  - "Run-Time 1105 Errors" on page 2-122
  - "Run-Time 1105 Errors: State 1 or State 2" on page 2-122
  - "Run-Time 1105 Errors: State 3" on page 2-122
  - "Run-Time 1105 Errors: tempdb" on page 2-124
  - "Recovery 1105 Errors" on page 2-125
  - "Recovery 1105 Errors on a User Database" on page 2-126
  - "Recovery 1105 Errors on the master Database" on page 2-127
  - "Recovery 1105 Errors on the model Database" on page 2-128
- "Action for SQL Server Earlier Than 4.8" on page 2-129
  - "Run-Time 1105 Errors" on page 2-129
  - "Run-Time 1105 Errors: State 1" on page 2-130
  - "Run-Time 1105 Errors: State 2" on page 2-130
  - "Recovery 1105 Errors" on page 2-131
  - "Recovery 1105 Errors on a User Database" on page 2-131
  - "Recovery 1105 Errors on the master Database" on page 2-132
  - "Recovery 1105 Errors on the model Database" on page 2-133

### Action for SQL Server Release 10.0 or Later

### Run-Time 1105 Errors: State 4

The log space is full on the indicated database. LCT (last chance threshold) was trying to empty the log but ran out of space trying to log information. Perform the same procedure as in "Run-Time 1105 Errors: State 3" on page 2-122.

If dbcc checktable on *syslogs* shows space available in the transaction log, check your *sysusages* system table. Error 1105 may occur if the *unreservedpgs* column for this database has NULL values in it when it should have numeric values. Update that column to 0 (instead of NULL).

# Action for SQL Server Release 4.8 or Later

If the error occurred on a SQL Server release earlier than 4.8, see "Action for SQL Server Earlier Than 4.8" on page 2-129. If the error occurred on SQL Server release 4.8 or later, use the sections below to correct the error, depending on whether the error was at run time or during recovery.

# **Run-Time 1105 Errors**

If the error occurred during run time on a 4.8 SQL Server or later, use the next section to correct the error. See "Recovery 1105 Errors" on page 2-125 if the error occurred during recovery.

#### Run-Time 1105 Errors: State 1 or State 2

The data segment is full on the indicated database. To obtain more data space, do one or both of the following:

- Use alter database to increase the size of the data segment.
- Drop objects from the database or delete rows from a table.

#### Run-Time 1105 Errors: State 3

The log segment is full on the indicated database. To clear space in the log, perform the following steps:

1. Determine approximately how many pages the transaction log occupies using the commands:

```
1> use database_name
2> go
```

```
1> select data_pgs (8, doampg)
2> from sysindexes where id=8
3> go
```

where "8" is the ID for *syslogs*. The result is the number of data pages (2K pages on most operating systems) that the transaction log occupies.

```
► Note
```

The query results may be inaccurate by as many as 16 pages, but using the query is much faster than counting the rows in the transaction log.

2. Dump the inactive portion of the transaction log using the dump transaction command. If this command fails with the 1105 error, retry the command using the with truncate\_only option.

If this command also fails with the 1105 error, run dump transaction with no\_log.

3. Repeat step 1. If the transaction log now occupies significantly fewer pages, continue with step 4 now.

If, however, there are still a large number of pages in the *syslogs* table, an outstanding transaction is probably preventing the log from being cleared. (See "Transaction Log Management" in the *SQL Server Troubleshooting Guide* for more information about the transaction log.) If this is the case, restart SQL Server and repeat step 2. When SQL Server starts and this database is recovered, the outstanding transaction is rolled back, allowing the log to be cleared by a subsequent dump transaction.

4. If the dump transaction command was executed with either the no\_log or the truncate\_only options in step 2, dump the database now, because these options prevent subsequent changes recorded in the log from being used to recover from a media failure. You must do a dump database now to ensure the recoverability of subsequent database modifications using transaction logs.

#### ► Note

This database dump is not required if your backup and recovery procedures do not include saving the transaction logs for media failure recovery.

Do not assume that the occurrence of 1105 errors automatically means your transaction log is too small. If the data and the log are on the same segment, either the State 1 and State 2, or the State 3 actions may free enough space without increasing the size of the transaction log.

If you are concerned that your transaction log is not large enough for your SQL Server, read "Transaction Log Management" in the *SQL Server Troubleshooting Guide* before deciding to increase the transaction log size. If appropriate, see the *SQL Server Reference Manual* for more information about the alter database command used to increase log size. Remember that once they have been increased, log and database sizes cannot be decreased.

Run-Time 1105 Errors: tempdb

1105 errors occur on *tempdb* while SQL Server is processing transactions, never during recovery.

The *tempdb* transaction log behaves just like a user database transaction log with the sp\_configure option trunc log on chkpt set on. But data management is more difficult with *tempdb* because it is more difficult to get information about space use: work tables cannot be referenced by the user, and the log is truncated approximately every sixty seconds. The default size of *tempdb* is 2MB when SQL Server is installed.

# ► Note

Read all relevant sections in "Transaction Log Management" in the *SQL Server Troubleshooting Guide* and all the sections about transaction log management in the *System Administration Guide* before making any changes to your site.

# Possible causes of the 1105 error in *tempdb*, and their symptoms and solutions include:

Symptom	Possible Causes	Solution
Too many transactions fill up <i>tempdb</i> transaction log.	Doing too much work in <i>tempdb</i> (or temporary tables) fills up the log.	Design your application for more efficient use of <i>tempdb</i> .
	Processing more transactions per 60 seconds than can be logged in <i>tempdb</i> with current space.	Increase size of <i>tempdb</i> 's transaction log (see the alter database entry in the <i>SQL</i> <i>Server Reference Manual</i> for details) or perform manual dump transaction with truncate only commands on <i>tempdb</i> .
One transaction is too big.	Your transaction may create a Cartesian product which fills the log.	Rewrite the transaction to avoid Cartesian products and long individual queries.
	A begin transaction may have too many steps before the commit transaction.	Read "Transaction Log Management" in the <i>SQL</i> <i>Server Troubleshooting Guide</i> for information and suggested
	A single statement such as insert, delete or update may involve so many rows that the transaction log fills up before the statement can finish.	strategies.

#### Table 2-4: Causes of Error 1105 in tempdb

# **Recovery 1105 Errors**

If the 1105 error occurred during run time, use the section "Run-Time 1105 Errors" on page 2-122 to correct the error. If the error occurred during recovery, use the procedure below which corresponds to the database on which the 1105 error occurred:

- A user database
- The *master* database
- The model database

#### **Recovery 1105 Errors on a User Database**

Use the following procedure to correct 1105 errors on a user database which occur during recovery:

1. Manually set the database's status to no chkpt on recovery and single-user:

#### ♦ WARNING!

Because the database was marked suspect on the original recovery attempt, this procedure also resets some internal status bits to allow the database to recover normally. Therefore, *do not use this procedure under any other circumstances*, or you may create additional serious problems.

```
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
1> begin transaction
2> go
1> update master..sysdatabases
2> set status = status & ~256
3> where name = "database_name"
4> go
1> update master..sysdatabases
2> set status = status | 4112
3> where name = "database_name"
4> go
```

2. Check that each of the above update commands affected only one row. If more than one row was affected, issue a rollback transaction. Otherwise, commit the transaction and shut down SQL Server:

```
    commit transaction
    go
    shutdown
    go
```

- 3. Restart SQL Server.
- 4. Dump the transaction log with the **no\_log** option and reset the database status:

```
1> use master
2> go
```

```
1> dump tran database_name with no_log
2> go
1> sp_dboption database_name, "no chkpt", false
2> go
1> sp_dboption database_name, single, false
2> go
1> use database_name
2> go
1> checkpoint
2> go
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
```

Recovery 1105 Errors on the master Database

If an 1105 error occurs on the *master* database during recovery and SQL Server will not run, correct the problem with the following procedure:

- 1. Create a runserver file that includes the 3607 trace flag. This trace flag allows SQL Server to restart without going through normal recovery, and should only be used in this procedure. See "How to Start SQL Server with Trace Flags" in the *SQL Server Troubleshooting Guide* for instructions.
- 2. Start SQL Server with the 3607 trace flag using the runserver file created in step 1. The recovery process will perform the minimum actions required to allow access to each database, including *master*.
- 3. Once SQL Server has restarted and the *master* database is accessible, dump the transaction log with the no\_log option and shut down the SQL Server:

```
1> dump tran master with no_log
2> go
1> shutdown with nowait
```

- 2> go
- 4. Restart SQL Server with the original runserver file. The database should recover normally. Now dump the *master* database:

```
1> dump database master
2> to logical_dump_device
3> go
```

Recovery 1105 Errors on the model Database

If an 1105 error occurs on the *model* database during recovery, SQL Server may not start. The error occurs because the *tempdb* database is required to start SQL Server, but cannot it be built due to the problem with *model*. To correct the problem and restart SQL Server follow these steps:

- 1. Create a new runserver file that includes the 3608 trace flag. This trace flag causes SQL Server to recover only the *master* database. See "How to Start SQL Server with Trace Flags" in the *SQL Server Troubleshooting Guide* for instructions.
- 2. Start SQL Server with the 3608 trace flag using the runserver file created in step 1.
- 3. Manually set *model* status to no chkpt on recovery and single-user, as follows:

```
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
1> begin transaction
2> go
1> update master..sysdatabases
2> set status = status & ~256
3> where name = "model"
4> go
1> update master..sysdatabases
2> set status = status | 4112
3> where name = "model"
4> go
```

4. Check that each of the above update commands affected only one row. If more than one row was affected, issue a rollback transaction. Otherwise, commit the transaction and shut down SQL Server:

```
1> commit transaction
2> go
1> shutdown
```

- 2> go
- 5. Restart SQL Server, using your regular runserver file.

6. Execute the following command to truncate the inactive portion of the transaction log in *model*:

```
1> dump tran model with no_log
2> go
```

7. Reset the database options on *model* with the following command:

```
1> sp_dboption model, "no chkpt", false
2> go
1> sp_dboption model, single, false
2> go
1> use model
2> go
1> checkpoint
2> go
```

8. Disable updates to the system catalog and shut down SQL Server:

```
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
1> shutdown
2> go
```

9. Restart SQL Server.

#### Action for SQL Server Earlier Than 4.8

If the error occurred on a SQL Server release level earlier than 4.8, use the sections below to correct the error, depending on whether the error occurred at run time or during recovery. If the error occurred on SQL Server release 4.8 or later, see "Action for SQL Server Release 4.8 or Later" on page 2-122.

#### **Run-Time 1105 Errors**

If the error occurred during run time on a SQL Server earlier than 4.8, use the next section to correct the error. If it occurred during recovery, go to "Recovery 1105 Errors" on page 2-131.

Run-Time 1105 Errors: State 1

The data segment is full on the indicated database. To obtain more data space, do one or both of the following:

- Use alter database to increase the size of the data segment.
- Drop objects from the database or delete rows from tables.

You must delete at least 16K of data to realize any space gain.

#### Run-Time 1105 Errors: State 2

The log segment is full on the indicated database. To clear space in the log, perform the following steps:

1. Determine how many rows are in the *syslogs* table using the commands:

```
1> use database_name
2> go
1> select count(*) from syslogs
2> go
```

- 2. Dump the inactive portion of the transaction log using the dump transaction command. If this command fails with the 1105 error, retry the command using the with no\_log option.
- 3. Repeat step 1. If the number of rows in *syslogs* decreased significantly, proceed to step 4. However, if there is still a large number of rows in the *syslogs* table, an outstanding transaction is probably preventing the log from being cleared. See "Transaction Log Management" in the *SQL Server Troubleshooting Guide* for more information about the transaction log.

If there is still a large number of rows in the *syslogs* table, restart SQL Server and repeat step 2. When SQL Server starts and this database is recovered, the outstanding transaction is rolled back, allowing the log to be cleared by a subsequent dump transaction.

4. If the dump transaction command was executed with either the no\_log or the truncate\_only options in step 2, dump the database now, because these options prevent subsequent changes recorded in the log from being used to recover from a media failure. You must do a dump database to ensure the recoverability of subsequent database modifications.

#### ► Note

This database dump is not required if your environment does not save the transaction logs for media failure recovery.

Do not assume that the occurrence of 1105 errors automatically means your transaction log is too small. If the data and the log are on the same segment, either the State 1 and State 2, or the State 3 actions may free enough space without increasing the size of the transaction log.

If you are concerned that your transaction log is not large enough for your SQL Server, read "Transaction Log Management" in the *SQL Server Troubleshooting Guide* before deciding to increase the transaction log size. If appropriate, see the *SQL Server Reference Manual* for more information about the alter database command used to increase log size.

#### Recovery 1105 Errors

If the error occurred during recovery, use the procedure below that corresponds to the database on which the 1105 error occurred:

- A user database
- The *master* database
- The *model* database

#### Recovery 1105 Errors on a User Database

1. Manually set the database status to no chkpt on recovery and singleuser, as follows:

```
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
1> begin transaction
2> go
1> update master..sysdatabases
2> set status = status & ~256
3> where name = "database_name"
4> go
```

```
1> update master..sysdatabases
2> set status = status | 4112
3> where name = "database_name"
4> go
```

- 2. Check that each of the above update commands affected only one row. If more than one row was affected, issue a rollback transaction. Otherwise, commit the transaction and shut down SQL Server:
  - 1> commit transaction
    2> go
    1> shutdown
    2> go
- 3. Restart SQL Server.
- 4. Dump the transaction log with the no\_log option and reset the database status:

```
1> dump tran database_name with no_log
2> go
1> sp_dboption database_name, "no chkpt", false
2> go
1> sp_dboption database_name, single, false
2> go
1> use database_name
2> go
1> checkpoint
2> go
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
```

Recovery 1105 Errors on the master Database

If an 1105 error occurs on the *master* database during recovery, the problem may be corrected by the following procedure:

- 1. Create a new runserver file which includes the 3607 trace flag. This trace flag prevents recovery from running on any database, including *master*, and should only be used in this procedure. Refer to "How to Start SQL Server with Trace Flags" in the *SQL Server Troubleshooting Guide* for instructions.
- 2. Start SQL Server with the 3607 trace flag using the runserver file created in step 1. The recovery process will perform the

minimum actions required to allow access to each database, including *master*.

3. Once SQL Server has restarted and the *master* database is accessible, dump the transaction log with the **no\_log** option and shut down the Server:

```
1> dump tran master with no_log
2> go
1> shutdown with nowait
2> go
```

4. Restart SQL Server with the original runserver file. The database should recover normally. Now dump the *master* database:

```
1> dump database master
2> to logical_dump_device
3> go
```

Recovery 1105 Errors on the model Database

If an 1105 error occurs on the *model* database during recovery, SQL Server may not start. The error occurs because the *tempdb* database is required to start SQL Server, but cannot be built due to the problem with *model*. The following procedures may be used to correct this problem and restart SQL Server.

- 1. Create a new runserver file which includes the 3608 trace flag. This trace flag causes SQL Server to recover only the *master* database at start-up. Refer to "How to Start SQL Server with Trace Flags" in the *SQL Server Troubleshooting Guide* for instructions.
- 2. Start SQL Server with the 3608 trace flag using the runserver file created in step 1.
- 3. Manually set *model* status to no chkpt on recovery and single-user, as follows:

```
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
1> begin transaction
2> go
1> update master..sysdatabases
2> set status = status & ~256
3> where name = "database_name"
4> go
```

```
1> update master..sysdatabases
2> set status = status | 4112
3> where name = "database_name"
4> go
```

4. Check that the above update commands affected only one row. If more than one row was affected, issue a rollback transaction. Otherwise, commit the transaction and shut down SQL Server:

```
1> commit transaction
2> go
1> shutdown
2> go
```

- 5. Restart SQL Server, using your regular runserver file.
- 6. Execute the following command to truncate the inactive portion of the transaction log in *model*:

```
1> dump tran model with no_log
2> go
```

7. Reset the database option on model with the following command:

```
1> sp_dboption model, "no chkpt", false
2> go
1> sp_dboption model, single, false
2> go
1> use model
2> go
1> checkpoint
2> go
```

8. Disable updates to the system catalog and shut down SQL Server:

```
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
1> shutdown
2> go
```

9. Restart SQL Server.

#### Severity Level

21

#### **Error Message Text**

Cannot deallocate extent %ld, database %d. Object id %ld, index id %d, status %d in extent does not match object id %ld, index id %d, status %d in object being deallocated. Run DBCC CHECKALLOC.

#### Explanation

Error 1108 occurs when SQL Server is deallocating an extent, if the extent information on the allocation page does not match the information for the object being deallocated. This can be caused by corruption of the allocation page or memory corruption.

A software problem can cause this error to occur:

- During rollback of a transaction
- When rows for an object are being deleted or updated
- When an object is being dropped

This is a serious error as it indicates corruption has occurred.

#### Action

To recover from Error 1108, do the following:

- 1. Run dbcc checkalloc on the database named in the message. Resolve any errors reported by dbcc checkalloc using information about those errors in this manual.
- 2. Restart SQL Server.
- 3. To determine whether the problem still exists, run the following:

```
1> dbcc traceon (3604)
2> go
1> dbcc extentdump (database_ID, extent_number)
2> go
```

where *database\_ID* is the ID of the database reported in the error and *extent\_number* is the number of the extent reported in the error.

- 4. If the object ID and index ID displayed by the above command's output matches the object ID and index ID reported in the error message for the object being deallocated, try to do the task that reported the error again. If you get additional errors, restore the database from a clean backup or call Sybase Technical Support.
- 5. If the object IDs and index IDs do not match, restore the database from a clean backup or call Sybase Technical Support.

# **Additional Information**

If you need assistance from Sybase Technical Support, have the following information available when you call:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all error messages
- Operating system error log
- Output of dbcc checkalloc

#### **Severity Level**

21

#### Error Message Text

Extent chain for object %ld is not correctly linked.

#### Explanation

This error occurs when allocation page checks are performed as part of creating or dropping objects. 1117 errors may be raised as a secondary (and benign) side effect of the actual incorrect page linkage event. **dbcc** checking mechanisms may also report 1117 errors.

```
► Note
```

Error 1117 is only raised in SQL Server release 4.2 and earlier.

#### Action

Be aware of and avoid the following circumstances that can contribute to 1117 errors:

- Creating or dropping objects during periods of heavy system usage
- Having two or more dbcc sessions active at the same time
- Memory or disk problems that may have corrupted buffer cache values
- Stored procedures that do heavy, repetitive insertions and deletions between several tables
- Use of dump transaction with no\_log during heavy system use
- Using Ctrl-c immediately after issuing a command to drop an object
- Extensive manipulations of text fields

Use of sp\_who, sp\_lock, and analysis of the processes involved surrounding a 1117 event can help to pinpoint where the problem occurs.

Recovering from 1117 errors usually involves rebuilding the page chain for the object. This can be done in several ways. The most common solution is to bcp out, drop the table, re-create it, and bcp back in. dbcc checkalloc with the fix option can also correct some 1117 errors.

Sometimes using select into from the original table to a new table works. However, there may be references to the original table held within various system tables and you may see the same problems on the new table.

#### **Severity Level**

20

#### **Error Message Text**

```
Attempt to call pg_readalloc on page %ld which is not an allocation page.
```

#### Explanation

Allocation pages are used to track space in the database. Each allocation page for an object has an entry in the Object Allocation Map (OAM) for that table or index. The OAM entry for each allocation page stores the number of used and unused data/index pages on that allocation page.

Error 1120 occurs when you have an invalid page number for an allocation page in the allocation section of an OAM. Therefore, SQL Server will not have accurate information on what pages have been allocated.

This is a very serious error as SQL Server cannot correctly insert or update data when it occurs. Action should be taken immediately or further database corruption will occur.

#### Action

Use the page number in the error message to identify the object (and thus the database) that is affected by the error. Refer to "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* for details.

To recover from Error 1120, restore your database either using bcp or from clean backups. Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide* for information about using bcp to save your data.

#### Additional Information

For information about OAM pages, see the *System Administration Guide*.

If you need assistance from Sybase Technical Support, have the following information available when you call:

• SQL Server release and EBF Rollup level

- SQL Server error log
- Text of all error messages
- Operating system error log
- Output of dbcc page for the corrupted page
- Output of dbcc checkalloc for the database identified by dbcc page

#### Severity Level

22

#### Error Message Text

#### Release 10.0.2 and Later

The requested update to the OAM for object %ld, dbid %ld, index %d, oam page %ld, would cause a negative page count. allocation page: %ld, oamarray used: %ld, oamarray unused: %ld, deltaused: %ld, deltaunused: %ld, spid: %ld. Contact your System Admin.

#### Release 10.0.1 and Earlier

The requested update to the OAM for object %ld, index %d, oam page %ld, would cause a negative page count.

#### Explanation

Most tables and all indexes have at least one Object Allocation Map (OAM) page. A single OAM page holds information about up to 250 allocation pages (248 if the object is an identity column). Each allocation page for an object has an entry in the OAM pages for that table or index. The OAM entry for each allocation page stores the number of used and unused data and index pages on that allocation page.

When a new page is needed or a page is deallocated, the OAM entry has to be updated. The number of used and unused pages on the allocation page is not allowed to be less than 0. If the number becomes less than 0, Error 1129 occurs.

#### Action

1. If the object encountering the error is **not** a system table (a system table's object ID is less than 100), continue with step 2.

If the object is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption but it may be necessary to restore the database from clean backups.

2. For user tables, if the index ID is 0 or 255, continue with step 3.

If the index ID is **not** 0 or 255, translate it into an index name:

```
1> use database_name
2> go
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

- 3. If the index ID is 0 or 255, do one of the following:
  - Restore the database from clean backups.
  - Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

# **Additional Information**

For more information on OAM pages, see the *System Administration Guide* and the *SQL Server Reference Manual*.

If you need to call Sybase Technical Support, have the following information ready:

- SQL Server version and EBF Rollup level
- Text of all error messages
- SQL Server error log
- dbcc listoam(database\_name, object\_ID, index\_ID) output (you must have sybase\_ts\_role granted to you to run this and also run dbcc traceon (3604) first)

#### **Severity Level**

22

#### **Error Message Text**

```
Page %ld was expected to be an OAM page for %ld and it is not.
```

#### Explanation

Every Object Allocation Map (OAM) page has a unique number in the database. Error 1133 occurs when an attempt by SQL Server to retrieve an OAM page by specifying the page number fails because there is no OAM page in the current database having that page number.

Error 1133 can occur when checkalloc is run if what is expected to be the OAM page in the OAM page linkage is not an OAM page. The linkage is accessed via the *doampg* or *ioampg* column for the object in *sysindexes*. (*doampg* is the *sysindexes* pointer to the **data** pages' OAM. *ioampg* is the *sysindexes* pointer to the **index** pages' OAM.)

This error can also occur when, due to a problem with SQL Server, the OAM page number that is requested is greater than the maximum page number for the database.

In these cases, Error 1133 is a serious error and may result in reading the wrong OAM page into data cache and corrupting a buffer in the process.

Error 1133 can also be a result of trying to use an OAM-based system function across databases. For example, if you incorrectly type:

```
1> use mydb
2> go
1> select reserved_pgs (id,doampg) from
2> master..sysindexes
3> where id = object_id ("master..table")
4> go
```

you will get Error 1133 or 1142, because the system function attempted to retrieve an invalid OAM page for the current database (*mydb*).

### Action

Use the following steps to clear Error 1133:

- 1. Identify the object name and index name by translating the page number displayed in the error message into an object name and an index name using the procedure described in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide*.
- 2. If the corruption is on a user table's index (object ID is 100 or greater, and the index ID is between 1 and 250), dropping and re-creating the index should clear the corruption.

Use sp\_helpindex to list all indexes on the table and then rebuild all the nonclustered indexes using one of two methods:

- If the index ID is between 1 and 250, drop and re-create each nonclustered index on the table.

Or:

- If a clustered index also exists on the table, drop and re-create it, which causes all nonclustered indexes to be automatically rebuilt. If your table is large, you may not have the space to do this (a rule of thumb is that 150 percent of your table size must be available).
- 3. If the corruption is not on an index page, run dbcc tablealloc or dbcc checkalloc on the corrupted object. The output should report OAM allocation errors. Try to fix the reported allocation errors first by referring to the appropriate writeups in this manual.

If the 1133 errors persist, call Technical Support for assistance.

# **Additional Information**

For more information about OAM pages, see "Error 1129" on page 2-141 of this manual or see the *System Administration Guide*.

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all the error messages
- Operating system error log

- Output from dbcc page for the corrupted page
- Output of dbcc checkalloc for the database identified by dbcc page  $% \left( {{{\mathbf{D}}_{\mathbf{n}}}^{\mathbf{n}}} \right)$

# **Severity Level**

20

# **Error Message Text**

```
Failed to get lock on OAM page %ld. This should never happen.
```

#### Explanation

Each allocation page for an object has an entry in the Object Allocation Map (OAM) pages for that table or index. The OAM entry for each allocation page stores the number of used and unused data and index pages on that allocation page.

This error is raised when SQL Server tries to lock the starting page for an OAM but fails to get the address lock.

Error 1134 is due to a SQL Server problem.

### Action

Call Sybase Technical Support.

# **Additional Information**

Refer to the *System Administration Guide* for more information about OAM pages.

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all the error messages
- dbcc page output for the page named in the error message
- dbcc tablealloc output for the table identified when dbcc page was run

#### **Severity Level**

22

#### **Error Message Text**

```
Invalid OAM Page %ld. Found pstat=%d, objid=%ld,
indid=%d.
```

#### Explanation

Depending on what caused it, Error 1142 can be either a serious error due to actual corruption in the database or a result of misusing a system function (data\_pgs, reserved\_pgs, rowcnt, or used\_pgs). Error 1142 can occur during run time as well as during recovery (during SQL Server startup, load database, or load transaction).

#### **Incorrect System Function Use**

The error can occur when using any of the following system functions to retrieve information about allocation pages from other databases:

- data\_pgs
- reserved\_pgs
- rowcnt
- used\_pgs

Object Allocation Map (OAM)-based system functions cannot be used across databases. For example, if you incorrectly type:

```
1> use mydb
2> go
1> select reserved_pgs (id,doampg) from
2> master..sysindexes
3> where id = object_id ("master..table")
4> go
```

you will get Error 1142 or 1133, because the system function attempted to retrieve an invalid OAM page for the current database (*mydb*).

# **Incorrect OAM Page Entries**

If you get Error 1142 and you are sure that you did not use system functions across databases, this error is a sign of serious corruption in your database.

Every OAM page has a unique number in the database. Error 1142 means that an attempt by SQL Server to retrieve an OAM page by specifying the page number failed because there was no OAM page in the current database with that page number.

#### Action

Recovery from this error depends on why the error occurred. Follow the instructions below, depending on what caused the error.

# **Incorrect System Function Use**

Do not use any of the system functions specified in "Incorrect System Function Use" on page 2-147 across databases.

### Incorrect OAM Page Entry in sysindexes

In this case, Error 1142 is a serious error and may result in reading the wrong OAM page into data cache and corrupting a buffer in the process.

Use the following steps to clear Error 1142:

- 1. Identify the object name and index name by translating the page number displayed in the error message into an object name and an index name by using the procedure described in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide*.
- 2. If the corruption is on a user table's index (object ID is 100 or greater and index ID is between 1 and 250), dropping and re-creating the index should clear the corruption.

Use sp\_helpindex to list all indexes on the table, and then rebuild all the nonclustered indexes using one of two methods:

- If index ID is between 1 and 250, drop and re-create each nonclustered index on the table.

Or:

- If a clustered index also exists on the table, drop and re-create it, which causes all nonclustered indexes to be automatically rebuilt. If your table is large, you may not have the space to do this (a rule of thumb is that 150 percent of your table size must be available).

3. If the corruption is not on an index page, run dbcc tablealloc or dbcc checkalloc on the corrupted object. The output should report OAM allocation errors. Try to fix the reported allocation errors by referring to the information in this manual.

If the 1142 errors persist, call Technical Support for assistance.

### **Additional Information**

For more information on OAM pages, refer to "Error 1129" on page 2-141 and the *System Administration Guide*.

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all the error messages
- Operating system error log
- Output from dbcc page for the corrupted page
- Output from dbcc tablealloc or dbcc checkalloc for the object identified by dbcc page

#### **Severity Level**

22

#### **Error Message Text**

```
Invalid OAM Page %ld. Expected objid = %ld. Found
objid=%ld, pstat=%d, indid=%d.
```

#### Explanation

This error occurs when the status on the page is not for an Object Allocation Map (OAM) page or the page is an OAM page and the object ID on the page does **not** match the object that the OAM page supposedly belongs to.

Error 1143 is a serious error and may result in reading the wrong OAM page into data cache and corrupting a buffer in the process.

For more information about OAM pages, see "Error 1129" on page 2-141 or the *System Administration Guide*.

#### Action

Use the following steps to clear Error 1143:

- 1. Identify the object name and index name by translating the page number displayed in the error message into an object name and an index name by using the procedure described in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide*.
- 2. If the corruption is on a user table's index (object ID is 100 or greater and index ID is between 1 and 250), dropping and re-creating the index should clear the corruption.

Use sp\_helpindex to list all indexes on the table and then rebuild all the nonclustered indexes using one of two methods:

- If index ID is between 1 and 250, drop and re-create each nonclustered index on the table.

#### Or:

- If a clustered index also exists on the table, drop and re-create it, which causes all nonclustered indexes to be automatically rebuilt. If your table is large, you may not have the space to do this (a rule of thumb is that 150 percent of your table size must be available). 3. If the corruption is not on an index page, run dbcc tablealloc and/or dbcc checkalloc on the corrupted object. The output should report OAM allocation errors. Try to fix the reported errors first by referring to the appropriate writeups in this manual.

If the 1143 errors persist, call Technical Support for assistance.

### **Additional Information**

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all the error messages
- Operating system error log
- Output from **dbcc page** for the corrupted page
- Output from dbcc tablealloc or dbcc checkalloc for the object identified by dbcc page

# Lock Manager Errors

This section contains error messages pertaining to the Lock Manager.

#### **Severity Level**

20

### **Error Message Text**

```
Caller of lock manager is incorrectly trying to
unlock an unlocked object. spid=%d locktype=%d
dbid=%d object id or page number=%ld.
```

#### Explanation

Before the Lock Manager tries to unlock an object, it checks to make sure the object is in fact locked. When an object is not locked during this check, Error 1203 occurs.

This error is usually due to a SQL Server problem which causes an incorrect session descriptor pointer or logical page number to be passed to the Lock Manager.

Error 1203 can occur during an insert, update, or select, particularly on a table with an index.

## Action

Call Sybase Technical Support.

#### Additional Information

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Output of sp\_lock
- Text of all error messages

## Severity Level

19

## **Error Message Text**

#### Release 10.0 and Later

SQL Server has run out of LOCKS. Re-run your command when there are fewer active users, or contact a user with System Administrator (SA) role to reconfigure SQL Server with more LOCKS.

### Release 4.9.x and Earlier

SQL Server has run out of LOCKS. Re-run your command when there are fewer active users, or ask your System Administrator to reconfigure SQL Server with more LOCKS.

#### Explanation

This error occurs when SQL Server runs out of locks, as controlled by sp\_configure.

Following is an example of the output from sp\_configure locks, and a brief description of the output related to locks:

name	minimum	maximum	config_value	run_value
locks	5000	2147483647	0	5000

- The *minimum* and *maximum* column entries represent the minimum and maximum number of configurable locks for SQL Server.
- The *config\_value* column contains the value to which the *locks* configuration parameter has been set with *sp\_configure*. The default value of *config\_value*, which corresponds to the minimum value, is 0 (zero).
- The *run\_value* column contains the value SQL Server is using. The default run value in *sysconfigures* is 5000.

#### Action

Rerun the command when there are fewer active users, or ask the Sybase System Administrator ("sa") to reconfigure SQL Server for more locks. Increasing the number of available locks does impact performance and memory requirements. Refer to the table "Current memory requirements of some user-configurable parameters" in the SQL Server Troubleshooting Guide which shows how much overhead is required for each lock.

#### WARNING!

Before making any changes to SQL Server, see "SQL Server Memory Issues" in the SQL Server Troubleshooting Guide or the System Administration Guide sections on memory management for more information about how SQL Server uses memory resources.

To increase the number of locks available, complete the following steps:

- 1. Verify the number of locks currently configured for SQL Server:
  - 1> use master 2> go 1> sp\_configure locks 2> go

#### WARNING!

If you increase the number of locks available by too many locks, SQL Server may run out of memory and be unable to restart. See "Cannot Start SQL Server After Altering Configuration" in the *SQL Server Troubleshooting Guide* if you run out of memory and cannot restart SQL Server.

2. Choose the number of locks you want to configure and issue the following command:

```
1> sp_configure locks, number_of_locks
2> go
```

3. Initialize the change with the reconfigure command:

```
1> reconfigure
2> go
```

4. Restart SQL Server to operate with the new values.

# **Additional Information**

See the table "Memory requirements for different SQL Server releases" in the *SQL Server Troubleshooting Guide* for information about locks and other SQL Server resources.

Refer to the *SQL Server Reference Manual* for information about sp\_configure.

#### **Severity Level**

13

### **Error Message Text**

```
Your server command (process id #%d) was
deadlocked with another process and has been
chosen as deadlock victim. Re-run your command.
```

#### Explanation

This error occurs when a process tries to acquire a lock on an object that is locked by a second process when the second process is waiting for a lock on an object that has been locked by the first process. This situation is a deadlock, and can involve more than two processes.

SQL Server detects this situation, rolls back the transaction that has accumulated the least amount of CPU time, and notifies the application program of this action with Error 1205. This allows the other users' processes to move forward.

Deadlocks are caused by a number of situations, including:

- Transactions modify tables in different orders. There is a greater chance of deadlock between two transactions if one is processing in the sequence A B C while the other runs C B A.
- Transactions access tables via a nonclustered index. If the optimizer chooses a different nonclustered index for the same table for two different queries, a nonclustered index is not in the physical data sequence and the two processes are acquiring page locks in a random order. Thus, there is a greater chance that one process will lock a page that the other needs.
- Transactions that use the keyword holdlock or use the set isolation level command to hold shared locks. When holdlock is appended to a select transaction it holds the shared lock for the remainder of the transaction. This increases the risk of deadlock.
- Transactions that require a long time to run. The longer a transaction runs, the more likely it is that another user will require a resource held by the transaction.

## Action

Restart the transaction that has been rolled back.

To minimize future occurrences of deadlocks, use any of the following procedures that apply to your site.

# **Application Error Handling**

Each application should have deadlock handling routines. See the dbsetuserdata entry in the *Open Client DB-Library Reference Supplement* for a sample deadlock handling routine.

# **Use Well-Constructed Transactions**

Using transactions constructed to avoid deadlocks greatly reduces their occurrence. Some techniques for writing transactions that avoid deadlock include:

- Accessing tables in the same order in each transaction. Use coding conventions that require all transactions that access several tables to process them in the same order.
- Accessing tables via a clustered index when possible. If it is not possible to change a nonclustered index to a clustered index to minimize deadlocks, then trap the deadlock error in the application and provide appropriate recovery routines.
- Minimizing the use of holdlock. Use holdlock only when you require repeatable reads within a transaction or use the set isolation level command if ANSI conformity is not required.
- Avoiding long-running transactions. Some ways to avoid long-running transactions are:
  - Never allow user interaction within a transaction.
  - Separate logical units of work into transactions. For example, acquiring a sequential key from a key table for use in an insert statement can be separated into transactions similar to the following (in the first set of commands, you acquire the key; in the second set of commands, you use the key to do the insert):

```
1> declare @key int
2> begin transaction
3> update key_table set key = key + 1
4> select @key = key
5> commit transaction
6> go
```

```
1> insert mytable
2> values (@key, ...)
3> go
```

- Check whether client applications, including third party tools, allow users to accidentally create long running transactions.

# **Additional Information**

See "locks" in the *Transact SQL User's Guide* and the *System Administration Guide*.

## **Severity Level**

20

### **Error Message Text**

Locksleep called with bad process id %d

## Explanation

SQL Server may put a process to sleep while it is waiting for another process to free a lock. It does this by calling an internal locksleep routine. If SQL Server calls this routine with a process ID that does not exist or is incorrect, Error 1207 occurs. This usually happens after a process has aborted abnormally and has not freed all of its locks.

► Note

This error message occurs only in SQL Server release 4.2.

#### Action

Run sp\_who to see if the "process id" from the error message is a valid SQL Server process. Then run sp\_lock, giving the "process id," to see if that process holds locks. If sp\_lock shows a process ID (*spid*) that is not in sp\_who, shut down SQL Server as soon as possible, without doing a checkpoint, using shutdown with the nowait option.

If restarting does not clear the error condition, or if the problem happens regularly, contact Sybase Technical Support. When calling Technical Support, be prepared to fax or otherwise provide the error log as explained in "Reporting Errors" on page 1-2. Also, have sp\_who and sp\_lock output available.

#### **Severity Level**

20

### **Error Message Text**

```
A lock manager routine expected a lock record as an argument but was passed something else.
```

#### Explanation

This error occurs when the SQL Server Lock Manager is:

- Releasing a lock
- Transferring a lock
- Waiting for a lock

and the lock record structure is not valid.

This error is usually due to a SQL Server problem.

#### Action

Shutting down and restarting SQL Server will usually clear the 1265 error. If it does not clear the error or if the error occurs repeatedly, call Sybase Technical Support.

#### **Additional Information**

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Output of sp\_lock at the time the 1265 error occurred
- Text of all error messages

# Sort Manager Errors

This section contains error messages pertaining to the Sort Manager.

### **Severity Level**

20

# Error Message Text

Sort failure

# Explanation

This error occurs during index creation when SQL Server fails to complete a sort operation. There are a number of different states for this error, each indicating a different reason for the failure:

State	Meaning
2	When creating a high-level index structure, SQL Server is unable to build a parent node in the ancestor list.
3	When creating a high-level index structure, SQL Server is unable to build a parent node from the root of the tree.
4	When creating a high-level index structure, SQL Server is unable to build a parent node from the new root of the ancestor tree.
6	This state occurs when SQL Server is adding the results of an internal sort of a special leaflet node to an external merge in progress and a single internal merge step has a merge order greater than 8.
7	This state occurs when SQL Server is adding the results of an internal leaf merge node and the previous merge step caused the merge order to be greater than 8.
8	This state occurs when SQL Server is starting a non-sort of already sorted data to create a clustered index and it cannot find a page to an index that was supposed to have been built for the sorted data.
9	This state occurs when SQL Server is unable to allocate and initialize space for managing the sort's buffers. This state only occurs in SQL Server release 10.0 and later.
10	This state occurs when SQL Server is building indexes and the length of a row in an overflow page is either zero or greater than the size of a page.

When this error occurs, no index is created.

In SQL Server release 10.0 and later, the configuration parameters csortbufsize and csortpgcount control how SQL Server uses memory during sorting. csortbufsize controls the sort buffer size and

csortpgcount controls the sort page count. The default value for both variables is 0. If you have changed the values of these variables, it can cause Error 1501.

Error 1501 can also be caused by a SQL Server problem.

## Action

- 1. If you have not changed the values of csortbufsize or csortpgcount, call Sybase Technical Support.
- 2. To determine your current values for the csortbufsize and csortpgcount variables, shut down SQL Server and enter the following command (see below for syntax for operating systems other than UNIX):

```
% buildmaster -yall -dmaster_device | grep sort
```

3. To change the value of csortbufsize (use similar commands for csortpgcount), enter one of the following commands:

Operating System	Command	
UNIX	<pre>% buildmaster -ycsortbufsize=new_value -dmaster_device</pre>	
VMS	<pre>\$ buildmaster /alter="csortbufsize=new_value" /disk="master_device"</pre>	
OS/2, Microsoft Windows NT	bldmastr -ycsortbufsize= <i>new_value</i> -d <i>master_device</i>	
Novell NetWare	LOAD BLDMASTR -ycsortbufsize=new_value -dmaster_device	
Stratus VOS	buildmaster.pm -ycsortbufsize=new_value -dmaster_device	

- 4. Restart SQL Server and retry the index creation command that generated the 1501 error.
- 5. If you are unable to resolve this error by adjusting the csortbufsize and csortpgcount values (for example, setting them back to 0), call Sybase Technical Support.

# Additional Information

See create index in the SQL Server Reference Manual and the Transact-SQL User's Guide for information about creating indexes.

#### **Severity Level**

14

### **Error Message Text**

```
Create unique index aborted on duplicate key. Primary key is '%S_KEY'.
```

#### Explanation

This error occurs when you try to create a unique index on a column or columns containing at least two rows with the same index value. Each time a unique index is created, SQL Server checks for duplicate index values (if data already exists), and displays this message if it finds any.

When this error occurs, no index is created.

A command of the following type generates a 1505 error if there are duplicate values on *col1*, *col2*, and *col3*:

#### Action

Decide whether you want to allow or prevent duplicate rows in the table where the clustered index will be created. Use one of the following procedures:

- If you decide to allow duplicate rows, do not use the keyword unique when you create the index.
- If you decide to prevent duplicate rows, delete rows that have duplicate index values.

If you choose to delete duplicate rows, use the following procedure. For a brief period of time some rows from the table exist only in a temporary table. If a power failure or other interruption occurs, you could lose the temporary table and the rows in it. Therefore, before you begin this procedure, make a backup copy of the table or modify the procedure to use a table you create for this purpose instead of a temporary table. 1. Find the rows that caused the error:

```
1> select * from table_name
2> group by col1, col2, col3 having count(*) > 1
3> go
```

The query result contains all the rows causing the 1505 error.

2. Put one instance of each distinct row from step 1 into a temporary table:

```
1> select distinct * into #temp_table_name
2> from table_name
3> group by col1, col2, col3 having count(*) > 1
4> go
```

3. Some of the rows may contain identical values in the column or columns to be indexed and different values in other columns. Use the query below to find these rows:

```
1> select * from #temp_table_name
2> group by coll, col2, col3 having count(*) > 1
3> go
```

Examine the individual rows to decide which rows should be deleted. Delete these rows from the temporary table now. The rows you delete in this step will be deleted from the original table in step 4.

4. Delete all the rows which caused the 1505 error from the original table:

```
1> delete table_name from #temp_table_name
2> where table_name.col1 = #temp_table_name.col1
3> and table_name.col2 = #temp_table_name.col2
4> and table_name.col3 = #temp_table_name.col3
5> go
```

5. Each row in the temporary table should have unique values for the columns in question. Add these rows back into the original table:

```
1> insert into table_name
2> select * from #temp_table_name
3> go
```

6. Now you can create the unique index on the table.

# **Additional Information**

See the *SQL Server Reference Manual* and the *Transact-SQL User's Guide* entries for create index for more information about indexes and their options.

2-166

#### **Severity Level**

14

### **Error Message Text**

```
Create index aborted on duplicate rows. Primary key is 'S_KEY'.
```

#### Explanation

This error occurs when you try to create a clustered index on a table that contains duplicate rows.

#### Action

Decide whether you want to allow or prevent duplicate rows in the table where the clustered index will be created. Use one of the following procedures.

#### Allow Duplicate Rows

Create the index with the allow\_dup\_row option:

```
1> create clustered index index_name on table_name
2> with allow_dup_row
3> go
```

Subsequent update and insert commands can create duplicate rows after the allow\_dup\_row option is set.

**Prevent Duplicate Rows** 

Create the index with the ignore\_dup\_row option:

```
1> create clustered index index_name on table_name
2> with ignore_dup_row
3> go
```

Existing duplicate rows will be deleted from the table as the index is created. Subsequent attempts to enter a duplicate row with insert or update are ignored and the insert or update is cancelled with an informational message. If the insert or update is part of a larger transaction, the rest of the transaction will continue as usual.

► Note

**ignore\_dup\_row** and **allow\_dup\_row** are mutually exclusive. Using them together in the same **create index** statement causes Error 1916.

#### **Severity Level**

20

#### **Error Message Text**

Row compare failure

### Explanation

This error occurs when SQL Server compares two rows during a sort and the compare fails because a row was being compared against itself.

Error 1509 can occur during normal processing or when you try to create an index on a table and the operation fails because of data corruption or a SQL Server problem.

This is a serious error as it indicates possible corruption.

### Action

- 1. Run dbcc checktable and dbcc tablealloc on the table being accessed at the time of the error to check for corruption.
- 2. If there is no corruption, call Sybase Technical Support.
- 3. If there is corruption, refer to the writeups in this manual for other error messages that appear. Run dbcc checktable and dbcc tablealloc again to determine whether the corruption has been cleared.
- 4. If corruption still exists after eliminating all error messages you can resolve, copy your data out and back in using the procedure in "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide* or restore your database from clean backups using the procedure in step 5.

#### ♦ WARNING!

Some data might be lost on this page if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

- 5. Restore your database from clean backups:
  - 1. Drop the database. If the drop fails, follow the instructions in "How to Drop a Database When drop database Fails" in the *SQL Server Troubleshooting Guide*.
  - 2. Create a database for load. Make sure the database you create has sizes as least as large as those in *sysusages* for the original database (and that all other *sysusages* values match the original values). See the entry for create database for load in the *SQL Server Reference Manual* for releases 4.8 and later, or the entry for create database for releases earlier than 4.8.
  - 3. Load the database from backup. See the entry for load database in the *SQL Server Reference Manual*.

# Additional Information

If you need to call Sybase Technical Support, have the following documentation ready:

- SQL Server release and EBF Rollup level
- SQL Server error log output
- Text of all error messages
- dbcc checktable and dbcc tablealloc output
- sp\_configure output

#### **Severity Level**

17

#### **Error Message Text**

Sort failed: Out of space in database '%.\*s'

#### Explanation

SQL Server uses a clustered index to sort rows so that their physical order is the same as their logical (indexed) order. When creating a clustered index, SQL Server needs a considerable amount of space, 120 to 150 percent of the size of the table, because it needs to sort the data in the table first and then create the clustered index on the sorted data. This space should be available in the database or on the segment where you want to create your index.

Error 1510 occurs when you try to create an index and one of the following is true:

- There is not enough space in the database to create the index. Usually this error occurs with an 1105 error.
- No more locks are available.

Unfortunately, there is no straightforward way to identify which problem caused the error.

#### Action

Select the appropriate action from the sections below.

Not Enough Space in the Database

Use either of the following two strategies to correct this error:

- Use the instructions from "Error 1105" on page 2-120 to increase the space available in your database.
- Move the table where the index is to be created to a different, larger, user-defined segment, and then create the index. See the *System Administration Guide* for information about segments.
  - 1. Check that you have enough space available on the new segment by typing:

```
1> sp_helpsegment segment_name
2> go
```

2. Verify that the database where you want to create your index has enough space allocated on the device or devices that contain the segment from the above query:

```
1> sp_helpdb database_name
2> go
```

No More Locks Available

If the 1510 error occurs because SQL Server has no more locks available, use sp\_configure to increase the number of locks by changing the value of the configuration parameter locks:

```
1> sp_configure locks, new_number_of_locks
2> go
1> reconfigure
2> go
```

Restart SQL Server to initialize the new configuration parameter value.

See "How SQL Server Uses Memory" in the *SQL Server Troubleshooting Guide* to find out how memory is allocated for configuration parameters.

# **Additional Information**

See the create index entry in the SQL Server Reference Manual and the Transact-SQL User's Guide.

#### Severity Level

18

#### Error Message Text

## **Release 10.0 and Later**

```
Sort failed for table '%.*s' in database '%.*s'
because internal count of used pages is incorrect.
Run DBCC TABLEALLOC to correct the value.
```

#### Release 4.9.x and Earlier

Sort failed because dpages in the Sysindexes row for table '%.\*s' in database '%.\*s' had an incorrect value. Please run DBCC CHECKTABLE on this table to find the correct value, then contact Technical Support.

#### Explanation

This error occurs when a create index statement fails while sorting the table data. The create index statement allocates a work area based on an estimate in the system catalog of the number of data pages in the table. If this value is too low, the work area is too small and the sort fails.

SQL Servers releases earlier than 4.8 do not maintain an up-to-date record of the number of data pages in a table or index on disk because doing so would cause a significant performance decrease.

#### Action

Use one of the following **dbcc** commands, depending on your SQL Server release level:

SQL Server Release Level	dbcc Commands to Correct Error
4.9.1 and later	1> dbcc tablealloc ( <i>table_name</i> ) 2> go
4.8 and 4.9.0	<pre>1&gt; dbcc checkalloc (database_name) 2&gt; go</pre>

Table 2-5: Error 1520 correction commands

SQL Server Release Level	dbcc Commands to Correct Error
Earlier than 4.8	<pre>1&gt; dbcc checktable (table_name) 2&gt; go</pre>

Table 2-5: Error 1520 correction commands (continued)

The dbcc command updates the page count. You can now create the index with the create index command.

#### **Severity Level**

16

### **Error Message Text**

Create index with sorted\_data was aborted because of row out of order. Primary key of first out of order row is '%S\_KEY'

#### Explanation

The sorted\_data option speeds creation of an index when the data in the table is already in sorted order. The speed increase becomes significant on tables larger than 1GB.

Error 1530 occurs when you try to create an index with the sorted\_data option on a column or columns containing data that is not in sorted order.

When this error occurs, no index is created.

► Note

The sorted\_data option speeds index creation only for clustered indexes or unique nonclustered indexes. Creating a nonunique, nonclustered index with the sorted\_data option may succeed, but there is no improvement in performance.

#### Action

To eliminate the 1530 error, do one of the following:

- Sort the data (using your standard operating system procedures), bcp it into the table, and then create the index with the sorted\_data option.
- Create the index without the sorted\_data option.

## **Additional Information**

See create index in the *SQL Server Reference Manual* for more information about indexes and their options.

## **Severity Level**

16

## **Error Message Text**

```
The sorted_data option cannot be used for a nonclustered index if the keys are not unique within the table. Create index was aborted because of duplicate keys. Primary key is '%S_KEY'.
```

## Explanation

This error occurs when you try to create a nonunique, nonclustered index with the sorted\_data option on a column or columns containing rows with the same index value. If this situation exists, a statement of the following type will fail:

```
1> create index index_name
2> on table_name (col1, col2, col3)
3> with_sorted_data
4> go
```

When this error occurs, no index is created.

## Action

To eliminate the 1531 error, do one of the following:

- Do not use the sorted\_data option.
- Delete rows with duplicate index values.

**Delete Rows With Duplicate Index Values** 

If you decide to delete rows with duplicate index values, use the following procedure.

## ► Note

For a brief period of time some rows from the table will exist only in a temporary table, and if a power failure or other interruption occurs, you could lose the temporary table and the rows in it. So before you begin this procedure, you may want to make a backup copy of the table, or you can modify the procedure to use a table you create for this purpose, instead of a temporary table.

1. Find the rows that caused the 1531 error:

```
1> select * from table_name
2> group by coll, col2, col3
3> having count (*) > 1
4> go
```

2. Put one instance of each distinct row from step 1 into a temporary table:

```
1> select distinct * into #temp_table_name
2> from table_name
3> group by col1, col2, col3 having count(*) > 1
4> go
```

3. Some of the rows may contain identical values in the column or columns to be indexed and different values in other columns. Use the query below to find these rows:

```
1> select * from #temp_table_name
2> group by col1, col2, col3 having count(*) > 1
3> go
```

- 4. Examine the individual rows to decide which rows should be deleted. Delete these rows from the temporary table now. The rows you delete in this step will be deleted from the original table in step 5.
- 5. Delete all the rows which caused the 1531 error from the original table:

```
1> delete table_name from #temp_table_name
2> where table_name.col1 = #temp_table_name.col1
3> and table_name.col2 = #temp_table_name.col2
4> and table_name.col3 = #temp_table_name.col3
5> go
```

6. Now each row in the temporary table should have unique values for the columns in question. Add these rows back into the original table:

```
1> insert into table_name
2> select * from #temp_table_name
3> go
```

7. Create the index with the sorted\_data option.

## **Additional Information**

See create index in the *SQL Server Reference Manual* for more information about indexes and their options.

# **Initialization Errors**

This section includes codes for errors occurring during initialization routines.

#### **Severity Level**

21

## **Error Message Text**

#### Release 10.0 and Later

There are not enough 'user connections' available to start a new process. Retry when there are fewer active users, or ask your System Administrator to reconfigure SQL Server with more user connections.

#### Release 4.9.x and Earlier

No Pss structures available for new process

#### Explanation

There is one "Pss" (Process Slot Structure) for each configured user connection. Error 1601 occurs when all process slot structures are in use and a request is made to create another SQL Server process.

The occurrence of Error 1601 means that SQL Server is not configured with enough user connections to support all requests.

This message may also occur on operating systems communicating over DECnet if there are not enough logical links available.

#### Action

Perform the actions from the appropriate sections below.

Too Few User Connections (All Operating Systems)

Use the system procedure **sp\_configure** to increase the number of user connections:

1. Set the new value for user connections:

1> sp\_configure "user connections", new\_value
2> go

2. Instruct SQL Server to accept the values with reconfigure:

1> reconfigure
2> go

3. Restart SQL Server to activate the changes.

Depending on the exact configuration of your SQL Server, each user connection typically requires between 40K and 60K of memory (86K for AXP OpenVMS), whether it is in use or not. Gradually increase the number of user connections until these errors no longer occur. When you change the user connections configuration parameter, you may need to increase the memory configuration parameter. If you have increased the stack size or default network packet size configuration parameters, each user connection will require more memory. See "SQL Server Memory Issues" in the *SQL Server Troubleshooting Guide* for more details.

If you have an OpenVMS system, recalculate the quotas using the *SYBASE SQL Server Installation Guide* worksheet, and make any needed Sybase quota file and operating system adjustments. (Worksheets are in the *System Administration Guide Supplement* for release 4.9.1 and later, or in the *Installation Guide* for releases earlier than 4.9.1.)

For more information on determining the number of user connections your SQL Server needs, refer to the *System Administration Guide*.

```
► Note
```

This error often appears in the SQL Server error log without the 1601 error number: "kernel: no pss structure available for new process." Take the above action whether or not the error number is displayed.

**DECnet Protocol** 

Check that there are enough logical links available. See the *System Administration Guide Supplement for VMS* for more information about DECnet and logical links.

#### **Severity Level**

21

### **Error Message Text**

Unable to initialize network %d

### Explanation

This error occurs during startup when SQL Server is unable to initialize a network connection specified in your network addressing (*interfaces* or *sql.ini*) file.

#### ► Note

Throughout this writeup, your network addressing file is referred to as your "interfaces file".

Error 1602 can occur as a result of any of the following problems in your interfaces file:

- Duplicate entries
- Incorrect network address
- Other incorrect information such as incorrect host name or illegal port number
- For NT:
  - The WINSOC libraries or the protocol on which you want to run (that is, IPX/SPX) are not loaded.
  - You try to bring up the SQL Server on a bad default Server name (NT has default values).
- For NetWare, if the socket (port number) is already used or if the protocol is not loaded (TCP/IP is optional in NetWare)

Error 1602 can also occur as a result of changes in your network which make entries in your interfaces file unavailable to SQL Server.

# Action

Check your interfaces file to make sure everything is correct. If any information is incorrect or duplicate entries exist, use one of the following to make corrections to your interfaces file:

- sybinit (for SQL Server release 10.0 and later)
- sybtli (for TLI-based SQL Server releases 4.9.x and earlier)
- sqledit (for OS/2 SQL Servers release 10.0 and later)
- listini and writeini (for OS/2 and NT SQL Servers release 4.x)
- sqledit or sybinit (for NT SQL Servers release 10.0 and later)

## ► Note

For pre-10.0 systems that are not TLI-based (that is, systems that use sockets), edit the interfaces file manually.

If you do not find any incorrect information in your interfaces file, ask your network administrator to verify (at the operating system level) the entries in the interfaces file to make sure all network connections are working properly.

# **Additional Information**

Refer to the *System Administration Guide Supplement* for your platform for information about modifying your interfaces file.

If you need to call Sybase Technical Support, have your error log output and a copy of your interfaces file ready before you call.

#### **Severity Level**

21

### **Error Message Text**

Failed to open virtual socket for new connections

### Explanation

The SQL Server Network Handler listens on the ports specified in your network addressing (*interfaces* or *sql.ini*) file. When a connection comes in, the Network Handler opens another socket to redirect that client, clearing the listener port for the next connection, hence the term "virtual" socket. The connection appears, from the outside, to be connected to the interfaces port, when it is actually connected at a different address to which there is a pointer.

Error 1605 occurs when a failure occurs in that process of opening the new socket. There are a number of things that can cause this:

- Exceeding the number of configured user connections
- Exceeding the number of configured sockets
- Network configuration problems such as using the same IP address for two different clients

Error 1605 error can be raised at the following times:

- At startup time, when the network is initialized
- At creation of a process, when a new connection request comes in and SQL Server is unable to handle it
- At the forced termination of a process, when a new connection request comes in and SQL Server is unable to handle it

The information below includes a number of internal configuration block parameters. You will not be changing these directly but they are included to help explain how to calculate **sp\_configure** variables that relate to connection handling. Following are the kernel and configuration parameters used to calculate *cnsocket* (the maximum number of virtual sockets):

• *kimaxfd* – the maximum number of configured per process file descriptors that a process can use at a given time. The error log message "Using ## file descriptors" shows what this value is at boot time.

- remote sites (sp\_configure variable) the maximum number of remote sites.
- user connections (sp\_configure variable) the maximum number of user connections allowed.
- cnsocket the maximum number of virtual sockets (user connections + remote sites + 3).

During startup, if the value for *cnsocket* is greater than the value for *kimaxfd*, then the value of *cnsocket* will be set to the value of *kimaxfd* and the message:

kistartup: number of virtual sockets set to ###

is printed to the log where ### is the value of kimaxfd/cnsocket.

SQL Server recalculates the value for *cnsocket* each time you run the reconfigure command.

#### Action

Check the error above the 1605 error. If the error above the 1605 error is one of the following:

nopen: No virtual sockets available
nopen: no room in socket table

then the value for the Server configuration block parameter *cnsocket* has been exceeded. To solve the problem in this case, increase user or network connections and possibly increase the maximum number of file descriptors per process:

1. Determine the current value for *cnsocket*.

Use dbcc resource (this command requires sybase\_ts\_role on releases 10.0 and above and dbcc tracecon(3604) on all SQL Server releases) or buildmaster -yall (in OpenVMS, buildmaster/alter="all") to see the current configuration for sockets.

2. Determine the current value for the maximum number of network connections:

```
1> select @@max_connections
2> go
```

If the value of *cnsocket* is less than the value of *@@max\_connections*, there will be a problem.

3. Determine the current value for devices (the number of database devices that SQL Server can use, excluding dump devices):

```
1> sp_configure "devices"
2> go
```

- 4. Determine an appropriate value for @@max\_connections, based on the number of users expected to be logged on at any given time, and the number of devices. Remember that each device takes up a user connection. @@max\_connections is the kernel resource value *rmaxuconn* and is calculated using the following information:
  - ksnmstprts number of Server listener ports configured (in your interfaces file, additional Server listener ports are listed as clones of the "master" line for the SQL Server, one additional line for each interface you want SQL Server to listen on).
  - *rmaxuconn* maximum value for user connections. It is calculated by: @@max\_connections = kimaxfd - ksnmstprts - 4 (5 on OpenVMS). The pool of four is reserved for:

stdout error log console dump/load device (debug on OpenVMS)

5. Compare the appropriate value for sockets to the current value for sockets (as determined in step 1). (Remember that the maximum number of virtual sockets = user connections + remote sites + 3.) If the current value is too low, use sp\_configure to increase the value of user connections and/or Server remote sites so that the calculated value for sockets will be correct. This will probably correct the error. If the error still occurs, call Sybase Technical Support.

# ► Note

Under UNIX, you may need to raise the maximum number of file descriptors at the operating system level for the SQL Server if the new value for *cnsocket* exceeds the maximum number of per process file descriptors.

In an OpenVMS environment, If you increase the value for *user connections*, you may also need to increase the value of the *channelcnt* OpenVMS operating system parameter.

Refer to the *System Administration Guide Supplement for Microsoft Windows NT* for information about increasing the maximum number of Windows Sockets connections available on the Server PC.

# **Additional Information**

Refer to the *System Administration Guide* for information about configuration parameters. Refer to the *SQL Server Reference Manual* for information about sp\_configure.

Refer to the appropriate *System Administration Guide Supplement* for parameter information specific to your platform.

If you need to call Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Operating system version and error log
- Output of dbcc resource and sp\_configure
- Text of all error messages

#### Severity Level

21

#### Error Message Text

#### Release 10.0 and Later

```
A client process exited abnormally, or a network
error was encountered. Unless other errors
occurred, continue processing normally.
```

#### Release 4.9.x and Earlier

A network error was encountered while sending results to the front end. Check the SQL Server errorlog for more information.

#### Explanation

This error occurs when a client process stopped without informing SQL Server and a subsequent attempt by SQL Server to send results to this client fails.

Error 1608 is usually not serious and may be viewed as notification that a client process no longer exists. Some of the possible reasons a client process can disappear are:

- The client application was killed or ended abnormally.
- The client machine was powered off or was restarted.
- There is excessive network traffic.
- The network has been shut down.

## ► Note

Using Ctrl-c (on non-PC operating systems) to break out of an application should not cause this error.

#### Action

Do not be concerned if this error occurs only sporadically. However, if the error occurs frequently and continuously, or frequently for short periods of time, it may be a result of excessive network traffic or network problems. If you suspect network problems, have your network administrator run diagnostics on the network between the client and the SQL Server machines to isolate the problem.

### Severity Level

21

### **Error Message Text**

```
Could not close network %d connection for server process %d.
```

### Explanation

This error occurs when SQL Server is unable to close a network connection for a Server process.

Error 1613 can be caused by:

- System parameters being set too low
- Network problems

This is not a serious problem unless it occurs frequently.

### Action

Check the values for your system parameters that affect network use:

- Under OpenVMS, these might include *ASTLM*, *FILLM*, *BYTLM*, and *CHANNELCNT*.
- Under UNIX, these might include the limit on the maximum number of file descriptors (limits the number of sockets that can be open at one time) and the per process limit on file descriptors.

If you suspect network problems, have your network administrator run diagnostics on the network between the client and the SQL Server machines to isolate the problem.

### **Severity Level**

21

### Error Message Text

```
Warning: dataserver -p option fails. Please check for state '%d' to see where it fails.
```

### Explanation

This error occurs when SQL Server is unable to create a new password for a user with "sso" role in response to the dataserver command with the -p option.

There are three possible states:

State	Meaning
1	Login passed with the <b>-p</b> option does not exist in <i>syslogins</i> .
2	Login exists but does not have the "sso" (system security officer) role.
3	Call to the routine that sets the passwords failed. This routine generates its own errors to help you diagnose the problem.

### Action

The actions for recovering from Error 1623 are listed below, by state.

### State 1

• If you can, log into SQL Server as any user.

Determine which login names have "sso" role:

```
1> use master
2> go
1> select l.name, s.name from sysloginroles slr,
2> syslogins l,syssrvroles s
3> where slr.srid = 1 and slr.suid = l.suid and
4> slr.srid = s.srid
5> go
```

• If you cannot log into SQL Server as any user and you have a historical dump of the *master* database, create a new master device, load that dump and see whether you can log into that.

Refer to "Master Device Completely Lost" in the *SQL Server Troubleshooting Guide* for instructions on how to re-create your master device.

• If you cannot log into SQL Server as any user and do not have a historical dump of the *master* database, call Sybase Technical Support.

## State 2

• If you can, log into SQL Server as any user.

Determine which login names have "sso" role:

```
1> use master
2> go
1> select l.name, s.name from sysloginroles slr,
2> syslogins l,syssrvroles s
3> where slr.srid = 1 and slr.suid = l.suid and
4> slr.srid = s.srid
5> go
```

- If you can log into SQL Server but none of the users you can log in as have "sso" role, or if you cannot log into SQL Server as any user and you have a historical dump of the *master* database, create a new master device. If you have a dump, load that dump and see whether you can log into that. Refer to "Master Device Completely Lost" in the *SQL Server Troubleshooting Guide* for instructions on how to re-create your master device.
- If you cannot log into SQL Server as any user and do not have a historical dump of the *master* database, call Sybase Technical Support.

State 3

Other error messages will be listed. Follow the directions in this manual for recovering from those errors. If no directions exist, call Sybase Technical Support.

## **Create Utilities Errors**

This section contains error messages pertaining to create operations (table, database, and so on).

### **Severity Level**

16

### **Error Message Text**

```
Create table failed because column '%.*s' in table '%.*s' exceeds the maximum of 250 columns.
```

### Explanation

This error occurs when you try to create a table with more than 250 columns. The maximum number of definable columns per table is 250.

► Note

Column names must follow the rules for identifiers. They must be unique within a given table, but you can use the same column name in different tables in the same database.

### Action

Reduce the number of columns in the table you are trying to create to 250 or less. See the *SQL Server Reference Manual* entry for create table for more information.

#### **Severity Level**

21

#### Error Message Text

```
Crdb_disk: Getnext SCAN_NOINDEX on
sysdevices.status=DEFAULT failed to find default
rows
```

#### Explanation

This error occurs during execution of a create or alter database command when there is no disk device defined as the default disk for a SQL Server.

By default, the master device is defined as a default disk. If you have used the stored procedure sp\_diskdefault to turn off default for the master device:

1> sp\_diskdefault master, defaultoff
2> go

and you have not used **sp\_diskdefault** to specify another default disk device:

1> sp\_diskdefault dev\_name, defaulton
2> go

(where *dev\_name* is the new default disk device) then you will not have any devices defined as default devices and Error 1808 will occur when you run create or alter database.

#### Action

There are two methods for determining whether you have any disk devices defined as the default. Both methods are described below.

Method 1

Type:

1> sp\_helpdevice
2> go

If you see a line such as:

device\_name physical\_name description
\_\_\_\_\_\_master d\_master special,default disk,physical disk,17.00MB

where "default disk" is listed, then that device will be used by create or alter database when the on *dev\_name* clause is omitted for allocation for the database. If no entries have "default disk" in the description field, then you do not have any disk devices defined as default.

### Method 2

You can see whether you have any disk devices defined by checking whether the first bit of the *status* column in *sysdevices* is turned on for any devices:

```
1> select status, name from sysdevices
2> where status & 1 = 1
3> go
status name
3 master
```

To clear Error 1808:

• Use **sp\_diskdefault** to specify a device as the default disk device

or

• Always use the on *dev\_name* clause with create and alter database statements.

## **Severity Level**

14

## **Error Message Text**

## Release 10.0 and Later

CREATE DATABASE must be preceded by a 'USE master' command. Check with your DBO <or a user with System Administrator (SA) role> if you do not have permission to USE master.

## Release 4.9.x and Earlier

CREATE DATABASE must be preceded by a 'USE master' command. Check with your DBO (or System Administrator) if you do not have permission to USE master.

### Explanation

This error occurs when you try to create a new database without being in the *master* database.

### Action

Before issuing a create database command, be sure you are in master:

1> use master
2> go

If you do not have the necessary permissions to create a database, ask your Sybase System Administrator to grant you permission to use the create database command.

### **Severity Level**

16

### **Error Message Text**

CREATE DATABASE failed because of incorrect size parameter(s). Total number of megabytes specified must be at least %d megabytes so that the Model Database can be copied to the new database.

#### Explanation

When a create database command is issued, SQL Server makes a copy of the *model* database, which contains the system tables needed by each database. Error 1810 occurs when you try to create a database of a size smaller than the size of the *model* database. The default size of *model* is 2MB.

#### Action

Make sure the size you plan to use in your create database command is at least as large as the *model* database (combine the data space size and the log space size). To check this size in *model*, use this command:

1> sp\_helpdb model
2> go

### **Additional Information**

Refer to *model* in the *System Administration Guide* or create database in the *SQL Server Reference Manual* for more information.

## **Severity Level**

16

## **Error Message Text**

```
'%.*s' is the wrong type of device for CREATE or
ALTER database. Please check sysdevices. The
CREATE or ALTER is aborted.
```

## Explanation

This error occurs when a dump device is specified in a create or alter database command. You cannot create or alter a database on a dump device.

## Action

Change the command to specify a device with a valid type.

1. Obtain a list of all of the devices recognized by your SQL Server as follows:

```
1> sp_helpdevice
2> go
```

2. Check the *description* field of the sp\_helpdevice output to identify which devices have the correct type. Any device with a value in the *description* field other than "dump device" can be used.

## **Additional Information**

For further information on *devices* or *sysdevices* see "Managing Physical Resources" in the *System Administration Guide*.

### **Severity Level**

16

### **Error Message Text**

```
Cannot open new database '%.*s'. CREATE DATABASE is aborted.
```

### Explanation

During a create database command, SQL Server opens the new database at several stages, first after inserting the rows into the system catalogs and later to initialize the log segment. Error 1813 occurs when SQL Server is unable to open the new database or initialize the log segment for the new database during a create database command.

This error can be caused by the following:

- Uncleared allocation buffers. This can occur after a database has been dropped and then a create database command is issued.
- Corrupt system catalogs such as sysdatabases.
- A SQL Server problem.

### Action

If Error 1813 occurs just after a database has been dropped, shut down and restart SQL Server to clear the allocation buffers and then re-enter your create database command.

If this does not clear the problem or you have not just dropped a database, call Sybase Technical Support.

### **Additional Information**

Have the following information available when you call Sybase Technical Support:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all error messages
- dbcc checkcatalog output for master

## **Severity Level**

16

## **Error Message Text**

```
Cannot create more than one clustered index on table '%.*s'. Drop the existing clustered index '%.*s' before creating another
```

### Explanation

Only one clustered index can exist on a table, although many nonclustered indexes are allowed. SQL Server uses the clustered index to sort rows so that their physical order is the same as their logical (indexed) order. The bottom or leaf level of a clustered index contains the actual data pages of the table.

A nonclustered index does not keep the physical order of the rows the same as the indexed order. The leaf level of a nonclustered index contains pointers to rows on data pages. In other words, a nonclustered index has an extra level between the index structure and the data itself.

### Action

Use the sp\_helpindex command to examine the existing indexes on the table. If a clustered index already exists, it must be dropped before a different clustered index can be created.

In SQL Server release 10.0 and later, using the primary key clause in a create table statement creates a unique clustered index.

### **Severity Level**

16

### Error Message Text

```
%d is the maximum allowable size of an index.
Composite index specified is %d bytes.
```

### Explanation

This error occurs when the sum of the lengths of all the columns used in a composite index is 256 bytes or greater. (A composite index is any index that uses from 2 to 16 columns.)

The following is an example of a composite index using objects from the *pubs2* sample database:

```
1> create index example
2> on authors (phone, state, postalcode)
3> go
```

*phone* is defined as char(12), *state* as char(2), and *postalcode* as char(10). Since each *char* takes 1 byte of storage, the total length of this composite index is 24 bytes.

### Action

Examine the lengths of the columns in your composite index and make sure that the lengths total less than 256 bytes. See "Datatypes" in the *SQL Server Reference Manual* for information about how to calculate the storage size for different SQL Server datatypes.

Additional Information

See create index in the *SQL Server Reference Manual* for more information on indexes.

### **Severity Level**

16

## **Error Message Text**

Cannot specify more than %d column names for index key list. %d is specified.

### Explanation

This error occurs when more than 16 columns are combined in a single composite index.

### Action

Change your create index statement to include no more than 16 columns.

## **Additional Information**

Refer to create index in the *SQL Server Reference Manual* and composite indexes in the *Transact SQL User's Guide* for more information.

### **Severity Level**

16

### **Error Message Text**

```
CREATE INDEX options %s and %s are mutually exclusive.
```

### Explanation

This error occurs when you try to create an index with two mutually exclusive create index options. When this error occurs, no index is created.

These options are incompatible with each other:

- ignore\_dup\_row
- allow\_dup\_row
- ignore\_dup\_key

The following table shows which options can be used with which type of indexes:

Table 2-6: Compatibility chart for index types and index options

Tupo of Indox	Index Option			
Type of Index	ignore_dup_key	ignore_dup_row	allow_dup_row	
Nonunique Nonclustered	Don't use	Don't use	Don't use	
Nonunique Clustered	Don't use	Okay	Okay	
Unique Nonclustered	Okay	Don't use	Don't use	
Unique Clustered	Okay	Don't use	Don't use	

### Action

- Use only one of the options.
- Use options appropriate for the type of index being created.

## **Additional Information**

See create index in the *SQL Server Reference Manual* for more information about indexes and their options.

## **Character Set Conversion Errors**

This section contains error messages pertaining to character set conversion.

### **Severity Level**

11

## **Error Message Text**

```
Character set conversion is not available between client character set '%.*s' and server character set '%.*s'.
```

### Explanation

Character set conversion is the change of the encoding scheme of a set of characters on the way into or out of SQL Server. Conversion is used when SQL Server and a client communicating with it use different character sets. For example, if SQL Server uses character set ISO 8859-1 (iso\_1) and a client uses Code Page 850, character set conversion must be turned on so that both SQL Server and client interpret data passing back and forth in the same way.

Character set conversion is only available for certain character sets. Valid conversion matches are listed in the *System Administration Guide*. Error 2401 occurs when an invalid character set conversion is requested (for example, ascii\_8 with ISO 8859-1). When this error is raised, no conversion is performed.

Error 2401 can occur in the following situations:

- When a client requests a connection, SQL Server checks whether it can convert from the client's character set to its own character set. If it cannot do the requested conversion because conversion is not available between the two character sets, Error 2401 is raised.
- If you use the set char\_convert command to start conversion between the SQL Server character set and a different character set and conversion for those character sets is not available, Error 2401 is raised.

### Action

Refer to the table of valid conversion matches in the *System Administration Guide*. Make sure you are not requesting conversion involving an incompatible character set.

## **Additional Information**

Refer to the *System Administration Guide* for details about character set conversion.

## **Severity Level**

16

## **Error Message Text**

```
Error converting client characters into server's character set. Some character(s) could not be converted.
```

### Explanation

This error occurs during insertion of data (insert or bcp) when SQL Server fails to convert a character to the required character set.

Error 2402 usually occurs for one of the following reasons:

- The character exists in the client character set but it does not exist in the SQL Server character set.
- The character exists in both the client and the SQL Server character set, but is represented by a different number of bytes in the client character set than in the SQL Server character set.

This error occurs during normal processing and it prevents query execution.

### Action

The following options are available for recovering from Error 2402.

### **Change Your Data**

Modify the incoming data so that it contains characters recognizable by SQL Server.

## Turn Off Character Set Conversion

If the error occurs while you are using isql, bcp, or defncopy, you can use the -J (UNIX and PC) or /clientcharset (OpenVMS) command-line option with no character set name to set the client's character set to NULL. If you use this command-line option without specifying a character set name, no conversion takes place and no error message is sent. As a result, some characters sent by the client to the SQL Server may not be interpreted correctly by the SQL Server and vice versa. (If only 7-bit characters are being handled, no incorrect interpretation will take place.) Otherwise, you can turn off conversion so that characters are sent and received unchanged with the following command:

```
1> set char_convert off
2> go
```

Turn Off Character Set Conversion Error Reporting

You can turn off the printing of error messages with the following command:

1> set char\_convert on with no\_error
2> go

Bytes which cannot be converted are replaced with an ASCII question mark ("?").

## **Additional Information**

Refer to the *System Administration Guide* for details about character set conversion.

#### **Severity Level**

11

#### **Error Message Text**

```
Cannot find the requested character set in Syscharsets: name = '%.*s'.
```

### Explanation

Information about the character sets that are defined for use by SQL Server is stored in *syscharsets*. Error 2409 occurs when SQL Server cannot find a character set needed for character set conversion.

Error 2409 can occur in the following situations:

- When a client requests a connection, SQL Server checks whether it can convert from the client's character set to its own character set. If SQL Server cannot do the requested conversion because a character set is not found, Error 2409 is raised.
- If you use the set char\_convert command to start conversion between the SQL Server character set and a different character set and the character set you specify does not exist in *syscharsets*, Error 2409 is raised.

### Action

Make sure all necessary character sets are loaded, including the client's character set (as shown in the error message output):

If you are missing needed character sets, follow the directions in the *SQL Server Installation Guide* for your platform to load them. You will be directed to use one of the following utilities:

sybinit for SQL Server releases 10.0 and later

- sybconfig for SQL Server release 4.9.x
- sybinstall for SQL Server releases prior to 4.9.x

### **Standalone Utilities**

By default, for standalone utilities, the character set used for the client is the default for the platform. This may not be the correct character set. If the 2409 error is occurring while you are using isql, bcp, or defncopy, use the -J (UNIX and PC) or /clientcharset (OpenVMS) command-line option to specify the client character set.

## **Open Client Applications**

For Open Client applications such as DB-Library, you can specify the character set for the client through the login packet. Refer to the *Reference Manual* for the application and the *Open Client/Server Supplement* for your platform for details.

## **Additional Information**

Refer to the *System Administration Guide* for details about character set conversion.

## dbcc Errors

This section contains error messages pertaining to dbcc commands.

### **Severity Level**

16

### **Error Message Text**

Table named %.\*s not found; check sysobjects.

### Explanation

This error occurs when a dbcc command is run if SQL Server is unable to find a table name or ID in *sysobjects*.

Error 2501 can occur when running any of the following dbcc commands:

- dbcc checktable
- dbcc checkdb
- dbcc tablealloc (error occurs when ID is specified instead of name)
- dbcc reindex
- dbcc fix\_text

This error can be serious as it indicates possible corruption.

### Action

If this error occurs on a command for which you have specified a table name, check to make sure you have spelled the table name correctly:

1> select \* from sysobjects
2> go

If this error occurs because the entry for an object has been removed from *sysobjects* but the object is still referenced by some other system tables, call Sybase Technical Support. They will help you try to remove the references to that object from the other system tables.

## Additional Information

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Output of dbcc checkalloc and dbcc checkdb
- Text of all error messages

### **Severity Level**

16

### **Error Message Text**

```
Table Corrupt: A page is linked in more than one chain; check this page:page number=%ld allocation status=%d
```

#### Explanation

This error occurs when SQL Server detects an inconsistency in the page chains associated with an object. For example, a page may be connected in more than one chain. If this error is detected by a dbcc command, a 605 error may also occur.

### ♦ WARNING!

If Error 605 occurs with this error, use the instructions presented under "Error 605" on page 2-36 or call Technical Support.

This is a serious error and must be corrected immediately.

### Action

- 1. Use the procedure in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to determine which table is associated with the corrupt page chain.
- 2. If it is a system table (object ID is less than 100), call Sybase Technical Support immediately.
- 3. If it is **not** a system table, follow one of the procedures below to correct the problem, depending on what kind of object is involved:
  - If the page belongs to an index, drop and re-create the index.
  - If the page does not belong to an index, recover your database from a known clean backup or recover the tables not affected by this error via **bcp**. Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide* for details.

4. Examine your operating system error log file and the SYBASE error log to determine if hardware errors may have caused the corruption.

## **Additional Information**

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all error messages
- Output of dbcc page

### **Severity Level**

16

### **Error Message Text**

Table Corrupt: Page linkage is not consistent; check the following pages: (current page#=%ld; page# pointing to this page=%ld; previous page# indicated in this page=%ld)

### Explanation

This error occurs when SQL Server detects an inconsistency in the page linkage of one of the page chains associated with a table. There is one doubly linked page chain for the table's data as well as one for each index level. For example, a page's "next" pointer points to a page whose "previous" pointer points back to a different page. If dbcc commands detect this error during run time processing, a 605 error will also occur.

### WARNING!

If Error 605 occurs with this error, use the instructions presented under "Error 605" on page 2-36 or call Technical Support.

This is a serious error and must be corrected immediately.

#### Action

- 1. Use the procedure in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which object is associated with the "current page".
- 2. If the object is a system table (object ID is less than 100), call Sybase Technical Support immediately.
- 3. If the object is **not** a system table, perform one of the actions below, depending upon what kind of object was involved in the error.
- 4. Examine your operating system error log file and the SYBASE error log to determine if hardware errors may have caused the corruption.

## **Current Page Belongs to a Nonclustered Index**

If the "current page" belongs to a nonclustered index (index ID is between 2 and 250), drop and re-create the index.

## **Current Page Belongs to a Clustered Index**

If the "current page" belongs to a clustered index (index ID is 1), drop all the indexes on that table, re-create the clustered index first, and then re-create the remaining nonclustered indexes.

## **Current Page Does Not Belong to an Index**

If the "current page" does not belong to an index at all (index ID is 0 or 255) and there were no errors other than the 2503 error, perform the following steps:

- 1. Use bcp or select into to retrieve your data from the affected table. Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide* for instructions.
- 2. If errors still exist, restore from clean backups or call Sybase Technical Support.

## **Severity Level**

16

## **Error Message Text**

```
Table Corrupt: The values in adjust table should
be in ascending order starting from the end of the
table (page#=%ld row#=%d); check adjust table in
this row
```

### Explanation

An adjust table is stored at the end of every data or index row which has variable length columns. It is used to improve performance of SQL Server for rows with lengths greater than 256 bytes.

Error 2506 occurs when dbcc checkdb or dbcc checktable detects inconsistencies in the values stored in the adjust table of the row specified by the error message.

This error can occur on a data page or an index page.

### Action

If you have clean backups of your database, recovery from backups may be your best option. If you do not have clean backups of your database, follow these steps to correct the error:

- 1. Use the procedure in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond to the page number in the error message text.
- 2. If the object with the error is **not** a system table (a system table's object ID is less than 100), continue with step 3.

If the object with the error is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption but it may be necessary to restore from clean backups.

3. For user tables, if the index ID is 0 or 255, continue with step 4.

If the index ID is **not** 0 or 255, translate it into an index name:

```
1> use database_name
2> go
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

- 4. If the index ID is 0 or 255, do one of the following:
  - Restore the database from clean backups.
  - Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

## ♦ WARNING!

Some data on this page might be lost if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

Before dumping your database, make sure it is working correctly. More specifically, the following commands should be run prior to each dump:

- 1. dbcc checkdb.
- 2. dbcc checkalloc or dbcc checkalloc with the fix option. (See "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about running these commands in multi-user mode and how to prevent spurious allocation errors from dbcc commands.)

### Hardware Errors

Hardware errors on your database devices can cause Error 2506. To help rule out this possibility, do the following:

- Check the SQL Server error log to see if there are other indications of hardware problems, such as kernel messages reporting I/O errors.
- Check the hardware error log or diagnostics utilities for I/O errors.

If problems persist, refer to the error documentation in this manual or contact Sybase Technical Support.

## **Additional Information**

For more information, see bcp in the *SQL* Server Utilities Guide and select into in the *SQL* Server Reference Manual.

Have the following information available before calling Sybase Technical Support:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Operating system error log
- Output of dbcc checkdb and dbcc checkalloc

#### **Severity Level**

16

#### Error Message Text

```
Table Corrupt: Offset table is incorrect
(page#=%ld row#=%d offset in offset table is=%d
correct offset is=%d)
```

#### Explanation

An offset table is stored at the end of every data page and indicates where rows are located on that page. SQL Server uses the offset table to quickly access the rows in each page.

Error 2507 occurs when dbcc checkdb or dbcc checktable detects inconsistencies in the offset table of a row on the page specified by the error message.

### Action

- 1. Use the procedure from "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which object ID and object name correspond to the page number from the error message text.
- 2. If the object with the error is **not** a system table (a system table's object ID is less than 100), continue with step 3.

If the object with the error is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption, but it may be necessary to restore from clean backups.

3. For user tables, if the index ID is 0 or 255, continue with step 4.

If the index ID is **not** 0 or 255, translate it into an index name:

```
1> use database_name
2> go
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the **sp\_helpindex** procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

- 4. If the index ID is 0 or 255, do one of the following:
  - Restore the database from clean backups.
  - Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

## ♦ WARNING!

Some data on this page might be lost if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

Before dumping your database, make sure it is working correctly. More specifically, the following commands should be run prior to each dump:

- 1. dbcc checkdb.
- 2. dbcc checkalloc or dbcc checkalloc with the fix option. (See "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about running these commands in multi-user mode and how to prevent spurious allocation errors from dbcc commands.)

### Hardware Errors

In addition to the above possible causes, hardware errors on your database devices can cause Error 2507. To rule out this possibility, do the following:

- Check the SQL Server error log to see if there are other indications of hardware problems, such as kernel messages reporting I/O errors.
- Check the hardware error log or diagnostics utilities for I/O errors.

If problems persist, refer to the error documentation in this manual or contact Sybase Technical Support.

## Additional Information

For more information, see bcp in the *SQL Server Utilities Guide* and select into in the *SQL Server Reference Manual*.

Have the following information available before calling Sybase Technical Support.

- SQL Server release and EBF Rollup level
- SQL Server error log
- Operating system error log
- Text of all error messages
- Output from dbcc checkdb, dbcc checktable, and dbcc checkalloc

#### Severity Level

16

### **Error Message Text**

Table Corrupt: The row number and offset of each row in the page should have a matching entry in row number table; check this page (page#=%ld row#=%d offset in row number table=%d)

#### Explanation

The offsets for data or index rows are stored at the end of every page (in the row number table) and indicate where a certain row is located on that page.

Error 2509 occurs when the dbcc checkdb or dbcc checktable command detects that a row does not have an entry matching its offset (location) on the page in the row offset table.

Attempts to delete the offending row will result in Error 631, while attempts to select the offending row may be successful.

This problem is probably a result of a problem within SQL Server but may also be caused by one of the following:

- Hardware failure
- Sybase System Administration problems
- UNIX System Administration problems

#### Action

First, make sure that you ruled out any of the above-mentioned causes of this error by referring to the appropriate sections in "Encyclopedia of Tasks" in the *SQL Server Troubleshooting Guide*.

After you eliminated other more serious errors on this table, follow these steps to correct the 2509 error:

- 1. Follow the instructions in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond to the page number from the error message text. Also note the object ID.
- 2. If the object with the error is **not** a system table (object ID is more than 100), continue with step 3.

If the object with the error is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption or it may be necessary to restore from clean backups.

- 3. If the object with the error is a user table, use one of the following three methods to clear the 2509 error:
  - Create a clustered index on the corrupted table. Creating a clustered index will copy the whole table onto new data pages, and will overwrite the row number table on each page. If a clustered index already exists on the table identified in step 1, drop the clustered index and re-create it.

# ♦ WARNING!

If you have other serious errors on this table (Errors 614, 2506, 2507, or 2524), eliminate these errors first, before you create a clustered index. Refer to the documentation in this manual for how to eliminate other errors on this table.

- Select the whole table into a new table, and drop the old table.
- Bulk copy the affected table out, drop and re-create the table, and bulk copy back in.

# ◆ WARNING!

Some data on this page might be lost if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

For more information about how to copy a table in a new table or file, refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

## Additional Information

For more information, see create index and select into in the *SQL Server Reference Manual* and in the *Transact-SQL User's Guide*. Also, see bcp in the *SQL Server Utility Programs* manual.

## **Severity Level**

16

## **Error Message Text**

## Release 10.0.2 and Later

Key mismatch between index page and data page in database '%.\*s'. Drop and re-create the index. (index page %ld, row %d, data page %ld)

## Release 10.0.1 and Earlier

Table Corrupt: Keys do not match between leaf page %1d and data page %1d; check row # %d on the data page

## **Explanation**

This message indicates that an index is inconsistent with the table data that it represents.

## Action

Use the following procedure to recover from this error:

- 1. Record the index page number or the leaf page number (depending on which SQL Server release you are running) from the error text. See "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond index or leaf page number. Record the object ID and index ID.
- 2. If the object with the error is **not** a system table (its object ID is 100 or greater), go to step 3.

If the object with the error is a system table, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index. 3. If the object with the error is a user table, find the name of the index involved by executing the following query:

```
1> select name from sysindexes
2> where indid = index_id
3> and id = object_id
4> go
```

Drop and re-create the index. (See the *SQL Server Reference Manual* for information on drop index and create index.)

Run dbcc checktable and dbcc tablealloc on the affected table to verify that all problems have been corrected at the table level.

#### ► Note

If you are using a SQL Server release earlier than 4.9.1, the dbcc tablealloc command is not available and the dbcc checkalloc command should be used. Occasionally dbcc checkalloc reports this error when no real error condition exists. Run dbcc checkalloc in single-user mode if you suspect the 2510 error messages are incorrect.

If problems persist, refer to the error documentation in this manual or contact Sybase Technical Support. Have the output from the appropriate dbcc commands available when you call.

# **Severity Level**

16

# **Error Message Text**

# Release 10.0.2 and Later

```
Keys of index id %d for table '%.*s' in %S_MSG
should be in ascending order. Drop and re-create
the index. (index page %ld)
```

# Release 10.0.1 and Earlier

Table Corrupt: Keys in %S\_MSG should be in ascending order; check page number %ld

## Explanation

This error occurs when an index is not ordered correctly.

There are three possible values of "S\_MSG" from the error message text:

- Data page (only on clustered indexes)
- Index page
- Leaf page

# Action

Follow these steps to correct the error:

- 1. Follow the instructions in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond to the page number from the error message text.
- 2. If the object with the error is **not** a system table (object ID is more than 100), continue with step 3.

If the object with the error is a system table, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

3. If the object with the error is a user table, find the name of the index involved by executing the following query:

```
1> select name from sysindexes
2> where indid = index_ID
3> and id = object_ID
4> go
```

Drop and re-create the index. To verify that all problems have been resolved at the table level, run the dbcc checktable and dbcc tablealloc commands on the affected table.

```
► Note
```

If you are using a SQL Server release earlier than 4.9.1, the dbcc tablealloc command is not available and the dbcc checkalloc command should be used. Occasionally dbcc checkalloc reports this error when no real error condition exists. Run dbcc checkalloc in single-user mode if you suspect the 2511 error messages are incorrect.

If problems persist, refer to the error documentation in this manual or contact Sybase Technical Support. Have the output from the appropriate dbcc commands available for their review.

## **Severity Level**

16

## **Error Message Text**

```
Table Corrupt: Object id %ld (object name = %S_OBJID) does not match between %.*s and %.*s
```

## Explanation

This error occurs when an object exists in one system table but not in another system table where SQL Server expected to find it.

Error 2513 is displayed by dbcc checkcatalog. This is a serious error as it indicates corruption.

Error 2513 occurs with the following states:

State	Meaning
1	The object was found in <i>syscolumns</i> but not in either <i>sysobjects</i> or <i>systypes</i> .
2	The object (a view) was found in <i>sysobjects</i> but not in <i>syscolumns</i> .
3	The object (a view) was found in <i>sysobjects</i> but not in <i>syscomments</i> .
4	The object (a view) was found in sysobjects but not in sysprocedures.
5	The object (a stored procedure) was found in <i>sysobjects</i> but not in <i>syscomments</i> .
6	The object (a stored procedure) was found in <i>sysobjects</i> but not in <i>sysprocedures</i> .
7	The object (a rule or default) was found in <i>sysobjects</i> but not in <i>syscomments</i> .
8	The object (a rule or default) was found in <i>sysobjects</i> but not in <i>sysprocedures</i> .
9	The object (a table) was found in sysobjects but not in syscolumns.
10	The object (a table) was found in sysobjects but not in sysindexes.
11	The object (a table) was found in sysindexes but not in sysobjects.

#### Action

Call Sybase Technical Support when this error occurs. They may be able to help you delete the object that is causing the error. However, because other objects may reference that object, deleting it cleanly may be difficult. If this is the case, recovering from backups may be a better choice.

#### Severity Level

16

#### **Error Message Text**

```
Table Corrupt: Type id %ld (type name = %.*s) does
not match between %.*s and %.*s
```

#### Explanation

*syscolumns* contains a row for every column in a table, and *systypes* contains a row for every type in a table.

Error 2514 occurs when dbcc checkcatalog detects a type mismatch between the *systypes* and the *syscolumns* system tables. This situation is most often caused by a select into across databases when at least one column from the original table is a user-defined datatype.

#### Action

The 2514 error often prints out a *usertype* value that seems out of place: 0 or a large number. If your 2514 error does not match this description, call Technical Support for further assistance.

Use the following procedure to find the incorrect entry in *syscolumns* and replace it with the correct value. Note that "c" and "t" are actually typed in; they are not variables. This method allows SQL Server to compare all tables to search for the mismatch.

1. Identify the rows containing incorrect entries:

2> 3> 4> 5>	<pre>select c.name, c.usertyp from syscolumns c where not exists (select * from systypes where t.usertype = c.use go</pre>	t	
	name	usertype	type
	PartNumber	0	56

More than one row may be returned.

2. Search for the correct value for each incorrect *syscolumns usertype* by querying the *systypes usertype* column, using the value reported in *type*. Do this for each row reported in step 1 above:

```
1> select t.name, t.usertype, t.type
2> from systypes t
3> where t.type in ( 56 )
4> and t.usertype < 100</pre>
```

Results of the query should resemble the following:

name	usertype	type
int	7	56

The value returned under *usertype* is the correct value which should appear in the *usertype* column of *syscolumns*, 7 in this example.

- 3. Change the incorrect row value in the *systypes* column *usertype* with the correct value you obtained in step 2:
  - 1. Enable updates to system tables (substitute your site's values for "7," "0," "56" and "PartNumber" in this query):

```
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
1> begin transaction
2> update syscolumns set usertype = 7
3> where usertype = 0 and type = 56
4> and name = "PartNumber"
5> go
```

2. To commit the transaction, type:

```
1> commit transaction
2> go
```

3. Repeat the steps above for any other incorrect rows, substituting the required values for *usertype, type,* and *name* in the query.

4. When you have updated all the incorrect rows, disable updates to system tables:

```
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
1> checkpoint
2> go
```

4. Run dbcc checkcatalog again to verify that the problem is corrected.

#### **Severity Level**

16

#### **Error Message Text**

```
Table Corrupt: Procedure id %ld (procedure name = %S_OBJID) does not match between %.*s and %.*s
```

#### Explanation

This error occurs when an entry for a compiled object (for example, a stored procedure or trigger) exists in the table *sysprocedures* and that entry does **not** exist in the *sysobjects* table. This error is displayed by dbcc checkcatalog.

#### Action

1. If the error occurred on an object in *sysprocedures* in *master*, enable updates to system tables:

```
1> use master
2> go
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
```

2. Check to see how many rows contain the object entry:

```
1> use database_name
2> go
1> select * from sysprocedures
2> where id = procedure_id
3> go
```

where *database\_name* is the name of the database that contains the table with the object and *procedure\_id* is the ID of the procedure in the 2517 error.

3. If you need to recover the stored procedure, get the text of the stored procedure:

```
1> sp_helptext stored_proc_name
2> go
```

4. Remove the object entry from sysprocedures:

```
1> begin transaction
2> delete sysprocedures
3> where id = object_id
4> go
```

where *object\_id* is the ID of the procedure in the 2517 error.

5. Check the results carefully, and commit the transaction only if the update affected the expected number of rows. If it affected more than the expected number of rows, roll back the transaction. To commit the transaction, type:

```
1> commit transaction
2> go
```

6. Disable updates to system tables:

```
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
1> checkpoint
2> go
```

- 7. Run dbcc checkcatalog again to verify that the problem is corrected. If the object entry still exists in other tables, you may get other error messages (such as Error 2513). If you do get other error messages, follow the directions in this manual for handling those errors.
- 8. If you wanted the object that has been deleted, re-create it using the information obtained in step 3.

#### **Severity Level**

16

#### **Error Message Text**

```
Database Corrupt: The last checkpoint in
sysdatabases is incorrect. Syslogs recorded a
different checkpoint.
```

#### Explanation

This error is raised by dbcc checkcatalog when SQL Server determines that the last checkpoint record in the *syslogs* table is not in sync with the last checkpoint record for the database.

Error 2519 can occur in the following situations:

- When a dump transaction is being run at the same time as a dbcc checkcatalog on the same database.
- When SQL Server issues a checkpoint during a dbcc checkcatalog run.

This is **not** a serious error.

## Action

This is a transient error (will disappear when SQL Server is rebooted) and therefore no action is required.

To prevent the error from occurring in the future, run dbcc checkcatalog in single-user mode. If you do not run dbcc checkcatalog in single-user mode, make sure dump transaction is not run at the same time as dbcc checkcatalog and you will reduce the chances of this error occurring.

### **Severity Level**

16

### **Error Message Text**

Database named %.\*s not found; check sysdatabases

## Explanation

This error occurs when SQL Server is unable to find a database you specify when running dbcc checkcatalog, dbcc checkdb, or dbcc dbrepair.

Some reasons for this error are:

- You have mistyped the name of a database or the database you specified does not exist.
- Corrupt indexes exist on sysdatabases.

#### Action

Check to see whether the database you are specifying exists:

```
1> sp_helpdb
2> go
```

Search the output to see whether you have misspelled the database name or whether the *dbid* for the database you specified exists but the *name* has been removed or changed.

If the database you are specifying exists and you are getting a 2520 error, run dbcc checkdb on the *master* database to find out whether it is corrupt. If dbcc checkdb returns error messages, corruption has occurred. Whether or not you believe corruption has occurred, call Sybase Technical Support. You may be able to rebuild the indexes for *master* or you may have to restore *master* from a backup.

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- Output of dbcc checkdb
- Text of all error messages

#### Severity Level

16

### **Error Message Text**

#### Release 10.0.2 and Later

Table Corrupt: Page is linked but not allocated. Run DBCC TABLEALLOC to correct the problem. (alloc page#=%ld, extent id=%ld, logical page#=%ld, object id in extent=%ld, index id in extent=%ld, object name=%S\_OBJID)

## Release 10.0.1 and Earlier

Table Corrupt: Page is linked but not allocated; check the following pages and table: alloc page#=%ld extent id=%ld logical page#=%ld object id in extent=%ld (object name = %.\*s) index id in extent=%d

#### Explanation

This error is serious, especially if it occurs on a table's data pages. It means that a page is currently in use by a table or index but it has not been marked as allocated. The same page could be allocated again, resulting in a loss of whatever data resides on the page.

Pages encountering the 2521 error will not be included in a database dump. This is because database dumps are performed by reading allocation pages and not by traversing page chains. Therefore, this error should be corrected before dumping the database.

## ► Note

The instructions below are for fixing 2521 errors once they have occurred. Two easy-to-use strategies exist for **preventing** this error from occurring in the future. Refer to "Preventing Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about these strategies.

#### Action

Occasionally dbcc checkalloc reports this error when no real error condition exists. You can either check to see if these errors are real, or continue with this section and take action to correct them, whether or not they reflect a real allocation error.

Because the process used to determine whether the errors are real can be time-consuming, you may want to go now to the "Action" section appropriate for your release level:

- "Action for 10.0 SQL Server and Later" on page 2-242
- "Action for 4.9.1 SQL Server and Later" on page 2-242
- "Action for 4.9.0 SQL Server and Earlier" on page 2-245

## Verifying That the Error Is Real

Run dbcc checkalloc in single-user mode if you suspect the 2521 error messages are incorrect. If the error is in *master*, use the section "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide* for instructions about how to invoke SQL Server in single-user mode. See the *SQL Server Reference Manual* for information about dbcc checkalloc.

#### Action for 10.0 SQL Server and Later

If many of these errors are occurring, it is possible to clear them all at once by using the dbcc checkalloc and dbcc checkalloc with fix option commands. Refer to "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about using dbcc checkalloc.

#### Action for 4.9.1 SQL Server and Later

If the text of the error message includes a real object name, not a number, then the error is on an existing object to which the system catalog has correct references. Continue now to "Identify Table: User or System Table" on page 2-242.

If a number appears instead of the object name, then that object only partially exists and the error must be corrected using the procedure described in "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide*.

#### Identify Table: User or System Table

Look at the value for "object id in extent" in the error message. If it is 100 or greater, go to "Action for User Tables" on page 2-243. If the

"object id in extent" is below 100, it is a system table and requires a different procedure described in the section "Action for System Tables" on page 2-244.

## **Action for User Tables**

If the "object id in extent" in the error message is 100 or greater, follow these steps to correct the error:

- 1. Check the value of the "index id in extent" in the error message to determine whether it is a table (value is 0) or an index (with a value between 0 and 255).
- Run dbcc tablealloc or dbcc indexalloc, depending on whether the object named in the 2521 error message is a table or an index. Before you run the appropriate command, keep the following in mind:
  - dbcc tablealloc can correct this problem on a table or an index, but if the problem is on an index, you can avoid affecting the entire table by using the dbcc indexalloc command. If the table is large or heavily used, it may be more practical to use dbcc indexalloc.
  - These commands correct the error only when run in the full or optimized mode. Do not specify the nofix option, or the 2521 error will not be corrected.
  - You can use either the "object name" or "object id in extent" values from the error message in the commands above where the argument "object\_name" appears.

Use the command appropriate for your situation:

For Tables (index id in extent = 0)	For Indexes (0 < index id in extent < 255)
1> dbcc tablealloc ( <i>object_name</i> ) 2> go	<pre>1&gt; dbcc indexalloc (object_name, 2&gt; index_id_in_extent) 3&gt; go</pre>

Refer to the *SQL Server Reference Manual* and the *System Administration Guide* for additional details on the dbcc tablealloc and dbcc indexalloc commands.

## Action for System Tables

If the "object id in extent" in the error message is less than 100, follow these steps to correct the error:

- 1. Put the affected database in single-user mode:
  - If the database is *master*, use the procedure in "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide*, and then continue with this procedure.
  - If the database is not *master*, use the sp\_dboption stored procedure to put the affected database in single-user mode:

```
1> use master
2> go
1> sp_dboption database_name, single, true
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

- 2. Check the value of the "index id in extent" in the error message to determine whether it is a table (value is 0) or an index (value is greater than 0).
- 3. Run dbcc tablealloc or dbcc indexalloc, depending on whether the object named in the 2521 error message is a table or an index. Before you run the appropriate command, keep the following in mind:
  - dbcc tablealloc command above can correct either a table or an index, but if the problem is on an index, you can avoid affecting the entire table by using the dbcc indexalloc command. If you need to minimize the amount of time the table is unavailable, it may be more practical to use dbcc indexalloc.
  - These commands correct the error only when run in the full or optimized mode, with the fix option specified. This is necessary because the default value is nofix when these commands are run on system tables.
  - You can use either the "object name in extent" or "object id" values from the error message in the commands above where the argument "object\_name" appears.

For Tables (index id in extent = 0)	For Indexes (0 < index id in extent < 255)
1> dbcc tablealloc ( <i>object_name</i> ,	1> dbcc indexalloc ( <i>object_name</i> ,
<pre>2&gt; full, fix)</pre>	<pre>2&gt; index_id_in_extent, full, fix)</pre>
3> au	3> a0

## Use the command appropriate for your situation:

- 4. Turn off single-user mode in the database:
  - If the database is *master*, use the procedure in "Returning SQL Server to Multi-User Mode" in the *SQL Server Troubleshooting Guide*.
  - If the database is not *master*, use the following procedure:

```
1> use master
2> go
1> sp_dboption database_name, single, false
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

Refer to the *SQL Server Reference Manual* and the *System Administration Guide* for additional details on the dbcc tablealloc and dbcc indexalloc commands.

## Action for 4.9.0 SQL Server and Earlier

If the "object id in extent" is less than 100, the error is on a system table, which cannot easily be corrected. Call Technical Support for assistance.

If the "object id in extent" is 100 or greater, then use the dbcc page command described in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to display information about the logical page from the error message. The value that dbcc page displays for "indid" should match the "index id in extent" printed at the end of the message.

If these do not match, contact Sybase Technical Support. If they do match, see what kind of index is involved. Refer to the table of index IDs contained in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide*.

For Tables (index id in extent = 0)

Contact Sybase Technical Support immediately. Before calling Technical Support, have the complete text of all the error messages available to fax or otherwise provide to Sybase Technical Support.

For Indexes (0 < index id in extent < 255)

The problem can be resolved by dropping and re-creating the index in question. Once the index is rebuilt, run dbcc checkalloc to verify that the problem no longer exists. If it persists, call Sybase Technical Support.

### **Severity Level**

16

## **Error Message Text**

Table Corrupt: Row length is inconsistent between the computed row length and the recorded row length on page; check the following page and row: pageno=%ld row#=%d computed row length=%d row length on page=%ld

#### Explanation

SQL Server computes the length of each row by using the values stored in the offset and adjust tables of each row. Errors 2524 occurs when dbcc checkdb or dbcc checktable detects that the actual length of a row does not match the computed length of that row.

This error can occur on a data page as well as on an index page.

#### Action

If you have clean backups of your database, recover from backups. If you do not have clean backups of your database, follow these steps to correct the error:

- 1. Use the procedure in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond to the page number from the error message text.
- 2. If the object with the error is **not** a system table (a system table's object ID is less than 100), continue with step 3.

If the object with the error is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption but it may be necessary to restore from clean backups.

3. For user tables, if the index ID is 0 or 255, continue with step 4.

If the index ID is **not** 0 or 255, translate it into an index name:

```
1> use database_name
2> go
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

- 4. If the index ID is 0 or 255, do one of the following:
  - Restore the database from clean backups.
  - Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

# ♦ WARNING!

Some data on this page might be lost if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

Before dumping your database, make sure it is working correctly by running the following commands prior to each dump:

- 1. dbcc checkdb.
- 2. dbcc checkalloc or dbcc checkalloc with the fix option. (See "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about running these commands in multi-user mode and how to prevent spurious allocation errors from dbcc commands.)

## **Hardware Errors**

This problem is probably a result of a problem within SQL Server but may also be caused by hardware errors on your database devices. To help rule out this possibility, do the following:

- Check the SQL Server error log to see if there are other indications of hardware problems, such as kernel messages reporting  $\rm I/O$  errors.
- Check the hardware error log or diagnostics utilities for I/O errors.

If problems persist, refer to the error documentation in this manual or contact Sybase Technical Support.

## **Additional Information**

For more information, see bcp in the *SQL* Server Utility Programs manual and select into in the *SQL* Server Reference Manual.

Have the following information available before you call Sybase Technical Support:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Operating system error log
- Output of dbcc checkdb and dbcc checkalloc
- Text of all error messages

## **Severity Level**

16

## **Error Message Text**

```
Table Corrupt: Object id wrong; tables: alloc page
%ld extent id=%ld l page#=%ld objid in ext=%ld
(name = %S_OBJID) objid in page=%ld (name =
%S_OBJID) objid in sysindexes=%ld (name = %S_OBJID)
```

#### Explanation

This error occurs when dbcc checkalloc detects one of the following errors:

- The object ID on a page does not match the object ID on the allocation structure (extent) associated with that page. This is the case when the "objid in ext" matches the "objid in page" from the error message.
- The object ID on a page does not match the object ID in the system table *sysindexes*. This is the case when the "objid in ext" does not match the "objid in sysindexes" from the error message.

#### Action

Compare the "objid in ext" to "objid in page" from the error message. If they do not match, go to "Object IDs Do Not Match" on page 2-250. If they are equal, go to "Object IDs Match" on page 2-251.

#### WARNING!

If you drop an object before you correct the 2525 error condition, the drop operation could cause pages belonging to another object to be deallocated. This can cause 1108 errors.

**Object IDs Do Not Match** 

There are two possible situations in this scenario. Either the object exists and may be corrupted or it no longer exists. To determine which scenario you have, do the following:

```
1> use database_name
2> go
```

```
1> select object_name(objid_in_extent)
2> go
```

where *database\_name* is the database where the corruption is reported and *objid\_in\_extent* is from the error message.

If the result of the query is a table that currently exists or if either object (object ID on the page or the object ID of the allocation structure associated with that page) is a system table (object ID is less than 100), call Sybase Technical Support. If you have dial-in facilities, Sybase Technical Support may be able to repair this corruption without losing any data. However, restoring from known, clean backups may be necessary.

If the query does not return any rows, this means the object that is marked as owning the extent no longer exists.

If the table is a user table, you may be able to correct this error. Back up the data in the table either by using **bcp** to copy out the table whose object ID appears in "objid in page" or by selecting it into a new table.

Once a copy of the table has been made, you can do one of the following:

- 1. Create a clustered index on the table. If you already have a clustered index, drop it and then re-create it.
- 2. Drop the corrupt table, re-create it, and move the data back into the table using a bcp in, an insert with a subselect from the holding table, or select into. Alternatively, the holding table itself can be renamed once the original table has been dropped.

#### **Object IDs Match**

There are two possible situations in this scenario. The pointer to a distribution page for a particular index may be invalid. Or, there may be a mismatch between the index ID on the allocation page and the index ID on the data page.

 Use the following commands to determine which index has the problem. This query compares the "id" column in the output to the "objid in sysindexes" from the error message, and the "distribution" column to the "l page #" from the error message.

```
1> use database_name
2> go
```

```
1> select indid, name, object_name(id)
2> from sysindexes
3> where id = objid_in_sysindexes and
4> distribution = l_page_#
5> go
```

If this query does not return any rows, contact Sybase Technical Support.

- 2. If the query does return a row, drop and re-create the index specified by this query to correct the problem. (See the *SQL Server Reference Manual* for more information about drop index and create index.)
- 3. Execute dbcc tablealloc to make sure that the problem is corrected. If the problem still exists, contact Sybase Technical Support.
- ► Note

If you are using a SQL Server release earlier than 4.9.1, the **dbcc tablealloc** command is not available and the **dbcc checkalloc** command should be used.

#### **Severity Level**

16

#### **Error Message Text**

```
Incorrect DBCC command: Please see the Commands Reference Manual for DBCC commands and options.
```

#### Explanation

The following situations can cause this error:

- An attempt was made to execute an invalid dbcc command.
- An incorrect number of parameters has been passed to a dbcc command.
- Some other syntax error was made on a dbcc command.

#### Action

Make sure you are using the correct name and syntax for dbcc commands. To determine the correct syntax, refer to the *SQL Server Reference Manual* and the *System Administration Guide*.

**Severity Level** 

16

**Error Message Text** 

Release 10.0.2 and Later

Table Corrupt: Attempted to get page %ld, object %ld; got page %ld, object %ld.

## Release 10.0.1 and Earlier

Table Corrupt: Page %ld not found.

## **Explanation**

This error occurs when SQL Server detects an inconsistency between the location of a page on disk and the page number stored in the page header. In the best case, only the value of the page number is wrong. In the worst case, the entire page is corrupt.

This is a serious error and must be corrected immediately.

If the page number in the error is evenly divisible by 256, the error has occurred on an allocation page. All objects referred to by that allocation unit (that page and the 255 pages following it) may be affected (up to 32 objects). Otherwise, the error occurred on a system or user table.

# ♦ WARNING!

Attempts to drop an object with a 2529 error may result in 821 and 813 errors and additional database corruption.

## Action

- 1. For releases 10.0.1 and earlier, refer to "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to determine the object affected.
- 2. Run dbcc checkalloc and dbcc checkdb on the database affected to determine the full extent of the corruption.
- 3. The 2529 error often results from hardware errors, or system administration problems such as inadvertent use of a raw

partition for two separate purposes. Check on and correct any such problems before attempting to restore the database:

- Refer to "Correct Use of Raw Partitions" in the *SQL Server Troubleshooting Guide* for information about using raw partitions.
- Check the SQL Server error log to see if there are other indications of hardware problems, such as kernel messages reporting I/O errors.
- Check the hardware error log or diagnostics utilities for I/O errors.
- 4. If the 2529 error is on a system table (object ID is less than 100), call Sybase Technical Support.
- 5. If the 2529 error is on a user table (object ID is greater than or equal to 100), you may be able to select the data into another table and drop the original. Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide* for instructions.

After you drop the original table, you may get an 821 error. In this case, shut down SQL Server and restart it. Try a select from the old corrupt table to make sure it is gone, then run dbcc checkalloc and dbcc checkdb to confirm that there is no more corruption.

In some cases, a 2529 error is first reported as a 2503 error, and becomes a 2529 error after SQL Server is restarted. Although the procedure described above may clear the 2529 error, it will not remove the main cause of your corruption.

# Additional Information

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Operating system error log
- Output of dbcc checkalloc
- Text of all the error messages

# **Severity Level**

16

# **Error Message Text**

```
Table Corrupt: Page is allocated but not linked;
check the following pages and ids: allocation
pg#=%ld extent id=%ld logical pg#=%ld object id on
extent=%ld (object name = %S_OBJID) indid on
extent=%ld
```

# Explanation

This error occurs when a page is marked as allocated to an object, although it is not being used. The page has not been used and no data loss is associated with this error. However, one page of disk space may not be used until the error is corrected: each 2540 error means the loss of one blank data page. The table data is not corrupt.

A few 2540 errors are no cause for concern. However, if many of these errors occur, the amount of "lost" disk space could be significant.

► Note

The instructions below are for fixing 2540 errors once they have occurred. Two easy-to-use strategies exist for **preventing** this error from occurring in the future. Refer to "Preventing Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about these strategies.

# Action for All SQL Server Versions

Occasionally dbcc checkalloc reports this error when no real error condition exists. You can either check to see if the error is real, or continue with this section and take action to correct it, whether or not it reflects a real allocation error.

Because the process used to discover whether or not the error is real can be time-consuming, you may want to go now to the "Action" section appropriate for your release level:

• "Action for 10.0 SQL Server and Later" on page 2-257

- "Action for 4.9.1 SQL Server and Later" on page 2-257
- "Action for 4.9.0 SQL Server and Earlier" on page 2-260

# Verifying That the Error Is Real

Run dbcc checkalloc in single-user mode if you suspect the 2540 error messages are incorrect. If the error is in *master*, use the section "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide* for instructions about how to invoke SQL Server in single-user mode. See the *SQL Server Reference Manual* for information about dbcc checkalloc.

# Action for 10.0 SQL Server and Later

If many of these errors are occurring, it is possible to clear them all at once by using the dbcc checkalloc and dbcc checkalloc with fix option commands. Refer to "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about using dbcc checkalloc.

## Action for 4.9.1 SQL Server and Later

If the text of the error message includes a real object name, not a number, then the error is on an existing object which the system catalog has correct references to, and you should continue now to "Identify Table: User or System Table".

If a number appears instead of the object name, then that object only partially exists and the error must be corrected using the procedure described in "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide*.

# Identify Table: User or System Table

Look at the value for "object id on extent" in the error message. If it is 100 or greater, continue with the next section, "Action for User Tables." If the "object id on extent" is below 100, it is a system table and requires a different procedure as described in the section "Action for System Tables" on page 2-258.

# **Action for User Tables**

If the "object id on extent" in the error message is 100 or greater, follow these steps to correct the error:

- 1. Check the value of the "indid on extent" in the error message to determine whether it is a table (value = 0) or an index (value > 0).
- 2. Run dbcc tablealloc or dbcc indexalloc, depending on whether the object named in the 2540 error message is a table or an index. Before you run the appropriate command, keep the following in mind:
  - dbcc tablealloc corrects this problem on a table or an index, but if the problem is on an index, you can avoid affecting the entire table by using dbcc indexalloc. If the table is large or heavily used, it may be more practical to use dbcc indexalloc.
  - These commands can correct the error only when run in the full or optimized mode, and with the nofix option not specified, the default for user tables.
  - You can use the object name or object ID in the following commands where the argument "object\_name" appears.

Use the command appropriate for your situation:

For Tables (index id in extent = 0)	For Indexes (0 < index id in extent < 255)
<pre>1&gt; dbcc tablealloc (object_name) 2&gt; go</pre>	<pre>1&gt; dbcc indexalloc (object_name, 2&gt; indid_on_extent) 3&gt; go</pre>

See the *SQL Server Reference Manual* and *System Administration Guide* for additional details on dbcc tablealloc and dbcc indexalloc.

# **Action for System Tables**

If the "object id on extent" in the error message is less than 100, follow these steps to correct the error:

- 1. Put the affected database in single-user mode:
  - If the database is *master*, use the procedure in "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide*, and then go to step 2.

- If the database is not *master*, use the sp\_dboption stored procedure to put the affected database in single-user mode:

```
1> use master
2> go
1> sp_dboption database_name, single, true
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

- 2. Check the value of the "indid on extent" in the error message to determine whether it is a table (value is 0) or an index (value is greater than 0).
- 3. Run dbcc tablealloc or dbcc indexalloc, depending on whether the object named in the 2540 error message is a table or an index. Then execute the appropriate command. Before you run the appropriate command, keep the following in mind:
  - dbcc tablealloc corrects either a table or an index, but if the problem is on an index, you can avoid affecting the entire table by using dbcc indexalloc. If you need to minimize the amount of time the table is unavailable, it may be most practical to use dbcc indexalloc.
  - These commands correct the error only when run in the full or optimized mode, with the fix option specified, because the default value is nofix on system tables.
  - You can use the "object name" or "object id on extent" in the commands above where the argument *object\_name* appears.

For Tables (index id in extent = 0)For Indexes (0 < index id in extent < 255)</th>1> dbcc tablealloc (object\_name,<br/>2> full, fix)1> dbcc indexalloc (object\_name,<br/>2> indid\_on\_extent, full, fix)<br/>3> go

Use the command appropriate for your situation:

- 4. Turn off single-user mode in the database:
  - If the database is *master*, see "Returning SQL Server to Multi-User Mode" in the *SQL Server Troubleshooting Guide*.

- If the database is not *master*, use the following procedure:
  - 1> use master
    2> go
    1> sp\_dboption database\_name, single, false
    2> go
    1> use database\_name
    2> go
    1> checkpoint
    2> go

Refer to the *SQL Server Reference Manual* and *System Administration Guide* for additional details on dbcc tablealloc and dbcc indexalloc.

# Action for 4.9.0 SQL Server and Earlier

Use either of the following methods to correct 2540 errors:

- Correct the error yourself by recovering the database from a known clean backup, using your usual SYBASE system administration procedures. (See the *System Administration Guide* for more information.)
- Sybase Technical Support may be able to correct the errors by dialing into your system and accessing the problem database. The database must be in read-only mode while Technical Support is working on it.

## **Severity Level**

16

## **Error Message Text**

```
Table Corrupt: object id does not match between
extent in allocation page and sysindexes; check
the following extent: alloc pg#=%ld extent#=%ld
object id on extent=%ld (object name = %.*s)
object id in sysindexes=%ld (object name = %.*s)
```

### Explanation

This error occurs when dbcc checkalloc detects a mismatch in the object ID between an allocation structure (an extent) and the system catalog (*sysindexes*).

## Action

Occasionally dbcc checkalloc reports this error when no real error condition exists. You can either check to see if these errors are real, or continue with this section and take action to correct them, whether or not they reflect a real allocation error.

Because the process used to discover whether or not the errors are real can be time-consuming, you may want to go now to "Fix the Error" on page 2-261.

### Verifying That the Error Is Real

Run dbcc checkalloc in single-user mode if you suspect the 2541 error messages are incorrect. If the error is in *master*, use the section "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide* for instructions about how to invoke SQL Server in single-user mode. See the *SQL Server Reference Manual* for information about dbcc checkalloc.

## Fix the Error

1. Note the "extent#" in the message and then refer to "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to find the object ID and the index ID of the object that has the error.

2. If the error is on a user table (a system table's object ID is less than 100), go to step 3.

If the error is on a system table and the index ID is **not** 0, the error is on a system table index. Refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the error is on a system table and the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption or it may be necessary to restore from clean backups.

3. For user tables, if the index ID is 0 or 255, continue with step 4.

If the index ID is not 0 or 255, translate it into an index name:

```
1> use database_name
2> go
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

- 4. If the index ID is 0 or 255, do one of the following:
  - Restore the database from clean backups.
  - Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.
- WARNING!

Some data on this page might be lost if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

## Severity Level

16

## **Error Message Text**

```
Table Corrupt: Extent structures are linked
incorrectly; check the following extent: alloc
pg#=%ld extent#=%ld previous extent# on this
extent=%ld previous extent should be=%ld
```

#### Explanation

This error occurs when dbcc checkalloc detects a break in the chain of allocation structures, or extents, for a particular object. SQL Server only references an object's extents when allocating or deallocating space for that object, so this error does not affect queries against the data. However, any operations that require allocating or deallocating space may fail at run time.

#### ► Note

This error does not occur in SQL Server versions 4.8 and later.

## Action

Determine whether the error occurred on table data or on an index:

- 1. Note the first "extent #" in the message text. Use "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to find the object ID and the index ID of the object that has the error.
- 2. If the error is **not** on a system table (a system table's object ID is less than 100), go to step 3.

If the error is on a system table, restore the database from clean backups.

3. For user tables, if the index ID is 0 or 255, continue with step 4.

If the index ID is **not** 0 or 255, translate it into an index name:

```
1> use database_name
2> go
```

```
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

- 4. If the index ID is 0 or 255, do one of the following:
  - Restore the database from clean backups.
  - Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

## ♦ WARNING!

Some data on this page might be lost if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

**Additional Information** 

Refer to the *System Administration Guide* and the *SQL Server Reference Manual* for information about dropping and re-creating indexes.

Have the following information ready when you call Sybase Technical Support:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all the error messages
- dbcc page output for the page named in the error message
- dbcc tablealloc and dbcc checktable output for the table identified when dbcc page was run
- A record of the steps you have taken to recover from this error that may affect Technical Support's course of action

## **Severity Level**

16

## **Error Message Text**

```
Table Corrupt: Extent id %ld on allocation pg# %ld
had object id %ld (object name = %.*s) on but used
bit off
```

#### Explanation

This error is reported by dbcc checkalloc if an allocation structure or extent is being allocated but not used.

Error 2544 can result in various run-time failures and should be corrected, but if no other errors are occurring, you can wait to correct the problem until nonpeak hours.

#### Action

- 1. Note the "Extent id" in the message and then refer to "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to find the object ID and the index ID of the object that has the error.
- 2. If the error is on a user table (object ID is greater than or equal to 100), go to step 3.

If the error is on a system table (object ID is less than 100) and the index ID is **not** 0, the error is on a system table index. Refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the error is on a system table and the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption or it may be necessary to restore from clean backups.

3. For user tables, if the index ID is 0 or 255, continue with step 4.

If the index ID is **not** 0 or 255, translate it into an index name:

```
1> use database_name
2> go
```

```
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

- 4. If the index ID is 0 or 255, do one of the following:
  - Restore the database from clean backups.
  - Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

## ♦ WARNING!

Some data on this page might be lost if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

#### **Severity Level**

16

### Error Message Text

```
Table Corrupt: Extent id %ld on allocation pg# %ld has objid %ld and used bit on, but reference bit off.
```

#### Explanation

This error is reported by dbcc checkalloc if an allocation structure or extent is not linked to the other extents for the object referenced by "objid", but does reference a page which is linked in the object's page chain. Each occurrence of this error can represent a loss of up to eight pages on disk, or 16K. There is no data loss because these pages have not been used, but the pages cannot be used until the error is corrected. This error can result in various run-time failures and should be corrected, but if no other errors are occurring, you can wait until nonpeak hours to correct the problem.

#### ► Note

The instructions below are for fixing 2546 errors once they have occurred. Two easy-to-use strategies exist for **preventing** this error from occurring in the future. Refer to "Preventing Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about these strategies.

## Action for All SQL Server Versions

Occasionally dbcc checkalloc reports this error when no real error condition exists. You can either check to see if the error is real, or continue with this section and take action to correct it, whether or not it reflects a real allocation error.

Because the process used to discover whether or not the error is real can be time-consuming, you may want to go now to the "Action" section appropriate for your release level:

- "Action for 10.0 SQL Server and Later" on page 2-268
- "Action for 4.9.1 SQL Server and Later" on page 2-268
- "Action for 4.9.0 SQL Server and Earlier" on page 2-271

## Verifying That the Error Is Real

Run dbcc checkalloc in single-user mode if you suspect the 2546 error messages are incorrect. If the error is in *master*, use the section "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide* for instructions about how to invoke SQL Server in single-user mode. See the *SQL Server Reference Manual* for information about dbcc checkalloc.

## Action for 10.0 SQL Server and Later

If many of these errors are occurring, it is possible to clear them all at once by using the dbcc checkalloc and dbcc checkalloc with fix option commands. Refer to "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about using dbcc checkalloc.

## Action for 4.9.1 SQL Server and Later

Execute the following query to make sure the object exists and is correctly referred to in the system catalog:

```
1> use database_name
2> go
1> select object_name ( objid_from_error_msg )
2> go
```

If an object name is returned, then the error is on an existing, correctly referenced object. If this is the case, go now to the section "Identify Table: User or System Table" on page 2-268.

If a number, or something other than an object name is returned, use "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide*.

## Identify Table: User or System Table

Look at the value for "objid" in the error message. If it is 100 or greater, continue with the next section "Action for User Tables". If the "objid" is below 100, it is a system table and requires a different procedure as described in the section "Action for System Tables" on page 2-269.

## **Action for User Tables**

If the "objid" in the error message is 100 or greater, follow these steps to correct the error:

- 1. Run the dbcc page command described in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to obtain the value for the "indid." Substitute the value for "extent id" in the 2546 error message in place of the "page\_number" as described in the procedure.
- 2. Run dbcc tablealloc or dbcc indexalloc, depending on whether the value for "indid" indicates it is a table or an index. Before you run the appropriate command, keep the following in mind:
  - dbcc tablealloc will correct this problem on a table or an index, but if the problem is on an index, you can avoid affecting the entire table by using dbcc indexalloc. If the table is large or heavily used, it may be more practical to use dbcc indexalloc.
  - These commands will correct the error only when run in the full or optimized mode as long as the nofix option is not specified, the default for user tables.
  - You can use the object name, if you know it, or the "objid" value from the error message in the commands above where the argument *objid* appears.

Use the command appropriate for your situation:

For Tables (index id in extent = 0)	For Indexes (0 < index id in extent < 255)
1> dbcc tablealloc ( <i>objid</i> ) 2> go	<pre>1&gt; dbcc indexalloc (objid, indid) 2&gt; go</pre>

Refer to the *SQL* Server Reference Manual and System Administration *Guide* for additional details on dbcc tablealloc and dbcc indexalloc.

## **Action for System Tables**

If the "objid" in the error message is less than 100, follow these steps to correct the error:

1. Run the dbcc page command described in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to obtain the value for the "indid." Substitute the value for "extent id" in the 2546 error message in place of the "page\_number" as described in the procedure.

- 2. Put the affected database in single-user mode:
  - If the database is *master*, use the procedure in "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide*, and then go to step 3.
  - If the database is not *master*, use the sp\_dboption stored procedure to put the affected database in single-user mode:

```
1> use master
2> go
1> sp_dboption database_name, single, true
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

- 3. Run dbcc tablealloc or dbcc indexalloc, depending on whether the value of the "indid" from step 1 above indicates that it is a table or an index, then execute the appropriate command. Before you run the appropriate command, keep these facts in mind:
  - dbcc tablealloc will correct either a table or an index, but if the problem is on an index, you can avoid affecting the entire table by using dbcc indexalloc. If you need to minimize the amount of time the table is unavailable, it may be more practical to use dbcc indexalloc.
  - These commands will correct the error only when run in the full or optimized mode with the fix option specified, because the default for system tables is nofix.
  - You can use the object name if you know it, or "object id" value from the error message in the commands above where the argument *objid* appears.

For Tables (indid = 0)	For Indexes (0 < indid < 255)
<pre>1&gt; dbcc tablealloc (objid, 2&gt; full, fix)</pre>	1> dbcc indexalloc ( <i>objid</i> , 2> <i>indid</i> , full, fix)
3> go	3> go

Use the command appropriate for your situation:

- 4. Turn off single-user mode in the database:
  - If the database is *master*, use "Returning SQL Server to Multi-User Mode" in the *SQL Server Troubleshooting Guide*.
  - If the database is not *master*, use the following procedure:

```
1> use master
2> go
1> sp_dboption database_name, single, false
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

See the *SQL Server Reference Manual* and *System Administration Guide* for additional details on the dbcc tablealloc and dbcc indexalloc commands.

Action for 4.9.0 SQL Server and Earlier

Use either of the following methods to correct 2546 errors:

- Recover the database from a known, clean backup, using your usual SYBASE system administration procedures.
- Call Sybase Technical Support. They may be able to correct the errors by dialing into your system and accessing the problem database. The database must be in read-only mode while Technical Support is working on it. Before calling, have the complete text of all the error messages available to provide to Sybase Technical Support.

#### **Severity Level**

16

#### **Error Message Text**

Missing segment in sysusages segmap.

#### Explanation

This error is reported by dbcc checkcatalog if there is a row in *sysusages* (in the *master* database) that has a value of 0 in the *segmap* column. A value of 0 means that the corresponding space will not be used for further space allocation (data, log, or user-defined objects). The error is nonfatal but it does indicate that any free space on that section of the database will not be used.

Causes of this error might be:

- All of the segments have been removed from a logical device (via sp\_dropsegment).
- A direct update has been made to the *segmap* column of *sysusages* where the value has been changed to 0.

## Action

If you do not want any new space allocated on the affected segment, no action is necessary.

Otherwise, run sp\_addsegment or sp\_extendsegment on the Sybase logical device that has the problem. If the segment does not already exist in *syssegments*, use sp\_addsegment. If the segment exists, use sp\_extendsegment.

To add a new segment:

```
1> use database_name
2> go
1> sp_addsegment segname, database_name, devname
2> go
```

where:

- *segname* is the name of the new segment.
- database\_name is the name of the database where the segment is to be defined.

• *devname* is the name of the database device where *segname* will be located. (A database device may have more than one segment associated with it.)

To extend an existing segment on the device that has a segmap of 0:

```
1> use database_name
2> go
1> sp_extendsegment segname, database_name, devname
2> go
```

where:

• *segname* is the name of the existing segment. You can extend the segment called *default*. If you do, the command would be:

```
1> use database_name
2> go
1> sp_extendsegment "default", database_name,
2> devname
3> go
```

- *database\_name* is the name of the database where the segment is to be extended.
- *devname* is the name of the database device that has the segmap 0.

To check that the problem has been resolved, type:

```
1> select * from sysusages where
2> dbid = db_id("database_name") and
3> segmap = 0
4> go
```

where *database\_name* is the name of the database where the segment you added or extended resides. If no rows are returned, the problem has been resolved.

## **Additional Information**

For further information about sp\_addsegment and sp\_extendsegment, refer to the *SQL Server Reference Manual*. For further information about segments, refer to the *System Administration Guide*.

## **Severity Level**

16

## **Error Message Text**

```
Extent not within segment: Object %ld, indid %d includes extents on allocation page %ld which is not in segment %d.
```

► Note

Error 2558 is only raised in SQL Server release 4.2 and earlier releases.

#### Explanation

A segment is a label used to point to one or more database devices. Segments map the future allocations of different types of data to different devices.

Error 2558 occurs when dbcc checkalloc discovers that a portion of a database resides on a segment which was not defined to contain that type of data. For example, if dbcc checkalloc discovers user data on a segment designated to hold transaction log data, Error 2558 will occur.

When a database gets this error, it may still function, possibly generating 1105 errors. There may also be problems with recoverability.

### **About Creating Segments**

By default, when SQL Server creates a database, it creates the following three segments:

- The **system** segment allows system catalog tables to be placed on devices that contain this segment for a particular database.
- The **default** segment allows user tables or indexes to be placed on devices that contain this segment without specifying the on clause (for example, create table test (i int) on segment\_1) for a particular database.
- The **log** segment allows the transaction log to be placed on devices that contain this segment for a particular database.

#### ► Note

A device can contain more than one type of segment, although a segment can contain only one type of data (user data, transaction log data, and so on). Thus, different data can be on the same device, but not on the same segment.

For example, this command creates the default, system and log segments for the database *example*:

```
1> create database example
2> on device_1 = 10
3> log on device_2 = 2
4> go
```

The *system* and *default* segments are mapped to *device\_1* and the *log* segment is mapped to *device\_2*. For the *example* database, all future space allocations for system or user objects will occur on *device\_1* and all transaction log records will be placed on *device\_2*. Other databases may use other portions of *device\_1* or *device\_2* and these devices may have different segment mappings for that database.

A 2558 error would result in the *example* database if system or user objects were located on *device\_2* or transaction log records on *device\_1*.

## What Causes a 2558 Error?

Several situations can cause the 2558 error. Among the most common are:

- Loading into a database which has different data and log mapping than the dumped database.
- Loading into a database which has different user segment definitions than the dumped database.

These potential causes are discussed in detail below.

### **Different Data and Log Mapping**

If this error occurred after you re-created and loaded a database from a dump, there were probably incorrect or different entries in the *master..sysusages* table. A database loaded from a dump must be created in exactly the same way as the database that was dumped. If it is not, different entries in *master..sysusages* may cause 2558 errors.

This can cause a "data on log" situation, where data such as user or system objects is loaded into a log segment. This means that less total

log space is available to the database. The data is still accessible, but processing may be slowed, if not totally stopped, by the shortage of log space.

There can also be a "log on data" situation, where portions of the transaction log are loaded into a data segment. This is not as serious as the "data on log" scenario. This is because the portion of the transaction log on the data segment is eventually truncated using the dump transaction command, freeing up the space. All future transaction log allocations will be correctly placed on the log segment.

Suppose the *example* database were altered as follows:

```
1> alter database example on device_1 = 2
2> go
```

The logical order of the database pages is:

- The first 10MB of pages are data (*default* and *system* segments).
- The next 2MB are log.
- The last 2MB are data.

A 2558 error could result if the database were dumped, then dropped and re-created with different segment mappings. For example:

```
1> create database example on device_1 = 12
2> log on device_2 = 2
3> go
```

Although both ways of creating the database (create and alter, or just create) allocate the same amount of data and log space on the same devices, the mapping of data and log space is not the same. Refer to *Figure 2-1: Example of a database load that creates 2558 errors.* 

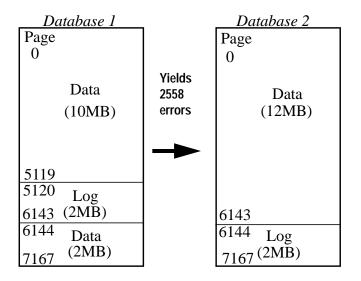


Figure 2-1: Example of a database load that creates 2558 errors

♦ WARNING!

This example shows how 2558 errors are caused. Do not load databases in this manner!

*Database 1* is dumped and loaded into *Database 2*, resulting in incorrect mapping of data and log space.

SQL Server dump and load commands operate at the logical page level. In this example, logical pages 5120 through 6143 on *Database 1* may contain transaction log records but when loaded on *Database 2*, they are mapped onto a data segment. Similarly, pages 6144 through 7167 are mapped for data on *Database 1*, but are mapped for log on *Database 2*.

If the 2MB data segment in this example was completely full on *Database 1*, there would be no log space available on *Database 2* after the load. Dumping the transaction log at this point would have no effect since the log segment is filled with data.

All space allocations that occur on *Database 2* after the load completes are correctly mapped.

## **Different User-Defined Segment Mapping**

The mapping of user-defined segments in *sysusages* for the database being loaded do not match those of the database that was dumped. This can result from:

- Creating user-defined segments in a different order on the database being loaded than on the database that was dumped.
- Dropping user-defined segments on the database that was dumped and not reflecting this change on the database being loaded into.

Once you do a load database with one of these error conditions, the 2558 error exists, but is not reported until you run a dbcc checkalloc on the loaded database.

## sp\_dropsegment

If sp\_dropsegment is used to remove a segment for which existing objects are still mapped, dbcc checkalloc will display Error 2558.

## Prevention

Keep an up-to-date copy of the *master..sysusages* system table as well as scripts to re-create all databases. Use the scripts to ensure that the entries in the *master..sysusages* table for the database being loaded match the corresponding entries for the database that was dumped. More specifically, the *segmap*, *lstart*, and *size* columns of *sysusages* must be identical in content and order. Verify this before beginning the load database command.

The entries in *sysusages* will be correct on the database being loaded into if you execute the following commands with the same parameters and in the same order as they were on the dumped database:

- create database
- alter database
- sp\_addsegment
- sp\_dropsegment
- sp\_extendsegment
- sp\_logdevice

#### ► Note

You cannot depend on the output of the system procedure **sp\_helpdb** to compare the contents of the *sysusages* table, because it does not necessarily display rows from *sysusages* in the actual logical page order. Instead, do direct selects against the *sysusages* table.

### Action

If there are a substantial number of 2558 errors due to an incorrect **load database**, it might be a good idea to drop and correctly re-create the database, and then reload it.

Use this section if you have 2558 errors and cannot re-create the database from scripts or hard copy.

2558 errors occur on these types of pages:

- Data pages or clustered index pages
- Nonclustered index pages
- text or image data pages
- System tables
- Transaction log pages

The action needed to correct this error depends on the type of page on which the error occurred.

Use the following table to match up the object ID value ("Object" in the error message) and index ID ("indid" from the error message) with the corresponding data page type, and then go to that section for the appropriate action. All action sections follow the table:

Type of Data	Corresponding Object ID and Index ID Values
Data Pages or Clustered Index Pages (page 2-280)	object ID > 99, index ID = 0 or 1
Nonclustered Index Pages (page 2-280)	object ID > 99, 1 < index ID < 255
text or image Data Pages (page 2-280)	object ID > 99, index ID = 255
System Table Pages (page 2-280)	object ID < 100 (and not equal to 8)
Transaction Log Pages (page 2-281)	object ID = 8

Table 2-7: Data types and object/index ID values

An object ID value of 100 will not occur.

**Data Pages or Clustered Index Pages** 

(Object ID > 99 and Index ID = 0 or 1)

The error occurred on the data page or on the clustered index of a user table. Resolve the problem with either one of the following procedures:

- Create a clustered index on the table, if you have the available space needed. If a clustered index already exists, drop and re-create it. If not, creating one will clear the error. You can then drop the clustered index.
- Copy the data out of the table, drop and re-create the table, and copy the data back in using bcp or select/into.

## **Nonclustered Index Pages**

(Object ID > 99 and 1 < Index ID < 255)

The error occurred on the nonclustered index of a user table. The error can be cleared by dropping and re-creating the index. Creating a clustered index causes all nonclustered indexes to be rebuilt as well. Therefore, if 2558 errors are occurring on several indexes on the same table, you can create a clustered index on that table to clear all these errors.

## text or image Data Pages

(Object ID > 99 and Index ID = 255)

The error occurred on data which is *text* or *image* datatype. Use either procedure below to correct the situation:

- 1. Select all the information from the bad table into a new table.
- 2. Drop the bad table. Any subsequent dbcc checkalloc will continue to show this error until the bad table is dropped.
- 3. Use sp\_rename to rename the new table with the old table name.

Or:

Copy the table out, using **bcp** or **select/into**. Then drop and re-create the table, and copy the contents of the table back in, using **bcp** or **select/into**.

System Table Pages

(Object ID < 100 and Not 8)

Call Sybase Technical Support.

**Transaction Log Pages** 

(Object ID = 8)

If the error occurs on pages containing transaction logs, do the following:

- 1. Have all users finish transactions and wait until this procedure is finished to initiate any further transactions.
- 2. Perform the checkpoint command:

```
1> use database_name
2> go
1> checkpoint
2> go
```

- 3. Dump the transaction log according to your normal procedures.
- 4. Notify users that they may resume normal operations.

If you received the error because of an sp\_dropsegment, you may be able to resolve it using sp\_addsegment or sp\_extendsegment.

## **Additional Information**

See the *System Administration Guide* for more information about segments and devices.

See the *SQL Server Reference Manual* for more information about checkpoint and dump transaction.

## **Severity Level**

16

## **Error Message Text**

```
Data page number %ld is empty but is not the first page. Status = 0x%x.
```

### Explanation

When an object is first created, a single, empty page is allocated to it. Usually there are no other empty pages. Error 2559 occurs when SQL Server encounters an empty page while traversing the page chain of an object which is not the first page of the chain.

If you are not encountering any other errors either at run time or from dbcc, it is possible that the empty page will not cause any further problems. Therefore, it is acceptable to wait to correct this problem until nonpeak hours. If other errors are occurring, refer to the information in this manual if applicable or contact Sybase Technical Support.

## Action

Correct this situation by forcing the page chain involved to be rebuilt. This can be accomplished in different ways, depending on whether or not the page chain involved is associated with an index or table data. To determine this, refer to "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to determine the index ID (*indid*) and object ID associated with the page number from the error message.

### Tables or Clustered Indexes (Index ID is 0 or 1)

There are two ways to correct this error, depending on whether you have enough extra disk space to create a clustered index on the table in question.

### Solution 1: Drop Index

If a clustered index exists, drop and re-create it. If not, create a dummy clustered index and then drop it. This will remove this problem since creation of a clustered index causes the table to be copied and does not copy the empty pages. Note that clustered index creation does require considerable disk space (see the *System Administration Guide*).

Solution 2: Use the Bulk Copy Utility

Use bcp to copy the table out of and then back to the Server:

- 1. **bcp** the table data out to an operating system file.
- 2. Empty the table with the truncate table command.
- 3. (Optional) Drop all indexes on the table.
- 4. bcp the table data into SQL Server.
- 5. (Optional) Re-create all indexes on the table.

For Nonclustered Indexes (1 < Index ID < 255)

There are two ways to correct this error on a nonclustered index:

**Solution 1: Drop Index** 

1. Execute the following query in the database in question to determine the index name:

```
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

where *object\_ID* and *index\_ID* are from the output of dbcc page run earlier.

2. Drop and re-create the nonclustered index specified in the query.

Solution 2: Create a Clustered Index

Create a clustered index since this forces all nonclustered indexes to be rebuilt. This method may not be suitable for large tables because of the overhead required for creating a clustered index.

## **Severity Level**

14

## **Error Message Text**

```
Permission denied. Only a user with System
Administrator (SA) authorization can use the DBCC
command '%.*s'.
```

## **Explanation**

As of SQL Server release 10.0, only the "sa" account or users with sa\_role can issue the following dbcc commands:

- dbcc dbrepair
- dbcc traceon
- dbcc traceoff
- dbcc checkalloc
- dbcc memusage
- dbcc page

Many stored procedures that use dbcc commands require sa\_role. Some of these stored procedures are:

- sp\_addsegment
- sp\_addthreshold
- sp\_dbremap
- sp\_dropsegment
- sp\_dropthreshold
- sp\_extendsegment
- sp\_logdevice
- sp\_modifythreshold
- sp\_placeobject

## Action

Run the dbcc commands and stored procedures listed above from the "sa" account or make sure you have sa\_role granted to you. To grant

sa\_role to a user, you must have sa\_role granted to you. To grant sa\_role to a user, type:

```
1> sp_role "grant", sa_role, login_name
2> go
```

where *login\_name* is the name of the login account to which sa\_role is being granted. When you grant a role to a user, it takes effect the next time the user logs into SQL Server. However, the user can immediately enable the role by using the set role command. For example, this command:

1> set role "sa\_role" on
2> go

enables the sa\_role role for the user.

## **Additional Information**

Refer to the *System Administration Guide* for information about using the "sa" account, associated roles, and how locking the "sa" account can affect scripts. Refer to the *SQL Server Reference Manual* for information about granting roles.

#### **Severity Level**

10

#### Error Message Text

Database '%S\_DBID' is not in single user mode - may find spurious allocation problems due to transactions in progress.

#### Explanation

This warning occurs when you execute dbcc checkalloc on a database that is not in single-user mode. dbcc checkalloc displays this message and then continues to execute.

Executing dbcc checkalloc on a database that is not in single-user mode may cause error messages to be displayed which do not reflect the actual state of your database. These error messages are referred to as spurious. Error messages 2521, 2540, 2541, 2546, 7939, 7940, and 7949 may be spurious under these conditions.

### Action

No action is required.

If other errors are reported by **dbcc checkalloc**, refer to this manual for information on how to deal with them.

If your operational environment allows it, put the database in question into single-user mode prior to running dbcc checkalloc.

- For the *master* database, see "How to Start SQL Server in Single-User Mode" in the SQL Server Troubleshooting Guide.
- For other databases, use sp\_dboption as follows:

```
1> use master
2> go
1> sp_dboption database_name, single, true
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

## **Additional Information**

Refer to "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide.* 

See the *SQL Server Reference Manual* for information about dbcc checkalloc.

## **Severity Level**

16

## **Error Message Text**

```
Database '%.*s' is not marked suspect.
You cannot drop it with DBCC.
```

## Explanation

This error occurs when you try to drop a database using dbcc dbrepair. Only databases which have been marked suspect can be dropped using this command.

Databases are marked suspect in the following ways:

- SQL Server marks a database suspect when some critical errors occur.
- The Sybase System Administrator can mark a database suspect in order to drop it with the dbcc dbrepair command.

## Action

Try to drop the database with the drop database command. If this fails, use the procedure described in "How to Drop a Database When drop database Fails" in the *SQL Server Troubleshooting Guide*.

#### **Severity Level**

16

#### Error Message Text

```
%S_PAGE has an incorrect pgfreeoff value of %d.
The offset should be %d.
```

#### Explanation

The free byte offset on a page indicates the offset of the first unused byte on that page. Error 2596 occurs when dbcc checktable detects that the free byte offset on a page is incorrect.

This is a serious error as it indicates possible corruption on the affected page.

This error can occur in conjunction with other errors, particularly Error 2506.

## Action

If Error 2596 occurs on a system table being processed by dbcc checktable, call Sybase Technical Support.

#### Case 1

If the number of the free byte offset shown in the error message is **lower** than the expected number, some data has been lost. To determine how much data has been lost, find the difference (in bytes) between the actual free byte offset and the expected free byte offset (both numbers are shown in the error message output). The amount of data lost includes 2 bytes overhead per row where there are only fixed-length columns or 4+n bytes overhead per row with variable-length columns. If you need help with this, call Sybase Technical Support.

Decide whether you should restore the database that contains the affected table from backups. In making that decision, consider the number of rows lost and the number of other corruption errors that have occurred on this table and database.

## Case 2

If the free byte offset on the page header is **higher** than the expected value, no data has been lost. However, some extraneous rows may exist.

If you have clean backups of your database, recover from backups. If you do not have clean backups of your database, follow these steps to correct the error:

- 1. Use the procedure in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond to the page number from the error message text.
- 2. If the object with the error is **not** a system table (a system table's object ID is less than 100), continue with step 3.

If the object with the error is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the object with the error is a system table and the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption or it may be necessary to restore from clean backups.

3. For user tables, if the index ID is 0 or 255, continue with step 4.

If the index ID is **not** 0 or 255, translate it into an index name:

```
1> use database_name
2> go
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

4. If the index ID is 255, delete or update the data row to remove the reference to the bad text page.

If the index ID is 0, do one of the following:

- Restore the database from clean backups.

- Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

## ♦ WARNING!

Some data on this page might be lost if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

Before dumping your database, make sure it is working correctly by running the following commands prior to each dump:

- 1. dbcc checkdb.
- 2. dbcc checkalloc or dbcc checkalloc with the fix option. (See "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about running these commands in multi-user mode and how to prevent spurious allocation errors from dbcc commands.)

## insert Errors

This section contains error messages pertaining to insert commands.

## **Severity Level**

14

## **Error Message Text**

```
Attempt to insert duplicate key row in object '%.*s' with unique index '%.*s'%S_EED
```

## Explanation

No two rows can have the same index value (including NULL) in a column or columns with a unique index. SQL Server checks for duplicate values when the index is created (if data already exists) and checks each time data is added with an insert or update. Error 2601 occurs when you try to put duplicate index values into a column or columns with a unique index.

### Action

Using a unique index makes sense only when uniqueness is a characteristic of the data itself. Choose one of the following solutions, depending on whether or not you need a unique index:

- If you need duplicate index values in the indexed column or columns, drop the unique index and create a nonunique index instead.
- If you need a unique index on data which contains duplicate values, you must change some values to remove the duplicates. To do this:
  - 1. Use a select statement to find the row which will be duplicated by your update or insert command.
  - 2. Modify either the data in the table or the data which you want to insert, so that the index values do not match.

## **Additional Information**

See create index in the *SQL Server Reference Manual* for more information about indexes and their options.

## **Severity Level**

22

## **Error Message Text**

Could not find leaf row in nonclustered index '%.\*s' that corresponds to data row from logical data page %ld, row offset %d during update index attempt after data page split.

## Explanation

This error occurs when SQL Server fails to update a nonclustered index at the leaf level after a page split (a page split occurs when a page fills up and there is no more room to write on that page). This error message indicates index corruption.

This is a very serious error. You may be able to recover from this error but you may have to restore from known, clean backups.

### Action

- 1. Record the value of the data page and index name (or ID if that is given instead) from the error message. Using this information and the instructions in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide*, determine which table corresponds to the data page number from the error message text.
- 2. If the table determined in step 1 is a system table (object ID is less than 100) and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index. Then go to step 5.

If the table determined is step 1 is a system table and the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption, but it may be necessary to restore from clean backups.

3. If the table determined in step 1 is **not** a system table and the message displayed the index ID rather than the index name, find

the name of the index involved by executing the following query:

```
1> select name from sysindexes
2> where indid = index_ID
3> and id = object_ID
4> go
```

where *object\_ID* is the ID of the table as determined in step 1 and *index\_ID* is the ID of the index specified in the error message.

- 4. Drop and re-create the index using the table name obtained in step 1 and the index name determined in step 3.
- 5. To verify that all problems have been resolved on this table, run dbcc checktable and dbcc tablealloc on the affected table. If you have a pre-4.9.1 SQL Server, dbcc tablealloc is not available; run dbcc checkalloc instead.

## **Additional Information**

If problems persist, this procedure may not be sufficient to clean up the index corruption, and you should contact Sybase Technical Support. Have the output from the dbcc checkalloc or dbcc tablealloc and dbcc checktable commands available for review.

## Severity Level

14

## **Error Message Text**

Attempt to insert duplicate row in table '%.\*s' with index '%.\*s' in database '%.\*s'. Could drop and recreate index with ignore duprow or allow duprow.

### Explanation

Tables with clustered indexes do not allow duplicate rows. Error 2615 occurs when you try to insert a duplicate row in a table which already has a clustered index.

#### Action

If the duplicate values are a result of incorrect data, change the data and rerun the transaction.

Otherwise, decide whether you want to allow or prevent duplicate rows in the table where the clustered index will be created. Use one of the following procedures.

### ► Note

ignore\_dup\_row and allow\_dup\_row are mutually exclusive options of the create index command. Do not use them together in the same statement, or you will cause Error 1916.

### Allow Duplicate Rows

Create the index with the allow\_dup\_row option:

```
1> create clustered index index_name on table_name
2> with allow_dup_row
3> go
```

Subsequent update and insert commands can create duplicate rows after the allow\_dup\_row option is set.

Do not use this procedure if you have a unique index on any column or columns in the table. Unique indexes are only used on tables where no duplicate rows exist.

## **Delete Duplicate Rows**

Create the index with the ignore\_dup\_row option:

```
1> create clustered index index_name on table_name
2> with ignore_dup_row
3> go
```

Existing duplicate rows will be deleted from the table as the index is created. Subsequent attempts to enter a duplicate row with insert or update are ignored and the insert or update is cancelled with an informational message. If the insert or update is part of a larger transaction, the rest of the transaction will continue as usual.

## **Additional Information**

Refer to create index in the *SQL Server Reference Manual* for information about ignore\_dup\_row and allow\_dup\_row.

## Severity Level

21

## **Error Message Text**

```
The offset of the row number at offset %d does not match the entry in the offset table of the following page: %S_PAGE.
```

### Explanation

The offset table is stored at the end of every data page and it indicates where rows are located on that page. SQL Server uses the offset table to quickly access the rows in each page. Error 2620 occurs during an insert operation when SQL Server detects inconsistencies in the offset table of the page specified by the error message.

This is a serious error. You may be able to recover from this error but you may have to restore from known, clean backups.

#### Action

- 1. Record the value of the page from the error message. Using this information and the instructions in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide*, determine which table and index corresponds to the page number.
- 2. If the object encountering the error is **not** a system table (a system table's object ID is less than 100), continue with step 3.

If the object is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption but it may be necessary to restore from clean backups.

3. For user tables, if the index ID is 0 or 255, continue with step 4.

If the index ID is **not** 0 or 255, translate it into an index name:

```
1> use database_name
2> go
```

```
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify that the corruption is gone.

- 4. If the index ID is 0 or 255, do one of the following:
  - Restore the database from clean backups.
  - Refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

#### ♦ WARNING!

Some data on this page might be lost if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

Before dumping your database, make sure it is working correctly. More specifically, the following commands should be run prior to each dump:

- 1. dbcc checkdb.
- 2. dbcc checkalloc or dbcc checkalloc with the fix option. (See "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about running these commands in multi-user mode and how to prevent spurious allocation errors from dbcc commands.)

### **Additional Information**

If problems persist, this procedure may not be sufficient to clean up the index corruption, and you should contact Sybase Technical Support. Have the output from the dbcc checkalloc or dbcc tablealloc and dbcc checkdb commands available for review.

## **Create Utilities Errors (Continued)**

This section contains error messages pertaining to create operations (table, database, and so on).

#### **Severity Level**

16

## **Error Message Text**

```
There is already an object named '%.*s' in the database.
```

#### Explanation

This error occurs when SQL Server tries to create an object which already exists.

Error 2714 can occur in the following situations:

- When you are creating procedures, triggers, tables, views, or rules, if you specify an object that already exists.
- If you have changed your sort order to be case-insensitive and then you try to create an object that is spelled the same way as an existing object with a different case (for example, PROC1 and proc1).
- When using temporary tables, the first 13 bytes of the table name, including the pound sign, must be unique for the current session. Long temporary table names are truncated to 13 characters with the pound sign. An attempt to create a second temporary table with the same name in the same session will raise Error 2714. (In contrast, temporary tables created by specifying *database\_name.owner.table\_name* may be shared among sessions, not only within one session.)

## Action

Make sure the name of the object you are trying to create does not already exist.

If you have stored procedures that create objects, make sure there is a check for the existence of the object in the stored procedure (before it is created). If you wish to re-create an object within a stored procedure, drop the object before you re-create it.

#### **Additional Information**

Refer to the *SQL Server Reference Manual* for information about creating and dropping objects.

## Procedure Manager Errors

This section contains error messages pertaining to the Procedure Manager.

### **Severity Level**

18

### **Error Message Text**

```
Cannot create procedure dbid %d, objid %ld, with a group number of %d.
```

#### Explanation

This error occurs when SQL Server is unable to insert a row into the *sysprocedures* table. This can happen during creation of a procedure, view, or trigger, or during remapping of a compiled object.

The most common reason for Error 2811 is lack of data space to add the procedure, and therefore it is often raised following an 1105 error.

#### Action

Examine the SQL Server error log to determine what errors occurred before the 2811 error. If an 1105 error occurred, refer to "Error 1105" on page 2-120 for recovery instructions. If errors other than 1105 occurred, follow the recovery instructions in this manual for those errors. If your manual does not include these instructions, call Sybase Technical Support.

## **Severity Level**

16

## **Error Message Text**

Stored procedure '%.\*s' not found. Specify owner.objectname or use sp\_help to check whether the object exists (sp\_help may produce lots of output).

### Explanation

This error occurs when an attempt is made to execute a stored procedure that does not exist. If the procedure actually does exist, that is, it appears when sp\_help is run with no parameters, Error 2812 can occur if you do not fully qualify the procedure name.

SQL Server's naming convention for database objects is as follows:

database.owner.object\_name.column\_name

The default value for *database* is the current database and the default value for *owner* is the current user. Remember that the owner is part of the object, so two different users can have two different procedures with the same *object\_name* in the same database, for example *user1.proc1* and *user2.proc1*.

If the procedure is not owned by the user attempting to execute it, and it is not owned by the database owner ("dbo"), then all references to it must include the owner name. For example, suppose *user1* creates a procedure called *proc1*. Any users other than *user1* must prefix the procedure name with the owner name when executing it, that is, exec *user1.proc1*.

The system stored procedures are an exception to the naming convention. These procedures reside in the *sybsystemprocs* database, are owned by the Sybase system administrator, and have names starting with "sp\_." You can execute system stored procedures from any database and they will reference the system catalog of the current database. The search path for all "sp\_" procedures is:

- 1. The current database
- 2. The sybsystemprocs database
- 3. The master database

If Error 2812 occurs while running the installmaster script, it means that the sp\_configure procedure was not found. You can ignore this.

If Error 2812 occurs while you are trying to run the Backup Server, you may be trying to dump to a SQL Server rather than to a Backup Server.

#### Action

If you do not know who owns the procedure, use sp\_help to display the owner. Run this procedure without any parameters to display objects owned by other users. Alternatively, you can use the following query to determine who owns the stored procedure you are attempting to execute:

```
1> select name,owner=user_name(uid)
2> from sysobjects
3> where name = "proc1"
4> go
```

If the procedure does not appear in the output of this query, it is either in a different database or does not exist.

If you do not own the procedure, qualify the procedure name with the owner name. For example:

```
1> execute user1.proc1
2> go
```

For procedures used by many users of a database, it is usually easiest if the database owner creates the procedure. This will allow any user to find the procedure without specifying an owner name.

If the procedure is not in the database where it is executed, fully qualify the procedure name with the database name. For example:

```
1> exec database_1.user1.proc1
2> go
```

The owner name is not needed if you or the database owner own the procedure. For example:

```
1> exec database_1..proc1
2> go
```

Execute permission must be provided so other users can execute this procedure, but no permissions are required to see the text of the procedure.

## **System Stored Procedures**

If this error occurs on system stored procedures, it might be resolved by running the installmaster script. This installs all of the system procedures and initializes various other SQL Server structures.

## **Backup Server**

If Error 2812 occurs while you are trying to run the Backup Server, you may be trying to dump to a SQL Server rather than to a Backup Server. Check your Backup Server assignment with the following command:

```
1> sp_helpserver SYB_BACKUP
2> go
```

The *network\_name* column is the interfaces file that contains the entry for this Server.

Check the status of this Server:

```
1> exec SYB_BACKUP...sp_ps
2> go
```

Open Server applications have the stored procedure sp\_ps while SQL Servers do not. Therefore, if this command fails because sp\_ps is not found, most likely the Server defined for SYB\_BACKUP is a SQL Server and not a Backup Server.

## **Additional Information**

Refer to the *Transact-SQL User's Guide* for additional information about naming conventions.

### **Severity Level**

19

### **Error Message Text**

```
Process %d cannot hold PROC_BUF 0x%1x named '%.*s'
because it holds another PROC_BUF 01%1x named
'%.*s'. A process can only hold one view, rule, or
default at a time.
```

#### **Explanation**

The Buffer Manager controls and synchronizes access to process buffers. A process can hold many procedures and triggers but it can only hold one view, rule, or default at a time. If a process already holds a view, rule, or default, and SQL Server tries to add another one for the process, Error 2824 is raised.

Error 2824 can be caused by:

- A view referenced by stored procedures or triggers has been dropped and re-created. (Stored procedures and triggers are not reresolved when a view they reference is dropped and re-created.)
- Procedure cache corruption has occurred.

#### Action

Use sp\_help view\_name to determine whether a view that stored procedures or triggers reference has been dropped and re-created. If the creation time of the view is later than the creation time of the procedure or trigger that refers to it, drop and re-create the stored procedures and triggers that reference that view.

If the 2824 error persists, there is probably procedure cache corruption. Stop and restart SQL Server to clear the corruption.

## dump and load Errors

This section contains error messages pertaining to the  $\mathsf{dump}$  and  $\mathsf{load}$  utilities.

## **Severity Level**

23

## **Error Message Text**

## For SQL Server Releases Up to 4.9.x

```
Attempt to dump database %.*s found logical page %ld when logical page %ld expected.
```

## **Explanation**

This error occurs when SQL Server finds an unexpected page number during a database dump. This error is fatal and stops the dump from completing.

Error 3002 is most likely caused by corruption of a database as a result of either:

- Hardware failure
- Inadvertently trying to use a raw partition for two separate purposes

## Action

Run dbcc checkalloc to determine whether your database has a bad page number or page header. If this is the case, dbcc checkalloc will display Error 2529. In this case, refer to "Error 2529" on page 2-254 to determine how to recover from this error.

## **Hardware Errors**

- Check the SQL Server error log to see if there are other indications of hardware problems, such as kernel messages reporting I/O errors.
- Check the operating system error log or diagnostics utilities for I/O errors.

## **Raw Partition Misuse**

Refer to "Correct Use of Raw Partitions" in the *SQL Server Troubleshooting Guide* to determine whether a raw partition is being used incorrectly.

## **Incompatible Drives**

Some drives which have their read-ahead cache enabled may cause Error 3002. Turning their read-ahead cache off clears the error. If you suspect that this is the cause of your corruption, contact your hardware vendor.

In this case, rebooting SQL Server and restarting the dump might clear the error.

## **Additional Information**

If none of the above actions have solved the 3002 error, call Sybase Technical Support. Before you call, have the following information on hand:

- SQL Server release and EBF Rollup Level
- SQL Server and Backup Server error logs
- Operating system error log
- Text of all error messages
- Results from dbcc checkdb and dbcc checkalloc if possible

#### Severity Level

10

### Error Message Text

#### Release 10.0 and Later

You must be the DBO of database %.\*s or have Operator (OPER) role to run DUMP DATABASE.

#### Release 4.9.x and Earlier

%ld uninitialized pages encountered while dumping database %.\*s. Run DBCC CHECKALLOC on this database for more information, then call technical support if there are any errors.

#### Explanation

#### Release 10.0 and Later

This error occurs when you try to run dump database on a database you do not own and for which you do not have Operator (oper) role.

#### Release 4.9.x and Earlier

This error occurs when SQL Server, during a database dump, finds a database page that is allocated but not initialized. Since the page is not used, it is reported as an uninitialized page.

This error does not affect the validity of the database dump, as long as dbcc checkalloc runs without reporting additional errors. Run dbcc checkalloc as described below to determine whether the dump is valid.

### Action

#### Release 10.0 and Later

Ask your database administrator to grant the operator role (oper\_role) to you.

## Release 4.9.x and Earlier

Once the database dump has completed, run **dbcc checkalloc** on the database in question. If no errors are reported, ignore this error.

Even if dbcc checkalloc does not return errors, the uninitialized pages may still exist. A subsequent database dump could display the same message. This condition is eventually resolved through normal SQL Server operations. If this error persists and the same number of pages are reported every time, Sybase Technical Support may be able to correct the cause of the error. (You must have dial-in facilities to get help from Technical Support on this problem.)

## Prevention

Run dbcc checkalloc and dbcc checkdb before each database dump.

#### Severity Level

16

### Error Message Text

## Release 10.0 and Later

Database in use. A user with System Administrator (SA) role must have exclusive use of database to run load.

#### Release 4.9.x and Earlier

Database in use. System Administrator must have exclusive use of database to run load.

#### **Explanation**

You cannot use **load database** while a database is in use by any user (even yourself). Error 3101 occurs when you try to load a database dump while other users are still accessing the database.

► Note

This error does not occur on master.

#### Action

Wait until all the users have finished using the database, then run the load database command again.

Make sure that you are not in the database being loaded while you issue the load database command. Sybase recommends running all load statements from within the *master* database, although any database other than the one being loaded can be used.

## **Severity Level**

16

## **Error Message Text**

```
LOAD DATABASE encountered page with invalid logical page number %ld.
```

## Explanation

This error occurs when SQL Server detects an invalid page number within a database dump. The invalid page number is greater than the highest page number for the database to be loaded into. This is a serious error and it indicates a corrupted database dump.

Attempts to access this database after it encounters Error 3104 may result in Error 930.

Some potential causes of the error are:

- Overlapping partitions.
- Using UNIX operating system files for Sybase database devices (that is, if the file system fills up).

## ► Note

This error only occurs in SQL Server releases earlier than 10.0.

## Action

- 1. Examine your operating system error log file and your SQL Server error log to determine if hardware errors may have caused the corruption.
- 2. If you have a clean backup of your database (not the one you just loaded), drop your database, re-create it with the **for load** option and load the dump into it.

If you do not have a known clean backup of your database, contact Sybase Technical Support for assistance.

## Prevention

In some cases, **dbcc** commands detect a problem in the database before the database is dumped, thus preventing the database from

being dumped and loaded. More specifically, the following commands should be run prior to each dump:

- 1. dbcc checkdb.
- 2. dbcc checkalloc or dbcc checkalloc with the fix option. (See "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about running these commands in multi-user mode and how to prevent spurious allocation errors from dbcc commands.)

Before loading your database dump into a database, make sure that:

- 1. The database you will load into has the same data and log device allocations as the database dump. For more information about data and log mapping and user segment definitions, see "Error 2558" on page 2-274.
- 2. The database dump has the same user segment definitions as the database you want to load into.

## Additional Information

Refer to:

- "Device Administration Issues" in the SQL Server Troubleshooting Guide
- load database in the System Administration Guide

Before contacting Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server and Backup Server error logs
- Operating System error log
- Text of all the error messages

## **Severity Level**

16

## **Error Message Text**

```
Data on dump will not fit into current database.
Need %d Mbyte database.
```

## Explanation

This error occurs when an attempt is made to load a database dump into a database smaller than the dumped database. You cannot use dump database or load database to shrink a database, regardless of the amount of actual data in the database.

## Action

If your goal is to load the database from the dump, you must increase the size of the database being loaded into to at least the configured size of the dumped database. You can do this with either the alter database command or by dropping and re-creating a larger database.

To avoid 2558 errors on pre-10.0 SQL Servers, the database being loaded into must have identical data and log mapping, and identical segment mappings, as the dumped database. See "Error 2558" on page 2-274 and the *System Administration Guide* for more information.

If your goal is to shrink the total size of a database, you cannot use dump and load. You must copy all the data out of the database using bcp, drop the database and re-create a smaller database, then re-create the database structure with scripts, and reload the data with bcp.

#### Severity Level

16

#### Error Message Text

#### Release 10.0 and Later

The database you are attempting to LOAD was DUMPed under a different sort order ID (%d) or character set ID (%d) than the ones running on this server (SortOrd = %d, CharSet = %d). If the sort orders differ, at least one of them is non-binary.

#### Release 4.9.x and Earlier

The database you are attempting to LOAD was DUMPed under a different sort order ID (%d) than the one currently running on this server (%d), and at least one of them is a non-binary sort order.

## Explanation

A database cannot be loaded into a Server using load database unless the sort order of the Server that dumped the database matches the sort order of the Server being loaded into. Error 3120 occurs when you try to load a database on a SQL Server with a different sort order from that of the dumped database.

## Action

You have the following options:

- Move the data via bcp rather than load.
- Change the sort order of the SQL Server.

The procedure to change SQL Server sort order depends on the version:

- For SQL Server version 4.9 and later, see the *System Administration Guide* for instructions.
- For SQL Server version 4.2 through 4.8, you must rebuild the entire contents of the SQL Server. See the *International Supplement to the System Administration Guide* and the *System Administration Guide* for instructions.

- If the SQL Server into which you are attempting to load is a new Server, you can:
  - Reinstall SQL Server with the correct sort order.
  - Re-create the databases with the load option.
  - Retry the load.

#### Severity Level

16

### **Error Message Text**

#### Release 10.0 and Later

No dump device has been specified.

### SQL Server Releases Up to 4.9.x

```
Can't open dump device '%.*s', device error or
device off line. Please consult the SQL Server
error log for more details.
```

#### Explanation

This error occurs when SQL Server is unable to access a dump device during a database dump. This error is fatal and stops the dump from completing.

Error 3201 is most likely caused by one of the following:

- The device you specified for the dump is offline or otherwise unavailable to SQL Server.
- Permissions for the dump device are not set correctly for user "sybase" or the user performing the dump (read and write privileges are required).
- A previous dump aborted and SQL Server believes the dump device is still in use.
- For SQL Server releases previous to 10.0, you are trying to run more than one dump database command at the same time. You can only do this (for tape devices) if the *cntrltype* values (defined using sp\_addumpdevice) of the devices to which you are dumping are different. For example, only one disk byte stream interface can be active at a time. You cannot dump multiple databases or transaction logs to the same tape device or disk dump.

As of SQL Server 10.0, dumps are performed by the Backup Server. During dump and load commands, the Backup Server automatically determines whether a tape or disk device is being used and what its controller type is. It ignores the *cntrltype* parameters specified with sp\_addumpdevice.

## Action

1. Check to make sure the device you specified is defined for your SQL Server:

If the device is not there, you can use sp\_addumpdevice to add it.

- 2. At the operating system level, check the permissions for the dump device for user "sybase" or the user performing the dump (read and write privileges are required).
- 3. For SQL Server releases 4.9.x and earlier with the dump database being done to a tape dump device, do this step. Using sp\_who and dump device information from the SQL Server error log, check whether SQL Server believes a dump device is still in use from a previous dump that aborted. If this is true, restart SQL Server. This should clear the error and allow you to do your dump.
- 4. For SQL Server releases previous to 10.0, if you wish to run more than one dump database (to tape) command at the same time and the *cntrltype* values (as shown above) of the devices to which you are dumping are the same, change the appropriate *cntrltype* values. For disk and file dump devices and for tape drives to be used as disk dump devices, you can only define one *cntrltype* value and the controller number parameter must be 2 (byte stream interface). For tape dump devices, the controller number parameter must be between 3 and 8 (tape volume). To change a *cntrltype* value, drop and re-add the dump device.
- 5. If you have gotten through all the steps above and still have not solved the problem, check your operating system error log for operating system errors. If no errors are logged in your operating system error log and there are no other errors in your SQL Server or Backup Server error log, contact Sybase Technical Support for assistance.

## **Additional Information**

Refer to the *SQL Server Reference Manual* and the *System Administration Guide* for information about dump devices.

If you call Technical Support, have the following information on hand:

- SQL Server version and EBF Rollup level
- Backup Server version
- SQL Server and Backup Server error logs
- Text of all error messages
- select \* from sysdevices output
- Operating system error log

### **Severity Level**

16

### **Error Message Text**

### Release 10.0 and Later

Received MULTARG is not for device name as expected.

### **Releases 4.9.x and Earlier**

Write on dump device '%.\*s' failed, vsn=%ld return=%d status=%ld. Please consult the SQL Server error log for more details.

### Explanation

#### Release 10.0 and Later

The meaning of Error 3202 for SQL Server 10.0 and later is different from the meaning when it is raised on SQL Server 4.9.x and earlier. A MULTARG is a structure SQL Server stores in memory to keep information about the device being dumped to. If the dump routine was passed a MULTARG which is not for a dump device and you are running the Diagnostics Server (not likely), Error 3202 will be raised. If you receive Error 3202 when using SQL Server 10.0 or later, call Sybase Technical Support.

#### **Releases 4.9.x and Earlier**

For SQL Server 4.9.x and earlier versions, Error 3202 means that an error occurred when "writing" out a packet of data to the backup.

Common causes are:

- Running out of space for your dump (to disk)
- A bad tape
- A bad tape size
- A bad block size
- Hardware failure

This error is fatal and stops the dump from completing. This invalidates the backup and makes it unusable for recovery purposes.

The error message output includes:

- *vsn* the virtual socket number.
- *return* the return value: 0 means successful; -2 means failure.
- status the ending status, displayed in decimal value. The most common value is 524288 which usually means an I/O error. This status is Sybase-specific and has no operating system correlation.

### Action

► Note

The following actions apply only to SQL Server 4.9.x and earlier versions. If a 3202 error occurs on SQL Server 10.0 or later, call Sybase Technical Support.

To determine the cause of the 3202 error, look at the message above the 3202 error in your SQL Server error log. You should see a generic message about a write to tape/disk failure during a dump database or dump transaction. Depending on the reason Error 3202 was raised, perform one of the following actions.

## Not Enough Space

Error 3202 can occur in dumps to disk if the database you are dumping is larger than the size of the file you are dumping to. This limit can be user-imposed or system imposed. Under VMS, a user can assign a limit for size on individual files. Under many versions of UNIX, there is a 2GB file system size limit. If you have a database that exceeds these limits, dump the database to tape.

Error 3202 can occur when dumping to tape if the tape runs out before the entire database has been dumped. You will encounter this if the size specified for the device via sp\_addumpdevice is larger than the capacity of the device.

## Hardware Errors

• Check the SQL Server error log to see if there are other indications of hardware problems, such as kernel messages reporting I/O errors.

• Check the operating system error log or diagnostics utilities for I/O errors.

Check your operating system error log for operating system errors.

If no errors are logged in your operating system error log and there are no other errors in your SQL Server or Backup Server error log, contact Sybase Technical Support for assistance.

## **Additional Information**

When you call Technical Support, have the following information on hand:

- SQL Server version and EBF Rollup level
- Backup Server version.
- SQL Server and Backup Server error logs
- Text of all error messages
- Operating system error log

#### **Severity Level**

16

### **Error Message Text**

#### Release 10.0 and Later

Backup Server has detected a SQL Server error.

#### Release 4.9.x and Earlier

```
Read on dump device '%.*s' failed, vsn=%ld
return=%d status=%ld. Please consult the SQL
Server error log for more details.
```

#### Explanation

For SQL Server release 10.0 and later, Error 3203 is displayed when a problem occurs with a remote procedure call to the Backup Server. In this case, the error is on the SQL Server side.

For SQL Server releases previous to 10.0, this message is displayed when an error occurs when doing a load from a SQL Server dump.

The error message output includes:

- *vsn* the virtual socket number.
- *return* the return value: 0 means successful; -2 means failure.
- *status* the ending status, displayed in decimal value. The most common value is 524288 which usually means an I/O error.

#### Some causes are:

- Write protect is turned on (it must be off when loading a tape because the device is opened read/write).
- No dump exists on the media being accessed (for example, a blank tape or the wrong device specified).
- The media that contains the dump is not readable.

## Action

## Release 10.0 and Later

- Check the SQL Server and Backup Server error logs to determine the cause of the error being sent from the SQL Server.
- Test the connection between the Backup Server and the SQL Server by logging into the SQL Server through isql and typing:

```
1> execute backupserver...sp_ps
2> go
```

where *backupserver* is the name of your Backup Server. This executes **sp\_ps** on the Backup Server.

## **Release 4.9.x and Earlier**

- Make sure write protect is turned off (it must be off when loading a tape because the device is opened read/write).
- At the operating system level, check the contents of the tape.

## **Additional Information**

If you need to call Technical Support, have the following information on hand:

- SQL Server version and EBF Rollup level
- Backup Server version
- SQL Server and Backup Server error logs
- Text of all error messages
- Operating system error log

#### **Severity Level**

21

### **Error Message Text**

```
I/O failed while attempting to clear pages starting at page %ld of disk number %ld.
```

#### Explanation

During a load database, create database, or alter database command, SQL Server initializes the non-allocated database pages. Error 3225 occurs if an I/O error occurs during this process.

This error can be caused by an operating system problem such as insufficient disk space (for a file system) or a disk going offline, or can be caused by a SQL Server problem.

#### Action

Check your operating system error logs to determine whether you have run out of disk space or are having other problems that could explain the I/O errors.

#### **Additional Information**

If you need to call Technical Support, have the following information on hand:

- SQL Server version and EBF Rollup level
- Backup Server version
- SQL Server and Backup Server error logs
- Text of all error messages
- Operating system error log

## **Undo and Redo Errors**

This section contains error messages pertaining to SQL Server undo and redo commands.

#### **Severity Level**

21

### **Error Message Text**

```
Process %d was expected to hold logical lock on page %ld.
```

#### Explanation

This error occurs when SQL Server, while committing or aborting a transaction, attempts to release a lock on the page displayed by the error message and the page is not locked.

This error may occur during recovery as well as during run time.

Common causes of this error are:

- Software failures.
- When SQL Server was performing a rollback, it expected a logical lock on the page displayed in the error message but the lock could not be found.
- When SQL Server attempted to deallocate a page, it expected a logical lock on that page but one did not exist.
- SQL Server tries to undo a page deallocation.
- SQL Server tries to roll back a user transaction which has executed a stored procedure that required reresolution. Reresolution is required for stored procedures that reference objects which have been dropped and re-created between executions. Refer to "Procedure Re-resolution" in the *SQL Server Troubleshooting Guide* for more information about reresolution.

#### Action

Look for other errors in the Sybase error log as well as in your operating system error log to find out the specific source of the problem and clear those errors first, as they might be the actual cause of the error. Restarting SQL Server will release any locks that active transactions might still hold. However, this will not clear the root cause of this error.

If there are no other errors in the Sybase error log or in your operating system error log, contact Sybase Technical Support for assistance.

## **Additional Information**

For more information about transactions, see transaction in the *Transact-SQL User's Guide*.

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Hardware error log
- Output of sp\_lock and sp\_who before restarting SQL Server
- Text of all error messages

## **Recovery Errors**

This section contains error messages pertaining to SQL Server recovery.

## **Severity Level**

22

## **Error Message Text**

```
During recovery initialization, page %ld was
encountered. This page belongs to object %ld, not
the log.
```

## Explanation

During recovery, SQL Server scans the transaction log until the last page is found. During this scan, if a page is found that is allocated to *syslogs* but the object ID on the page header is not the same as that of *syslogs*, Error 3403 occurs.

Situations that may cause this error include the following:

- A bad page allocation or write occurred due to a SQL Server problem.
- SQL Server went down after the allocation page was updated but before the transaction log page was written. The database was later rebuilt without the transaction log pages being cleared and entries from the old log still exist.

This is a serious error and you will have to rebuild the affected database either using **bcp** or from clean backups.

## Action

If you have a good backup, restore the database from your backup:

- 1. Drop the database. If the drop fails, follow the instructions in "How to Drop a Database When drop database Fails" in the *SQL Server Troubleshooting Guide*.
- 2. Create a database for load. Make sure the database you create has sizes as least as large as those in *sysusages* for the original database (and that all other *sysusages* values match the original values). See the entry for create database for load in the *SQL Server Reference Manual* for releases 4.8 and later, or the entry for create database for releases for releases earlier than 4.8.
- 3. Load the database from backup. (See load database in the *SQL Server Reference Manual*.)

If you do not have a good backup, copy the data from the corrupted table into a new (dummy) table or into a file. Then rename your old, corrupted table and copy the data back into a new table using the original name. For information about doing this, refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

To prevent some occurrences of this error in the future, checkpoint each database that is being used before shutting down SQL Server.

### **Additional Information**

See the *System Administration Guide* or *SQL Server Reference Manual* for complete information about how to safely create, dump, load, and re-create databases.

### Severity Level

10

### **Error Message Text**

### Release 10.0 and Later

Database '%.\*s' (dbid %d): Recovery failed. Check the SQL Server errorlog for further information as to the cause.

### Release 4.9.x and Earlier

Database '%.\*s' (dbid %d): Recovery failed. Please contact Technical Support for further instructions.

### **Explanation**

This error occurs when SQL Server could not complete the recovery of the specified database when starting.

You cannot use the database until whatever caused the error has been corrected because SQL Server marks the database suspect.

### Action

To determine why recovery failed, examine the SQL Server error log for any errors prior to the 3414 error. It is important to identify the errors before the first occurrence of the 3414 error because subsequent attempts to start SQL Server may not give the detailed error information you need to diagnose the problem.

If you do not have sufficient information to recover from the previous errors, you can recover from known, clean backups or contact Sybase Technical Support for assistance.

### Severity Level

21

**Error Message Text** 

### Release 10.0 and Later

Not enough deses to open '%.\*s'.

### Release 4.9.x and Earlier

Not enough descriptors to open model.

### Explanation

This error occurs when SQL Server could not complete recovery of the specified database during SQL Server startup.

Error 3418 is called for the following system databases which existed in *sysdatabases* prior to starting recovery:

- model (release 4.2 and later)
- sybsecurity (release 10.0 and later)
- sybsystemprocs (release 10.0 and later)

Possible reasons for 3418 include:

- During database recovery, SQL Server failed to find the corresponding row in *sysdatabases* (State 1 or State 3 is returned with the 3418 error).
- SQL Server was unable to open a database for recovery (State 2 is returned with the 3418 error).

When Error 3418 occurs, SQL Server shuts down automatically. Since the affected databases are required for SQL Server to start successfully, manual intervention is required to start SQL Server when this error occurs.

### Action

To determine why recovery failed, examine the SQL Server error log.

Contact Sybase Technical Support for assistance with the manual intervention required to start SQL Server and recover from this error.

# **Severity Level**

21

# **Error Message Text**

```
Transaction (%ld, %d) not found in transaction table.
```

# Explanation

This error occurs when, during SQL Server recovery, load transaction, or load database, an end (commit or rollback) transaction log record was found that does not have a corresponding begin transaction record. Therefore, the transaction could not be rolled back or committed and recovery or load could not complete for that database.

You cannot use the affected database until whatever caused the error has been corrected because SQL Server marks the database suspect.

## Action

If the error occurred during recovery, determine which database had the error by looking at your SQL Server error log.

If you have a clean backup, restore your database using that backup.

If you do not have a clean backup, call Sybase Technical Support.

# **Additional Information**

If you need assistance from Sybase Technical Support, have the following information available when you call:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all error messages

# **Error and Exception Handling Errors**

This section contains error messages pertaining to error handling and exception handling.

# **Severity Level**

10

# **Error Message Text**

Duplicate key was ignored.

# Explanation

This error occurs when you try to insert duplicate key values in a column or columns which have a unique index that was created with the option ignore\_dup\_key.

SQL Server ignores the commands in a transaction which try to insert a duplicate index. The rest of the commands in the transaction will complete normally.

# Action

No action is necessary.

If you want the row inserted, use one of the following procedures:

- Eliminate the unique index.
- Remove duplicate values. If the duplicate values are a result of incorrect data, change the data and rerun the transaction.

# **Additional Information**

See create index in the *SQL Server Reference Manual* for more information about indexes and index options.

# **Severity Level**

10

# **Error Message Text**

Command has been aborted.

# Explanation

This error occurs when SQL Server is unable to complete a command. Error 3621 may be the last message displayed by your application even though another message was sent by SQL Server when the error actually occurred. The rest of the commands in the transaction will complete normally.

Some situations that may cause Error 3621 include:

- Division by 0 in your application
- Entry of a duplicate row during use of your application
- Other illegal functions in your application

## Action

Look for an error before the 3621 error occurs. Resolve the problem that is causing the error before the 3621 error.

If client applications receive this error, ask your SQL Server system administrator to check the SQL Server error log.

# **Severity Level**

17

# **Error Message Text**

The transaction was aborted because it used too much stack space. Either use sp\_configure to increase the stack size, or break the query into smaller pieces. spid: %d, suid: %d, hostname: %.\*s, application name: %.\*s

## **Explanation**

This error occurs when a process requires more stack space than is currently available in the SQL Server. Because this error occurs before any stack corruption, SQL Server does not shut down. Instead, SQL Server rolls back the transaction in progress and returns control to the user process.

A SQL Server process usually encounters this error for one of two reasons:

- The stack size is too small. The information placed on the stack is valid but its total size exceeds the specified stack size. This is often caused by a single, complex query which requires more than the average amount of stack space.
- An incorrectly handled error results in recursive calls which eventually fill the stack, regardless of its size.

If this error occurs frequently, there is increased risk a process may overflow its stack area and corrupt another process' stack. This situation will cause SQL Server to shutdown with the error:

kernel: \*\*\* Stack Guardword corrupted

See "Stack Guardword Error" on page 2-526 for more information about the stack guardword error.

► Note

Error 3626 is only raised in SQL Server release 4.9.1 and later releases.

#### Action

If this error occurs with a transaction which contains a where or wherein clause, you may need to rewrite the query. See "Error 404" on page 2-23 for additional actions which may be required.

If this error occurs because a complex query has run out of stack area, you can resolve it in one of two ways:

- Break up the query into smaller pieces.
- Use the following query to increase the stack size:

```
1> sp_configure "stack size", new_value
2> go
1> reconfigure
2> go
```

Shut down and restart SQL Server to make these changes take effect.

When changing the stack size on SQL Server, consider the following facts:

- The stack size should always be an even increment of 2K (2,048 bytes). If a stack size is specified that is not divisible by 2K, sp\_configure will round it up to the next 2K increment.
- Increase the stack size in small increments to avoid making it too large. If the stack size is too large, you will lose cache space and you may be unable to start SQL Server.
- SQL Server requires more memory for each configured user connection and device when the stack size increases. Therefore, increasing the stack size without increasing the memory configuration parameter will result in less total cache space for SQL Server and could affect performance. See "SQL Server Memory Issues" in the *SQL Server Troubleshooting Guide* for more information.

# drop Errors

This section contains error messages pertaining to drop commands.

### **Severity Level**

11

## **Error Message Text**

```
Cannot drop the %S_MSG '%.*s', because it doesn't exist in the system catalogs.
```

### Explanation

This error occurs when you try to drop an object that is not found in at least one system table where SQL Server expected to find it.

Error 3701 can occur due to the following circumstances:

- The object you are trying to drop does not exist
- Inconsistent system catalog tables
- A SQL Server problem has occurred

### Action

Make sure you entered the object name correctly in your drop command.

If you are entering the object name correctly and the drop fails with Error 3701, try to re-create the object. If your create command displays the message:

```
Msg 2714, Level 16, State 1:
Line 1:
```

There is already an object named 'object\_name' in the database.

then your system tables are incorrect with regard to this object. If this occurs, run dbcc checkcatalog and dbcc checkdb. Then call Sybase Technical Support. They will probably be able to help you delete the object that is causing the error. However, because other objects may reference that object, deleting it cleanly may be difficult. If this is the case, you may need to recover from backups.

# Additional Information

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- Text of all error messages
- Output of dbcc checkdb and dbcc checkcatalog

### **Severity Level**

16

### **Error Message Text**

```
Cannot drop the %S_MSG '%.*s', because it is currently in use.
```

### Explanation

This error occurs when SQL Server tries to drop a database, user table, procedure, rule, default, trigger, or view that is currently in use.

Error 3702 can occur in the following circumstances:

- During a drop command such as drop database, drop table, drop procedure, or drop rule
- During a dbcc dbrepair (database\_name, dropdb) command

Error 3702 can occur for the following reasons:

- The object you are trying to drop is in use
- The process using the object was abnormally terminated
- Data corruption has occurred
- A SQL Server problem has occurred

### Action

Use sp\_who to determine whether the database or object name in your drop command is currently in use. If it is in use, contact the user shown in the sp\_who output.

If the process using the object was abnormally terminated (for example, with an operating system kill command), the object may not have been completely removed from the SQL Server system tables. The process would remain invisible while keeping the object open and in use. This prevents the drop command from succeeding. In this case, shutting down and restarting SQL Server should clear the 3702 error. After you restart SQL Server, run dbcc checkcatalog to determine whether corruption has occurred.

If Error 3702 persists, call Sybase Technical Support.

# Additional Information

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- Text of all error messages
- Output of dbcc checkdb, dbcc checkcatalog, and dbcc checkalloc for the database that contains the object named in the error message

### **Severity Level**

11

### **Error Message Text**

```
Cannot drop the %S_MSG with object-id %ld in database %d, because it doesn't exist in the system catalogs.
```

### Explanation

This error occurs when SQL Server fails to drop a database, user table, procedure, rule, default, trigger, or view because the object being dropped does not exist in the appropriate system table.

Error 3703 can occur during a drop command such as drop database, drop table, drop procedure, or drop rule.

Error 3703 occurs with the following states.

State	Meaning
1	During a <b>drop trigger</b> command, if SQL Server fails to find the trigger or the target table of the trigger, Error 3703 occurs with State 1.
2	During a <b>drop index</b> command, if SQL Server fails to find the object ID of the table for which the index is being dropped in <i>sysobjects</i> , Error 3703 occurs with State 2.
4	During any <b>drop</b> command, if SQL Server fails to find a <i>sysobjects</i> entry for the object being dropped, Error 3703 occurs with State 4.
5	The <i>sysconstraints</i> table has one row for each referential and check constraint associated with a table or column. You can drop constraints using the alter table command. If, during such an alter table command, SQL Server fails to find an entry for the constraint in <i>sysconstraints</i> , Error 3703 occurs with State 5.
6	The <i>sysreferences</i> table has one row for each referential integrity constraint declared on a table or column. If the object being dropped is a referential integrity constraint, SQL Server searches <i>sysreferences</i> for the referential integrity constraint ID. If the ID is not found, Error 3703 occurs with State 6.
7	When SQL Server fails to drop a database, user table, procedure, rule, default, trigger, or view that is not some other object's subsidiary part (for example, dropping a trigger because its associated table is being dropped), and the object ID is not found in <i>sysobjects</i> , Error 3703 occurs with State 7.

State	Meaning
8	If the database being dropped has referential dependencies, SQL Server tries to drop the references from the other databases. If the ID of the reference from the other database is not found in <i>sysreferences</i> , Error 3703 occurs with State 8.

## Action

Recovering from Error 3703 requires manually modifying one or more system tables. Call Sybase Technical Support for assistance in determining which tables to modify and how to do it.

# **Additional Information**

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- Text of all error messages

# **Transaction Errors**

This section contains error messages pertaining to transaction processing.

## **Severity Level**

21

# **Error Message Text**

```
Can't unsplit logical page %ld in object
'%S_OBJID' in database '%S_DBID' - both pages
together contain more data than will fit on one
page.
```

## Explanation

Data and index pages are split when SQL Server determines that a row cannot be inserted due to insufficient room on the existing page. During transaction rollback (either by a user request or during database recovery), SQL Server tries to undo the changes done by the transaction. Error 3904 occurs when SQL Server fails to unsplit pages which were split during the transaction.

Error 3904 is caused by a SQL Server problem.

When this error occurs, data integrity may have been compromised.

## Action

- 1. Use the procedure in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond to the page number in the error message text.
- 2. If the object encountering the error is **not** a system table (a system table's object ID is less than 100), continue with step 3.

If the object with the error is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index.

If the object with the error is a system table and the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption, but it may be necessary to restore from clean backups.

3. For a user table, if the index ID is **not** 0, run the **sp\_helpindex** procedure on the index to get information for rebuilding it. Drop the index.

Re-create the index. This clears the corruption in most cases.

Run dbcc checktable on the table to verify data integrity. If errors occur, call Sybase Technical Support.

4. For a user table, if the index ID is 0, run dbcc checktable on the table to verify data integrity. If errors occur, restore the database from clean backups.

# **Additional Information**

See the *System Administration Guide* and *SQL Server Reference Manual* for information about:

- Dropping and re-creating indexes
- How to safely create, dump, load, and re-create databases

# **Data Server Errors**

This section contains general error messages pertaining to SQL Server.

# **Severity Level**

14

# **Error Message Text**

Login failed.

# Explanation

This error occurs when a login request is rejected by SQL Server. Some reasons for Error 4002 are:

- User name does not exist in *syslogins* (names are case-sensitive).
- An incorrect password has been entered (passwords are case-sensitive).
- The account for the user trying to log in is locked (SQL Server 10.0 and later).
- The user is trying to log in to a SQL Server that:
  - does not exist, or
  - the user does not have access to, or
  - is not running.
- SQL Server is in single-user mode and only "sa" can login.
- SQL Server shutdown is in progress and only "sa" can login.
- Remote SQL Server names are translated from the login record to a site ID. If the SQL Server name is not found in *sysservers*, then the login fails.
- A remote login is being attempted and *sysremotelogins* is missing.
- A remote login is being attempted and the user cannot be found in *sysremotelogins*.
- For "sa," your Replication Server password is different from your SQL Server password.

## Action

Determine whether any of the reasons listed above are causing the 4002 error. If you have checked each of these and you still cannot login to SQL Server, call Sybase Technical Support.

Following are instructions for checking on the above reasons:

- Login name does not exist in syslogins (names are case-sensitive).
  - When SQL Server is first installed, the only login is "sa" and it has no password.
  - If using isql, make sure the user is specifying the -U option and that the correct name and case are being specified.
  - Check *syslogins* to make sure the user is spelling the login name correctly:

```
1> select * from syslogins
2> go
```

- An incorrect password has been entered (passwords are case-sensitive).
  - If using isql, make sure the user is specifying the -P option and the correct password and case are being specified.
  - If a user cannot remember his or her password:

**For SQL Server 10.0 and later**, change the password using sp\_password (only the System Security Officer can change someone else's password):

```
1> sp_password caller_passwd,
2> new_passwd, login_name
3> go
```

where *caller\_passwd* is the password of the login account that is currently executing **sp\_password** and *new\_passwd* is the new password for *login\_name* (the name of the user whose password you are changing).

**For SQL Server 4.9.x and earlier**, look up the user's old password by selecting from *syslogins* to see what it was. To change it, type:

```
1> sp_password old_password,
2> new_passwd, login_name
3> go
```

- If you cannot remember your "sa" password and you cannot get into SQL Server with any other login:

**For SQL Server 10.0 and later**, run dataserver -p to generate a new password for a System Security Officer's account. You start SQL Server with -p, immediately log in to SQL Server with the new random password and execute **sp\_password** to reset

your password to a more secure one. Refer to the *SQL Server Utility Programs* document for your platform for details.

For SQL Server 4.9.x and earlier, you must rebuild your Server.

• For SQL Server 10.0 and later, the account for the user trying to login is locked. To check this, type:

```
1> sp_displaylogin login_name
2> go
```

where *login\_name* is the login name of the user. Check the value for the *status* column:

- If the value is 1, the password is less than 6 characters or NULL.
- If the value is 2, the account is locked.
- If the value is 3, the account is expired.

To unlock a user account:

```
1> sp_locklogin login_name "unlock"
2> go
```

(Only System Administrators and System Security Officers can use sp\_locklogin to lock and unlock logins.)

- The user is trying to log into a Server that does not exist or that the user does not have access to. Check the user's environment file to make sure the value of *DSQUERY* is defined to be the Server the user is trying to log in to.
- When SQL Server is in single-user mode, only "sa" can login.
  - Refer to "Returning SQL Server to Multi-User Mode" in the *SQL Server Troubleshooting Guide*.
  - Check your runserver file to make sure it does **not** have the **-m** option specified.
- When SQL Server shutdown is in progress, only "sa" can login. Tell the user to wait until SQL Server has come back up.
- Remote Server names are translated from the login record to a site ID. If the Server name is not found in *sysservers*, then the login fails. Use sp\_helpserver to list the Servers in *sysservers*.
- If a remote login is being attempted and *sysremotelogins* is missing, then the login fails. To check this, type:

1> sp\_helpdb sysremotelogins
2> go

• If a remote login is being attempted and the user cannot be found in *sysremotelogins*, then the login fails. To check this, type:

1> sp\_helpremotelogin
2> go

• For "sa," your Replication Server password is different from your SQL Server password. Make sure the "sa" passwords are the same.

# **Additional Information**

Refer to the *System Administration Guide* for information about logins and the procedures mentioned in this writeup.

# dump and load Errors (Continued)

This section contains error messages pertaining to the  $\mathsf{dump}$  and  $\mathsf{load}$  utilities.

### **Severity Level**

17

### **Error Message Text**

```
Unable to continue logged version of DUMP
TRANsaction. Please free up some space in database
%d or use the NO_LOG option.
```

### Explanation

This error occurs when SQL Server runs out of space in the transaction log during a dump transaction command.

#### Action

You have two options for recovering from Error 4204:

- Extend the transaction log segment.
- Perform a dump transaction with the no\_log option. dump transaction with no\_log removes the inactive part of the log without making a backup copy and without recording the procedure in the transaction log (performs a checkpoint). You should only do this if you cannot extend the transaction log segment.

Each of these options is described below.

### **Extending the Transaction Log Segment**

- 1. Read "Transaction Log Management" in the *SQL Server Troubleshooting Guide* before deciding whether to increase the transaction log size.
- 2. To get information about the log segment, enter a command like the following and replace "test" with the name of the database in the error message:

	1> use test 2> go	
	1> sp_helpsegment logsegment 2> go	
segment	name sta	atus
2	logsegment	0

device	size	free_pages
dev1	1.0MB	512
table_name	index_name	indid
syslogs	syslogs	0

3. If you have more space in the current log device fragment, increase the amount of space allocated for the transaction log with commands like the following:

```
1> use master
2> go
1> alter database database_name
2> log on device_name = additional_space
3> go
```

where:

- *database\_name* is the name of the database for which the log device fragment is to be extended.
- *device\_name* is the name of the database device on which to locate the database extension.
- *additional\_space* is the number of megabytes to extend the log device fragment.
- 4. If you do **not** have any more space in the current log segment, you can use **sp\_extendsegment** to extend the range of the log segment to another database device (the database must already exist on that device fragment before you run these commands):

```
1> use database_name
2> go
1> sp_extendsegment logsegment, database_name,
2> device_name
3> go
where:
```

- *database\_name* is the name of the database where the segment is to be extended.
- *device\_name* is the name of the additional database device for the log segment.

# Recovery Using dump transaction with no\_log

If you were not able to recover from the 4204 error by extending the transaction log segment, perform the following steps:

1. For SQL Server release 4.8 and later, use the following method to determine approximately how many pages the transaction log occupies:

```
1> use database_name
2> go
1> select data_pgs (8, doampg)
2> from sysindexes where id=8
3> go
```

where "8" is the ID for *syslogs*. The result is the number of data pages (2K pages on most operating systems) that the transaction log occupies.

# ► Note

The query results may be inaccurate by as many as 16 pages, but using the query is much faster than counting the rows in the transaction log.

For SQL Server release 4.2, use the following method to determine how many rows are in the *syslogs* table:

```
1> use database_name
2> go
1> select count(*) from syslogs
2> go
```

- 2. Dump the transaction log using the no\_log option.
- 3. Repeat step 1. If the transaction log now occupies significantly fewer pages, continue with the next step now.

If, however, there are still a large number of pages in the *syslogs* table, an outstanding transaction is probably preventing the log from being cleared. (See "Transaction Log Management" in the *SQL Server Troubleshooting Guide* for more information about the transaction log.) If this is the case, restart SQL Server and repeat step 2. When SQL Server starts and this database is recovered, the outstanding transaction is rolled back, allowing the log to be cleared by a subsequent dump transaction.

4. Dump the database now, because the **no\_log** option prevents subsequent changes recorded in the log from being used to recover from a media failure. You must do a **dump database** now to

ensure the recoverability of subsequent database modifications using transaction logs.

► Note

This database dump is not required if your backup and recovery procedures do not include saving the transaction logs for media failure recovery.

If you are concerned that your transaction log is not large enough for your SQL Server, read "Transaction Log Management" in the *SQL Server Troubleshooting Guide* before deciding to increase the transaction log size. If you choose to increase the size, see the *SQL Server Reference Manual* for more information about the alter database command used to increase log size. Remember that once they have been increased, log and database sizes cannot be decreased.

# **Severity Level**

16 State 2

# **Error Message Text**

# Release 10.0 and Later

Dump transaction is not allowed because a nonlogged operation was performed on the database. Dump your database or use dump transaction with truncate\_only until you can dump your database.

# **Release 4.9.x and Earlier**

DUMP TRANsaction is not allowed while the select into/bulk copy option is enabled: use DUMP DATABASE, or disable the option with sp\_dboption.

# Explanation

This error occurs when you have performed a minimally logged operation such as bcp in or select into on the database, followed by a dump transaction to a dump device. The dump transaction is not allowed because your database contains data that cannot be recovered from the log. The log could not be used to recover the database properly. For instance, if a user does a select into followed by an update, data for the select into is not recorded in the log, so the update cannot be recovered from the log.

## ► Note

A minimally logged operation is one in which changes to the data itself are not logged to the transaction log. All changes to the allocation structures resulting from the operation are logged to the transaction log.

Note that simply enabling the select into / bulkcopy option does not, in itself, cause this condition. The option must be set and a minimally logged operation executed before this error message is reported during a dump transaction to a dump device.

Dumping the database creates a new save point on which subsequent transaction logs are based. Therefore, dumping the database ensures that the minimally logged changes are recoverable and dumping the transaction log to a device is once again allowed.

### Action

Perform a dump database. Disabling the select into/bulkcopy option using sp\_dboption will not allow you to dump the log. You must dump the database to clear the 4207 error condition. You may want to disable the select into/bulkcopy option to prevent any further nonlogged operations in your database.

You may need to use dump transaction with truncate\_only until you can perform a full database dump. If you must do this, you will lose the ability to recover up-to-the-minute changes in data in case of a media failure.

# **Severity Level**

21

# **Error Message Text**

DUMP TRANsaction encountered an unexpected end to the log page chain. Last page in log as scanned: %ld.

## Explanation

During a dump transaction command, SQL Server deallocates contiguous, inactive log pages after their contents have been dumped (it truncates the log). During the page deallocation process of dump transaction, if SQL Server believes it is trying to deallocate the last page of the transaction log, Error 4219 is raised.

This error can be caused by:

- A bad page chain or allocation (syslogs corruption)
- A SQL Server problem

## Action

1. Determine whether there is corruption in *syslogs*:

```
1> use database_name
2> go
1> dbcc checktable (syslogs)
2> go
1> dbcc tablealloc (syslogs)
2> go
```

where *database\_name* is the name of the database for which you are performing the dump transaction command.

If no problems were reported, call Sybase Technical Support.

If problems were reported, try to fix them using the following steps.

- 2. Put the affected database in single-user mode:
  - If the database is *master*, use the procedure described in "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide*, and then continue with this procedure.

- If the database is not *master*, use the sp\_dboption stored procedure to put the affected database in single-user mode:

```
1> use master
2> go
1> sp_dboption database_name, single, true
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

3. Run the dbcc tablealloc command to try to fix the corruption (dbcc tablealloc can fix allocation problems but not page chain problems):

```
1> use database_name
2> go
1> dbcc tablealloc (syslogs, full, fix)
2> go
```

If no problems were reported, try your dump transaction command again.

If problems were reported by dbcc tablealloc or dump transaction, call Sybase Technical Support.

- 4. If no problems were reported, turn off single-user mode in the database:
  - If the database is *master*, use the procedure in "Returning SQL Server to Multi-User Mode" in the *SQL Server Troubleshooting Guide*.
  - If the database is not *master*, use the following procedure:

```
1> use master
2> go
1> sp_dboption database_name, single, false
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

## **Additional Information**

Refer to the *SQL Server Reference Manual* and the *System Administration Guide* for additional information about dbcc tablealloc. Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Output of dbcc checktable and dbcc tablealloc
- Text of all error messages

### **Severity Level**

16

## **Error Message Text**

Specified file '%.\*s' is out of sequence. Current time stamp is %S\_DATE while dump was from %S\_DATE.

### Explanation

During transaction log dump loading, SQL Server checks the time stamp on the dumped database and each dumped transaction log to see that the sequence is correct. If the load sequence of the transaction logs is incorrect, Error 4305 occurs and the load fails.

If you are missing a transaction log dump, you can only restore up to the transaction log dump before the missing dump.

In addition, you cannot load multiple transaction log dumps that have been created with the no\_truncate option. The no\_truncate option assumes that nothing further will be done with the transaction log after that point: it is only useful for recovering a database after failure of the data device. If you have used the no\_truncate option and your database is still okay, do a dump database as soon as possible after that.

This error is serious: it stops the load processing and the load transaction fails.

### Action

Load the log backups in the same sequence in which they were made. Check your transaction log dumps to see if one was loaded out of order. If you find the correct dump, load it and continue with the rest of the dumps, in order.

### **Additional Information**

For more information, see recovery and load transaction in the *System Administration Guide* and in the *SQL Server Reference Manual*.

### **Severity Level**

16

### **Error Message Text**

```
There was activity on database since last load,
unable to load. Must restart load sequence with
the load database to continue.
```

### Explanation

If user transactions modify the transaction log between transaction dump loads to a database, Error 4306 occurs and the load fails.

#### Action

Begin again with a load database or proceed to use the database without recovering all transactions.

Before loading a database, lock the database by using sp\_dboption to set the no chkpt on recovery, dbo use only, and read only options to TRUE. This will prevent other users from making changes to the database during the load process. Set the options **before loading**, as setting them after loading invalidates the load sequence from the start.

```
1> sp_dboption mydb, "no chkpt on recovery", true
2> go
1> sp_dboption mydb, "dbo use only", true
2> go
1> sp_dboption mydb, "read only", true
2> go
1> use mydb
2> go
1> checkpoint
2> go
```

#### ► Note

You cannot use this procedure to lock *master*.

After the last transaction log is loaded successfully, set the above options back to FALSE and checkpoint the database.

2-368

### **Additional Information**

For more information, see recovery and load transaction in the *System Administration Guide* and in the *SQL Server Reference Manual*.

# **Bulk Copy Utility Errors**

This section describes errors generated by the Bulk Copy Utility (bcp).

### **Severity Level**

16

### Error Message Text

```
You cannot run the non-logged version of bulk copy in this database. Please check with the DBO.
```

### Explanation

This error occurs when the select/into bulkcopy option is set to false and you use "fast" bulk copy into a table that has no indexes or triggers.

#### ► Note

By default, the select into/bulkcopy option in newly created databases is set to the same as that in *model*.

### Action

You have the following two options for recovering from this error.

#### Turn On select/into bulkcopy

Use the sp\_dboption stored procedure to set the select/into bulkcopy option to true:

```
1> sp_dboption database_name,
2> "select into/bulkcopy", true
3> go
1> use database_name
2> go
1> checkpoint
2> go
```

Once a minimally logged operation such as "fast" bulk copy runs in the database, you are not allowed to dump the transaction log to a device, because unlogged changes are not recoverable. Instead, you should do a dump database as soon as possible to restore recoverability and allow transaction dumps to devices again. Create an Index or Trigger for the Table

Put an index or trigger on the table you are trying to copy into. This causes inserts to be logged. You can create an insert trigger that does not perform any actions.

When you copy into a table that has indexes or triggers, a slower version of bcp is automatically used. The slow version, which does log data inserts in the transaction log, can cause the transaction log to become very large. You may need to use dump transaction with truncate\_only until you can perform a full database dump. If you must do this, you will lose the ability to recover up-to-the-minute changes in data in case of a media failure. You may also consider using smaller batch sizes.

## alter database Errors

This section contains error messages pertaining to the alter database command.

### **Severity Level**

16

### **Error Message Text**

```
Could not find enough space on disks to extend database %.*s.
```

### Explanation

This error occurs when an **alter database** command refers to a SQL Server device that is completely full.

If the device is not full but has less space than the alter database command requests, the database will be extended to fill up the available space on the device and no error message will be displayed. Instead, you will see the usual status message from alter database, but it will report fewer pages than you requested.

This error message does not usually appear in the error log. However, if the exhausted device also uses up the available physical disk drive space as well as the space allocated for SQL Server, kernel messages declaring the file system full will occur in the error log. If this happens, fix the file system problem as well as the lack of space for SQL Server.

### Action

To recover from Error 5006, do one of the following steps:

- Use a device name in the alter database command which is large enough. Use sp\_helpdevice and sp\_helpdb to find out which devices have enough room.
- Add a new device to SQL Server with the disk init command. Then refer to this device with the alter database command.
- Although there is no particular performance benefit to keeping a database entirely on a single device, if you need to do this at your site, use the following procedure:
  - 1. Back up all the databases on the device by dumping them to a valid dump device with the dump database command.
  - 2. Drop the device with sp\_dropdevice.
  - 3. Initialize a larger device with disk init.

- 4. Create the old databases on the new device using standard procedure. It is important to create the databases exactly like the old ones or errors will occur. See "Error 2558" on page 2-274 or "Backup Server" in the *SQL Server Troubleshooting Guide* for more information.
- 5. Reload each database with the load database command.

See the *System Administration Guide* or *SQL Server Reference Manual* for complete information about how to safely create, dump, load and re-create databases.

## **Severity Level**

16

## **Error Message Text**

Cannot extend the MASTER database onto any device other than 'master'. The ALTER DATABASE was aborted.

### Explanation

This error occurs when you try to extend the *master* database onto a device other than the master device.

## ► Note

It is recommended that you keep user objects off the master device. By keeping user objects off the master device, you decrease the chance that you would need to extend the *master* database. In addition, if you should ever need to rebuild the master device, it will be easier if it does not contain user objects.

### Action

System 10 (or later) users can move any "home-grown" system procedures that start with "sp\_" to *sybsystemprocs* (by dropping them from *master* and creating them in *sybsystemprocs*).

Extend the *master* database only if absolutely necessary! If you are sure you must increase the *master* database size and have no room on the current master device, use the following procedure to remove user databases from the master device.

## **Move User Databases**

- 1. Dump the user databases with the dump database command.
- 2. Rename the dumped databases on the master device with sp\_renamedb.
- 3. Re-create the databases with their original names on another device with create database. Be sure they are created exactly like the old databases, to avoid 2558 and other errors. See "Error 2558" on page 2-274 for more information.

- 4. Load the dumps with load database.
- 5. Check the databases in their new location to make sure the load was successful (that is, perform a simple query with isql), and if everything loaded successfully, drop the old databases from the master device.

You can now try to increase the size of the *master* database on the master device with the alter database command.

Increase Master Device Size

If the master device contains only the *master* database and the master device is too small, then use the following procedure:

♦ WARNING!

Altering the master device is extremely risky! Avoid it if at all possible. Be familiar with the recovery methods in "System Database Recovery" in the *SQL Server Troubleshooting Guide* in case you lose your *master* database or master device.

- 1. Back up the *master* database with the dump database command.
- 2. Use the buildmaster utility to build a new master device with enough extra space so that you will never need to increase the master device again. When buildmaster completes, a new *master* database will exist on the new master device.
- 3. Expand the size of the new *master* database with the alter database command, if necessary, so that it matches the size of the dumped *master* database.
- 4. Execute the following command in isql:

```
1> select name, high from master..sysdevices
2> where name = "master"
3> go
```

and note the "high" value for the master device.

- 5. Start SQL Server in single-user mode. Refer to "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide* for instructions.
- 6. Load the dump of the *master* database. Test the database to make sure it is functioning correctly before deleting any extra backups of *master*.
- 7. Allow updates to the system catalog:

```
    sp_configure, allow, 1
    go
    reconfigure with override
    go
    Reset the "high" value in master..sysdevices:
```

```
1> begin transaction
2> go
1> update master..sysdevices
2> set high = <value of high from step 4>
3> where name = "master"
4> go
```

- 9. If the previous update affected only one row, commit the transaction.
- 10. Restart SQL Server.
- 11. Turn off allow updates:

```
1> sp_configure, allow, 0
2> go
1> reconfigure
2> go
```

## **Disk Errors**

This section contains error messages pertaining to disk activation and system administrator disk commands.

### **Severity Level**

16

### **Error Message Text**

I/O error during disk initialization. PHYSNAME
'%.\*s' may be incorrect or %ld (VSTART + SIZE) may
exceed the size of the device or permissions may
be wrong for the device.

### Explanation

This error occurs when a disk init command fails. It can occur for the following reasons:

- Incorrect physical device name
- Permission problem
- Incorrect size
- Virtual device number not unique
- UNIX kernel not configured for asynchronous I/O

### ► Note

This error only occurs in SQL Server releases before 10.0.2.

### Action

Check each section below, make any needed changes and then run the command again.

**Incorrect Device Name** 

Make sure you have the correct physical device name. See the *Installation Guide* or the *System Administration Guide* for instructions.

### **Permission Problem**

If SQL Server was started by the "sybase" account, it must have permission to write to the raw partition or to create a file in the directory specified by disk init. Raw partitions used by SYBASE should always be owned by the "sybase" account. Correct the permissions problem and execute the command again, using a different *vdevno*. See "Virtual Device Number" on page 2-381 for more information.

### **Incorrect Size**

The size parameter of the disk init command must be specified in units of 2K blocks (size of most data pages). There are 512 2K blocks in a megabyte. Use the formula:

```
#_of_Megabytes * #_of_data_pages_per_Megabyte = #_of_2K_blocks
```

For example, to initialize a 10MB SYBASE device, specify 5120 for the size:

10MB \* 512 = 5120

► Note

For Stratus users: data pages are 4K and there are 256 4K blocks in a megabyte; for a 10MB device, specify 2560 (10MB \* 256 = 2560).

To determine the number of 2K blocks to use for a raw partition on a UNIX system, convert the size of the partition, in sectors, to 2K blocks. The size of a sector varies on different UNIX systems. For example, it is 512 bytes on Sun, so use sectors/4 for the size parameter. It is 1024 bytes on HP, so use sectors/2. Also, see the information below about virtual device numbers. You must restart SQL Server or use a different virtual device number when you reissue the disk init command.

See the *Installation Guide* or *System Administration Guide Supplement* for more information about SYBASE and your operating system's partitions.

► Note

Not all operating systems have partitions as described above.

Virtual Device Number

The vdevno parameter of the disk init command must be unique. Also, if a disk init command fails, SQL Server saves the vdevno used, even though the disk init command failed. To make the vdevno available again, restart SQL Server. Use sp\_helpdevice to determine which virtual device numbers have been used.

The available number of virtual devices is equal to the number of configured devices minus one. For example, if the devices configuration parameter is set to 10, then there are nine available devices, 1 through 9.

## **UNIX Kernel**

If the UNIX kernel (on SunOS 4.x/Sun Solaris 1.x systems only) is not configured for asynchronous I/O, the 5115 error occurs only if your master device is on a UNIX file, and you are trying to do a disk init on a raw partition. Have your UNIX administrator rebuild the kernel to allow for asynchronous I/O. See "Error 823" on page 2-88 for more information.

### Severity Level

16

### **Error Message Text**

DISK INIT encountered an error while attempting to open/create the physical file. Please consult the SQL Server error log (in the SQL Server boot directory) for more details.

### Explanation

This error occurs when a disk init command fails while trying to initialize an operating system file or a raw partition for use as a database device. It can occur for the following reasons:

- Incorrect physical device name
- Permission problem
- Incorrect size specified (for raw devices, when you request initialization of a device larger than the size of the raw partition)
- Not enough room on the device

### Action

Check each section below, make any needed changes and then run the command again.

### **Incorrect Device Name**

Make sure you have the correct physical device name. See the *Installation Guide* or the *System Administration Guide* for instructions.

### **Permission Problem**

If SQL Server was started by the "sybase" account, it must have permission to write to the raw partition or to create a file in the directory specified by disk init. Raw partitions used by SYBASE should always be owned by the "sybase" account. Correct the permissions problem and execute the command again, using a different vdevno. See "Virtual Device Number" on page 2-381 for more information. ► Note

SQL Server should **not** be started by "root." If it is started by *root*, you will get Error 5123 if the device or file is owned by "sybase."

**Incorrect Size** 

When you request initialization of a raw device, make sure the size you specify is not larger than the size of the raw partition.

The size parameter of the disk init command must be specified in units of 2K blocks (size of most data pages). There are 512 2K blocks in a megabyte. Use the formula:

```
#_of_Megabytes * #_of_data_pages_per_Megabyte = #_of_2K_blocks
```

For example, to initialize a 10MB SYBASE device, specify 5120 for the size:

10MB \* 512 = 5120

► Note

For Stratus users: data pages are 4K and there are 256 4K blocks in a megabyte; for a 10MB device, specify 2560 (10MB \* 256 = 2560).

To determine the number of 2K blocks to use for a raw partition on a UNIX system, convert the size of the partition, in sectors, to 2K blocks. The size of a sector varies on different UNIX systems. For example, it is 512 bytes on Sun, so use sectors/4 for the size parameter. It is 1024 bytes on HP, so use sectors/2. Also, see the information below about virtual device numbers. You must restart SQL Server or use a different virtual device number when you reissue the disk init command.

See the *Installation Guide* or *System Administration Guide Supplement* for more information about SYBASE and your operating system's partitions. For example, on UNIX, there is a size limitation of 2GB for a device file.

► Note

Not all operating systems have partitions as described above.

## **Additional Information**

Refer to the *System Administration Guide* for information about disk init.

# **Reconfiguration Errors**

This section contains error messages pertaining to the reconfigure command.

### **Severity Level**

16

### **Error Message Text**

```
Do not recommend ad hoc updates to system catalogues, use override option to force this configuration.
```

### Explanation

This informational message occurs in one of the following situations:

- You try to enable the allow updates configuration parameter using the reconfigure command without specifying the with override option.
- You try to execute the reconfigure command when the allow updates variable is already set to one.
- You try to set a configuration parameter to a value which SQL Server recognizes as likely to interfere with performance or smooth operation.

Leaving the allow updates variable enabled means that you allow direct updates to the system tables. Therefore, any user who can log on as Sybase System Administrator ("sa" account or "sa" role) can update the system tables directly with ad hoc queries, and can create stored procedures that update the system catalog. Incorrect changes to the system tables can cause unrecoverable database corruption or data loss.

### ♦ WARNING!

The system catalog should *never* be directly modified except when specifically directed to do so by Technical Support or by this manual, as this can result in unrecoverable database corruption and data loss.

## Action

If you are aware of the danger and have a specific reason to enable the variable allow updates, for example if either Technical Support or this manual suggested it, use this procedure to enable allow updates:

1. Issue the command reconfigure with override after enabling the allow updates variable:

```
1> sp_configure "allow updates", 1
2> go
1> reconfigure with override
2> go
After making the personal changes to the
```

2. After making the necessary changes to the system tables, be sure to disable the allow updates variable again:

```
1> sp_configure "allow updates", 0
2> go
1> reconfigure
2> go
```

- 3. Restart SQL Server to initialize changes.
- ♦ WARNING!

Always perform step 2 to prevent ad hoc queries on the system tables which might result in unrecoverable database corruption or data loss.

### **Severity Level**

16

### **Error Message Text**

```
Unable to run with specified memory size of %ld.
Please see the System Administration Guide for
more information on how to calculate this number.
```

### Explanation

This error occurs when a reconfigure command fails because SQL Server determined there was not enough memory available for subsequent starts of SQL Server.

This error is a result of using sp\_configure to make configuration changes that require more total memory than the specified value of the memory variable. Because reconfigure must be executed before the changes take effect, SQL Server has a chance to verify that there will be enough memory to fulfill all the requirements. See the *SQL Server Reference Manual* for more information.

Due to the complexity of SQL Server memory allocation, reconfigure cannot always detect potential problems from over-configuring. If such a problem goes undetected, SQL Server will not start. See "System Database Recovery" in the *SQL Server Troubleshooting Guide* for information on how to proceed.

### Action

This error occurs when the value of the memory variable is too low to allocate all of the requested resources. You can avoid this problem by increasing the value of the memory variable with sp\_configure or by decreasing the value of other variables with sp\_configure.

Once these changes are made, execute reconfigure. If the command completes without error, the problem is probably resolved. See "SQL Server Memory Issues" in the *SQL Server Troubleshooting Guide* for more information about SQL Server's memory use.

The calculations performed by the reconfigure command are internal to SQL Server and do not account for machine resources, which may or may not be adequate to support the SQL Server configuration.

## **Process Kill Errors**

This section contains error messages pertaining to killing SQL Server processes.

### **Severity Level**

17

### **Error Message Text**

```
Unable to do cleanup for the killed process received Msg %d.
```

### Explanation

Upon termination of any SQL Server process, SQL Server attempts to clean up by releasing resources used by the terminated process. Error 6103 occurs when SQL Server is unable to complete this cleanup.

Error 6103 usually occurs after some other error or when a SQL Server process is aborted. This error can be serious as it can cause data corruption or an abnormal SQL Server shutdown.

Error 6103 is usually caused by the errors that occurred before it or by a SQL Server problem.

### Action

Review the error log to see what errors may have occurred before the 6103 error. Use the information in this manual to recover from those errors. If errors are not covered in this manual, call Technical Support for assistance.

To recover from Error 6103 after reviewing other errors in the error log, shut down and restart SQL Server.

If you cannot perform a SQL Server shutdown, kill the Server process at the operating system level.

Run dbcc checkdb for the database that was being accessed by the killed process to determine whether data corruption has occurred. If other errors are displayed when dbcc checkdb is run, refer to the writeups in this manual for those errors to determine how to recover from them.

If the 6103 error occurs again or resources (such as locks) are not released, call Sybase Technical Support.

# **Timestamp Errors**

This section contains error messages pertaining to timestamps.

### Severity Level

21

### **Error Message Text**

```
Page timestamp value falls between the old and new timestamps from log. Page #=%ld, object id = %ld, page timestamp=%04x %08lx. Log: old timestamp=%04x %08lx, new timestamp=%04x %08lx.
```

### **Explanation**

The objective of recovery algorithms in SQL Server is to bring all the databases to a consistent state: that is, each transaction is either completely applied to the database or no part of it is applied. (Database recovery occurs at SQL Server startup as well as when a load database or load transaction command is processed.)

Whenever a page is modified during run time, the page timestamp before the change and the page timestamp after the change are recorded in the log record for that modification. By comparing the timestamp on the page at recovery time with that of the timestamps in the log record, SQL Server can tell whether the page is in the state before or after the change was made to it.

Error 6902 occurs during SQL Server recovery if the timestamp of the page which is being recovered is neither the old log timestamp nor the new log timestamp. This is a fatal error and causes the recovery process to fail.

Error 6902 is probably a result of a more serious underlying error. Common causes of this error are:

- Data corruption during normal processing
- Abnormal shutdown of SQL Server during a transaction update
- Hardware errors

### Action

Recover your database from backups or call Sybase Technical Support.

To prevent this error from occurring in the future, use one of the following sections, depending on what caused the 6902 error.

## **Shutdown Method**

shutdown with nowait or abnormal shutdown of SQL Server should be avoided, if possible, after a heavy update period in your databases:

• Use shutdown to shut down SQL Server instead of shutdown with nowait after a period of heavy update activity in your databases.

Or:

• checkpoint each database that is being used before shutting down SQL Server.

## Hardware Errors

Check the Sybase error log to see if there are other indications of hardware problems, such as kernel messages reporting I/O errors.

Check the hardware error log or diagnostics utilities for I/O errors.

### Using dump transaction with no\_log

Using dump transaction with no\_log can, in some circumstances, result in a 6902 error, especially when used after a heavy update period. Therefore, do not use dump transaction with no\_log unless your transaction log is 100 percent full. Try using dump transaction with truncate\_only first and use the no\_log option only as a last resort.

If you find that the no\_log option is used periodically, you might not be performing database or transaction log dumps often enough, or you may not have enough space allocated to the database or transaction log. Check for any occurrences of dump transaction with no\_log in any of your scripts or procedures and replace them with dump transaction with truncate\_only. For more information about dump transaction with no\_log, see "Truncating the Transaction Log" in the SQL Server Troubleshooting Guide.

### ♦ WARNING!

Use dump transaction with no\_log only when there are no active users in the database. For more information about dump transaction, see the *System Administration Guide*.

### **Additional Information**

For more information about recovery, see recovery in the *System Administration Guide* and in the *SQL Server Reference Manual.* 

2-394

## **Text Manager Errors**

This section contains error messages pertaining to the Text Manager.

### **Severity Level**

16

### Error Message Text

```
WRITETEXT with no log is not valid at this time.
Database %s must first be configured for this.
```

### Explanation

This error occurs when you try to execute the writetext command or dbwritetext routine in a database which does not have the sp\_dboption option select into/bulkcopy enabled. Because these operations do not log the changes to your data, the option must be enabled.

Often, writetext and dbwritetext run against a temporary table. The select into/bulkcopy option must be enabled in *tempdb* before these will run successfully.

### Action

Enable the sp\_dboption option select into/bulkcopy on all databases affected by the query just before you perform this nonlogged operation. To do this, the System Administrator or Database Owner can use the following steps:

- 1. Enable the option using this procedure:
- 1> use master 2> go 1> sp\_dboption database\_name, 2> "select into/bulkcopy", true 3> go 1> use database\_name 2> go 1> checkpoint 2> go 2. If you want to verify that the change is active, use this
- procedure:

1> sp\_helpdb database\_name
2> go

3. Execute the nonlogged operation.

- 4. When the nonlogged operation is complete, dump the database using normal system administration procedures.
- ♦ WARNING!

After you have executed a nonlogged operation, do not perform a regular transaction log dump — although the procedure may appear to succeed, if you try to load the dump you may have only empty text pages! Once you have made nonlogged changes to your database, you must perform a dump database, since these changes cannot be recovered from transaction logs.

In release 10.0 and later releases, you cannot perform a dump transaction after a nonlogged operation.

5. Return the database to its original condition by disabling the select into/bulkcopy option:

```
1> use master
2> go
1> sp_dboption database_name,
2> "select into/bulkcopy", false
3> go
1> use database_name
2> go
1> checkpoint
2> go
Repeat step 2 above if you want to verify that the change is
```

active.

### **Additional Information**

For more information about writetext and dbwritetext see the SQL Server Reference Manual and the Open Client DB-Library Reference Manual.

For more information about database dumps and recoverability, see the *System Administration Guide*.

## **Distributed Database Network Errors**

This section contains error messages for distributed database network code.

### **Severity Level**

18

### Error Message Text

```
Can't open a connection to site '%S_SRVID'. See
the error log file in the SQL Server boot
directory.
```

### Explanation

This error occurs when SQL Server is unable to execute a remote procedure call (RPC) on a remote SQL Server, Backup Server, or Open Server application.

► Note

In this writeup, "Server" is used to indicate a SQL Server, Backup Server, or Open Server application.

### Action

To determine why this error is occurring, perform the following steps:

1. Check whether the remote Server is running by issuing an isql statement similar to the following:

% isql -Sname\_of\_remote\_server

If you receive the following message:

Operating-system error: Invalid argument DB\_LIBRARY error: Unable to connect: SQL Server is unavailable or does not exist.

then the remote Server may not be running. Log in to the remote Server machine and check whether the Server is running. If it is not running, start it. Check whether you receive the following message after issuing the isql statement:

```
Operating-system error:
No such file or directory
DB_LIBRARY error:
Could not open interface file.
```

If you do, this indicates that your SYBASE environment variable (for UNIX, OS/2 and Novell users) or your SYBASE logical name (for OpenVMS users) is not set to the correct directory.

The value of the SYBASE parameter has to be set to the full path name of the *SYBASE* home directory where the interfaces (or *sql.ini*) file can be found. The value of the SYBASE parameter specifies the device as well as the path for *SYBASE* home. If you move the *SYBASE* directory structure to a new device, make sure the device is included in the SYBASE environment variable string (UNIX, OS/2, and Novell) or in the SYBASE logical name string (OpenVMS).

- 2. Check the error log for both the sending Server and the receiving Server. If other errors exist, refer to the writeups in this manual for information about recovering from them.
- 3. If, after resolving any problems found in steps 1 and 2, you still get Error 7205, call Sybase Technical Support with the information listed below.

## **Additional Information**

For more information about managing remote SQL Servers, see the chapter "Managing Remote Servers" in the *System Administration Guide*.

Before calling Technical Support, have the following information available to fax or otherwise provide to Sybase Technical Support:

- SQL Server release and EBF Rollup level
- SQL Server (and Backup Server if appropriate) error log for both the local and remote Servers
- Operating system error log
- sp\_helpserver and sp\_configure output
- Contents of interfaces file
- Text of all error messages

### **Severity Level**

18

### **Error Message Text**

```
Can't open a connection to site '%S_SRVID' because 'set up channel' message was not acknowledged.
```

### Explanation

This error occurs when SQL Server is unable to execute a remote procedure call (RPC) on a remote SQL Server, Backup Server, or Open Server application because the remote Server did not acknowledge the connection message sent by the local Server.

One reason for this error is that the physical connection has died before a response was received. The transaction is then aborted.

It may also occur if two access attempts occur within such a short time frame that the first physical connection established has not yet timed out.

► Note

In this writeup, "Server" is used to indicate a SQL Server, Backup Server, or Open Server application.

### Action

To find out why this error is occurring, perform the following steps:

1. When automatic timeouts are enabled, a timeout occurs when there is no activity for one minute. If you believe timeouts are occurring, consider disabling automatic timeouts by issuing the following command:

```
1> sp_serveroption server_name "timeouts", false
2> go
```

where *server\_name* is the name of the SQL Server for which you are disabling automatic timeouts.

2. Check the error log for both the sending Server and the receiving Server. If other errors exist, refer to the writeups in this manual for information about recovering from them.

3. If, after resolving any problems found in steps 1 and 2, you still get Error 7211, call Sybase Technical Support.

## Additional Information

For more information about managing remote SQL Servers, see "Managing Remote Servers" in the *System Administration Guide*.

Before calling Technical Support, have the following information available to fax or otherwise provide to Sybase Technical Support:

- SQL Server release and EBF Rollup level
- SQL Server (and Backup Server if appropriate) error log for both the local and remote Server
- Operating system error log
- sp\_helpserver and sp\_configure output
- Contents of interfaces file
- Text of all error messages

### **Severity Level**

11

### **Error Message Text**

```
Can't close connection %d on site '%S_SRVID' because it is not active.
```

### Explanation

This error occurs when SQL Server is unable to close a connection for a remote site on a SQL Server, Backup Server, or Open Server application because the remote connection is not active. *%d* is the logical connection channel number and *%S\_SRVID* is the remote SQL Server name.

State	Meaning
2	Error 7212 occurs with State 2 when SQL Server is attempting to close the logical connection to the remote site and the logical connection is not found.
3	Error 7212 occurs with State 3 when SQL Server is sending an attention packet to the remote connection and the logical connection is not found.

One reason for this error is that the physical connection died before a response was received. The transaction is then aborted.

#### Action

To find out why this error is occurring, perform the following steps:

1. When automatic timeouts are enabled, a timeout occurs when there is no activity from a remote user for one minute. If you believe timeouts are occurring, consider disabling automatic timeouts by issuing the following command:

```
1> sp_serveroption server_name, "timeouts", false
2> go
```

where *server\_name* is the name of the SQL Server for which you are disabling automatic timeouts.

- 2. Check the error log for both the sending Server and the receiving Server. If other errors have been raised, refer to the writeups in this manual for information about recovering from them.
- 3. If, after resolving any problems found in steps 1 and 2, you still get Error 7212, call Sybase Technical Support.

## **Additional Information**

For more information about managing remote SQL Servers, see "Managing Remote Servers" in the *System Administration Guide*.

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server (and Backup Server if appropriate) error log for both the local and remote Server
- Operating system error log
- sp\_helpserver and sp\_configure output
- Contents of interfaces file
- Text of all error messages

#### **Severity Level**

11

#### **Error Message Text**

```
No site name received in server login packet.
Server login rejected.
```

#### Explanation

This error occurs when executing a remote procedure call (RPC) from a SQL Server which has not been given a name via the **sp\_addserver** system procedure. The SQL Server name referred to here is stored in the *sysservers* system table as well as in the *@@servername* global variable.

#### Action

Assign a name to the SQL Server initiating the RPC by using the sp\_addserver procedure. Type the following while using the SQL Server that initiated the RPC (see the *SQL Server Reference Manual* for information about restrictions on valid SQL Server names):

1> sp\_addserver sql\_server\_name, local
2> go

In order for the change to take effect, you have to restart SQL Server. After restarting, you can verify that the SQL Server initiating the RPC has been given a name by typing the following:

```
1> select @@servername
2> go
```

or by examining the SQL Server start message. The output will report information on the SQL Server initiating the RPC (the local server). If the output returned is NULL, then the local SQL Server has not yet been assigned a name.

#### Additional Information

For more information, see:

- "Server Is Unnamed" on page 2-525 of this manual
- remote access in the System Administration Guide
- "Identifiers" in the SQL Server Reference Manual

#### Severity Level

18

#### Error Message Text

Can't send to site '%S\_SRVID'.

#### Explanation

This error occurs when you try to execute a remote procedure call (RPC) on a remote SQL Server, Backup Server which is not currently running. The remote SQL Server's name can, however, be found in the local interfaces file.

► Note

In this writeup, "Server" is used to indicate a SQL Server or a Backup Server.

#### Action

Perform the following steps:

1. Check whether the remote Server is running by issuing an isql statement similar to the following:

```
% isql -Sname_of_remote_server
```

If you receive the following message:

```
Operating-system error:
Invalid argument
DB_LIBRARY error:
Unable to connect: SQL Server is unavailable or
does not exist.
```

then the remote Server may not be running. Log in to the remote Server machine and check whether the Server is running. If it is not running, start it.

2. Check whether you receive the following message after issuing the isql statement:

Operating-system error: No such file or directory DB\_LIBRARY error: Could not open interface file. If you do, this indicates that your SYBASE environment variable (for UNIX, OS/2 and Novell users) or your SYBASE logical name (for OpenVMS users) is not set to the correct directory.

The value of the SYBASE parameter has to be set to the full path name of the *SYBASE* home directory where the interfaces file can be found. The value of the SYBASE parameter specifies the device as well as the path for *SYBASE* home. If you move the *SYBASE* directory structure to a new device, make sure the device is included in the SYBASE environment variable string (UNIX, OS/2, and Novell) or in the SYBASE logical name string (OpenVMS).

# Additional Information:

For more information about managing remote SQL Servers, see "Managing Remote Servers" in the *System Administration Guide*.

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server (and Backup Server if appropriate) error log for both the local and remote Server
- Operating system error log
- sp\_helpserver and sp\_configure output
- Content of interfaces file
- Text of all error messages

### **Severity Level**

11

# **Error Message Text**

Site '%.\*s' not found in sysservers.

# Explanation

This error occurs in one of the following situations:

- When the remote SQL Server name in a remote procedure call (RPC) is not defined in the *sysservers* table on the SQL Server initiating the RPC.
- When you type @@version from isql instead of select @@version.

# Action

Add the remote SQL Server name to the *sysservers* table on the SQL Server that initiated the RPC by typing the following (see the *SQL Server Reference Manual* for information about restrictions on valid SQL Server names):

1> sp\_addserver server\_name
2> go

The easiest way to name a SQL Server is by assigning it the name stored in the DSQUERY environment variable or the name which is used in the **-S** option when the server is accessed via isql. For example:

% isql -SPRODUCTION

In the above example, PRODUCTION is the SQL Server name. You can check to see if the Server has been added to the *sysservers* table by issuing the following command:

```
1> sp_helpserver
2> go
```

If you want to find out the SQL Server release you are using, type select *@@version* instead of *@@version*.

# **Additional Information**

For more information, see sp\_addserver, sp\_helpserver, and "Identifiers" in the *SQL Server Reference Manual*.

#### **Severity Level**

11

#### **Error Message Text**

Site '%s' not found in interfaces file.

#### Explanation

When you execute a remote procedure call (RPC) from one SQL Server to another, the local interfaces file of the initiating SQL Server has to contain information about the remote SQL Server.

Error 7220 occurs in one of the following situations:

- When you try to execute an RPC on a SQL Server which is not referenced in the local interfaces file. Its name can, however, be found in the *sysservers* table.
- When you attempt to use the dump or load commands on SQL Server release 10.0 and later and the Backup Server is not listed in the interfaces file.

#### ► Note

Throughout this writeup, your network addressing file is referred to as your "interfaces file."

### Action

Eliminate the 7220 error by following the appropriate instructions below.

### The Remote SQL Server is Not in the Local Interfaces File

Use one of the following to check the items in the steps below:

- sybinit (for SQL Server release 10.0 and later)
- sybtli (for SQL Server release 4.9.x and earlier)
- sqledit (for OS/2 SQL Servers release 10.0 and later)
- listini and writeini (for OS/2 and NT SQL Servers release 4.x)
- sqledit or sybinit (for NT SQL Servers release 10.0 and later)

► Note For pre-10.0 systems that are not TLI-based (that is, systems that use sockets), edit the interfaces file manually. 1. Check if the local interfaces file has the query entry for the remote SQL Server. 2. Check if the SQL Server name is spelled correctly in the local interfaces file and if the tabs are set correctly at the beginning of each row which contains information about the SQL Server ports. 3. If you cannot find any information about the remote SQL Server in any interfaces file, you have two options. You can either: - Go to a backup copy of the local or remote interfaces file, and add the name of the remote SQL Server to the local interface file manually. Add the query entry for the remote SQL Server to the local interfaces file. Or, you can: Look for lines like the following in your remote SQL Server error log (the example is from a SunOS 4.x system). If you do not have a backup copy of the local interfaces file, check the remote SQL Server's error log for the network name and port numbers at the last restart (that is, at the end of the error log file). See the "interfaces file" entry in SYBASE Installation and *Operations* for more information about SQL Server ports. 00:93/11/08 17:06:01.01 server Opening Master Database ... 00:93/11/08 17:06:01.65 server Loading SQL Server's default sort order and character set 00:93/11/08 17:06:01.91 kernel network name baldrick, type ether, port 2056 00:93/11/08 17:06:02.06 server Recovering database 'master' . . . .

Look for the port number, the network type, and the machine name.

. . .

For SQL Server 10.0 and later, add an entry to the interfaces file like the following:

#### REL100\_PRODUCTION

query tcp ether baldrick 2056 master tcp ether baldrick 2056

For SQL Server release 4.9.x, look for information like the following so you can include more lines regarding the console entry:

00: 94/03/07 11:52:46.25 kernel: Virtual device 0 started using standard unix i/o.

00: 94/03/07 11:52:46.51 kernel: network name sourdough, type sun-ether, port 1035

00: 94/03/07 11:52:46.86 server: Number of buffers in buffer cache: 1184.

00: 94/03/07 11:52:46.98 server: Number of proc buffers allocated: 296.

00: 94/03/07 11:52:47.10 server: Number of blocks left for proc headers: 281.

00: 94/03/07 11:52:47.63 server: Opening Master Database ...

00: 94/03/07 11:52:48.39 server: Loading SQL Server's default sort order and character set

00: 94/03/07 11:52:48.58 kernel: network name sourdough, type sun-ether, port 1034

#### REL491\_PRODUCTION

query tcp sun-ether sourdough 1034 master tcp sun-ether sourdough 1034

Typically, for SQL Server release 4.9.x and earlier, the console entry is equal to the master entry plus 1.

#### WARNING!

Before you add SQL Server's query, master, and console entries to the local interfaces file, check that no other SQL Server has the same port numbers assigned to it in the local interfaces file.

- 4. After changing the interfaces file, try accessing the remote SQL Server through isql or, if it is not running, restart it.
- 5. Also check your hardware error log for possible hardware failures, in case you still cannot reboot or access the remote SQL Server.

The Backup Server is Not Listed in the Interfaces File

The entry for the local Backup Server is created when you install SQL Server. If you have installed a remote Backup Server on another machine, you can choose one of the following options to resolve this error:

- Copy the interfaces file entry for this Backup Server to the interfaces file on the machine where your SQL Server runs.
- Locate the interfaces file on a file system shared by both machines.
- Use sybinit (or the appropriate utility as listed earlier) to update your interfaces file on the machine where your SQL Server runs. Refer to your *SQL Server Installation Guide* for information about sybinit.

# Additional information

For more additional information, see remote access in the *System Administration Guide* and the "interfaces file" entry in the *SQL Server Installation Guide.* 

# **Severity Level**

14

# **Error Message Text**

Login to site '%S\_SRVID' failed.

# Explanation

This error occurs when you execute a remote procedure call (RPC) in one of the following situations:

- A name was not assigned, via sp\_addserver, to the SQL Server which is initiating the RPC.
- A remote login for the SQL Server initiating the RPC was not added to the SQL Server that is receiving the RPC.
- The SQL Server receiving the RPC is not configured for remote access or for enough remote sites or remote connections.
- The SQL Server receiving the RPC does not have an entry in the network addressing (*interfaces* or *sql.ini*) file for the Server initiating the RPC (or the other way around).
- The SQL Server receiving the RPC is in single-user mode and already had one user at the time the RPC was executed.
- The password for the user initiating the RPC is not the same on both SQL Servers.

# Action

Eliminate the 7221 error by determining which of the above situations apply to you and following the appropriate steps:

1. A name was not assigned to the SQL Server which is initiating the RPC.

Use the select @@servername command to check if the SQL Server initiating the RPC has a name assigned to it.

If the select *@@servername* command returns the name of the server which is initiating the RPC, then continue your checking with step 2.

If the output returned by select *@@servername* is NULL, then assign a name to the local SQL Server via the sp\_addserver

command (see the *SQL Server Reference Manual* for information about restrictions on valid SQL Server names):

1> sp\_addserver name\_of\_server\_initiating\_rpc,local
2> go

Restart SQL Server after assigning it a name.

Usually you want to name a SQL Server the same name as is stored in the *DSQUERY* environment variable or the name which is used in the -S option when the server is accessed via isql. For example:

% isql -SPRODUCTION

In the above example, PRODUCTION is the SQL Server name.

2. A remote login was not added to the SQL Server that is receiving the RPC for the SQL Server that is initiating it.

Use sp\_helpremotelogin to check if the receiving SQL Server has a remote login for the SQL Server initiating the RPC. If the output returned by sp\_helpremotelogin contains information about the remote SQL Server, then continue with step 3).

If the output returned by sp\_helpremotelogin does not contain information about the remote SQL Server, then add a remote login to the remote SQL Server for the SQL Server that is initiating the RPC by typing the following (on the remote SQL Server):

1> sp\_addremotelogin name\_of\_server\_initiating\_rpc
2> go

3. The SQL Server receiving the RPC is not configured for remote access or for enough remote sites or remote connections.

Check the remote SQL Server's error log for more information on why the RPC initiated by the local SQL Server did not succeed. See remote access in the *System Administration Guide* for more information about how to configure SQL Server for more remote sites or remote connections (using sp\_configure) and consult this manual for more information on other RPC errors.

4. The SQL Server receiving the RPC does not have an entry in the interfaces file for the Server initiating the RPC (or the other way around).

Add the required entries for the SQL Server initiating the RPC to the interfaces file of the receiving Server. Use one of the following utilities or copy the entries from the interfaces file of the initiating Server and make sure that they have a tab in front of each entry.:

- sybinit (for SQL Server release 10.0 and later)
- sybtli (for SQL Server release 4.9.x and earlier)
- sqledit (for OS/2 SQL Servers release 10.0 and later)
- listini and writeini (for OS/2 and NT SQL Servers release 4.x)
- sqledit or sybinit (for NT SQL Servers release 10.0 and later)

# ► Note

For pre-10.0 systems that are not TLI-based (that is, systems that use sockets), edit the interfaces file manually.

5. The SQL Server receiving the RPC is in single-user mode and already has one user at the time the RPC is executed.

Wait until the receiving SQL Server is not being used anymore and then execute the RPC or restart the receiving SQL Server in multi-user mode.

6. The password for the user initiating the RPC is not the same on both Servers.

Make user passwords the same for the remote SQL Server.

# **Additional Information**

For more information, see:

- sp\_addremotelogin, sp\_addserver, and sp\_configure in the SQL Server Reference Manual
- "Server Is Unnamed" on page 2-525 of this manual
- remote access in the System Administration Guide

# **Severity Level**

16

# **Error Message Text**

Cannot %S\_MSG site '%S\_SRVID'.

### Explanation

This error occurs when SQL Server fails to complete a read or send operation during a remote procedure call (RPC). The failure can occur in a number of states. Each of these states is described below.

#### ► Note

SQL Server uses tokens to establish a dialog request with a client. There are several kinds of tokens to send different signals.

State	Meaning
1	SQL Server uses buffering for sending and receiving packets to and from remote connections. If an error occurs while SQL Server is trying to read the next packet from the remote connection while SQL Server is being run in debug mode, Error 7223 is raised with State 1. <i>%S_MSG</i> is replaced by "read from" and <i>%S_SRVID</i> is the remote SQL Server name.
2	If SQL Server confirms that no token was read from the remote connection, then Error 7223 is raised with State 2. <i>%S_MSG</i> is replaced by "read from" and <i>%S_SRVID</i> is the remote SQL Server name.
3	If an error occurs while SQL Server is sending a token to a remote connection, Error 7223 is raised with State 3. <i>%S_MSG</i> is replaced by "send to" and <i>%S_SRVID</i> is the remote SQL Server name.
4	If an error occurs while SQL Server is sending data to a remote connection, Error 7223 is raised with State 4. <i>%S_MSG</i> is replaced by "send to" and <i>%S_SRVID</i> is the remote SQL Server name.
5	While SQL Server is establishing a network connection to a remote SQL Server, it translates the remote Server name to a Server ID, opens the logical connection, and then logs in. Error 7223 is raised with State 5 when SQL Server fails during one of these operations. <i>%S_MSG</i> is replaced by "send to" and <i>%S_SRVID</i> is the remote SQL Server name.

State	Meaning
6	SQL Server builds a remote procedure call request message in the remote connection send buffer and then sends it. Error 7223 occurs with State 6 when this send fails. <i>%S_MSG</i> is replaced by "send to" and <i>%S_SRVID</i> is the remote SQL Server name.
7	If SQL Server confirms that no data was read from the remote connection, then Error 7223 is raised with State 7. <i>%S_MSG</i> is replaced by "read from" and <i>%S_SRVID</i> is the remote SQL Server name.

### Action

- 1. Check the network connection between the locations of your local and remote SQL Servers. Make sure packets of data can be sent and received. You may need to ask your network administrator for help.
- 2. If you did not find a problem in step 1, call Sybase Technical Support.

### Additional Information

Have the following information ready when you call Sybase Technical Support:

- SQL Server release and EBF Rollup level
- SQL Server error log for both the local and remote SQL Servers
- Text of all error messages
- **sp\_configure** output

#### **Severity Level**

17

#### Error Message Text

#### Release 10.0 and Later

Can't open a connection to site '%S\_SRVID' because SQL Server is not configured for remote access. Ask a user with System Administrator (SA) role to reconfigure SQL Server to allow remote access.

#### Release 4.9.x and Earlier

Can't open a connection to site '%.\*s' because SQL Server is not configured for remote access. Ask your System Administrator to reconfigure SQL Server to allow remote access.

#### Explanation

This error occurs when you try to execute a remote procedure call (RPC) and the initiating or receiving SQL Server is not configured for remote access.

#### Action

Configure the local or remote SQL Server for remote access by changing the remote access configuration parameter via sp\_configure.

```
1> sp_configure "remote access", 1
2> go
1> reconfigure
2> go
```

After enabling the remote access variable and restarting SQL Server, the remote logins, remote sites, remote connections, and pre-read packets configuration parameters will be set to their default values.

If you determine that the default values for remote logins, remote sites, remote connections, and pre-read packets are not adequate for your needs, then the sp\_configure procedure can be used to change these configuration parameters as well.

After issuing the above commands, restart SQL Server so that the configuration change can take effect.

# **Additional Information**

For more information, see remote access in the *System Administration Guide* and sp\_configure in the *SQL Server Reference Manual*.

# Parser Errors

This section contains error messages pertaining to the Parser.

#### **Severity Level**

16

#### Error Message Text

```
Tables '%.*s' and '%.*s' have some exposed names.
Use correlation names to distinguish them.
```

#### Explanation

An exposed name is an unqualified object name which assumes the default owner and database. For example, *table1* (exposed name) and *master.dbo.table1* (qualified name) may **not** be the same table but SQL Server will think you mean the same table if you reference just *table1* as *dbo* in the *master* database. Explicit correlation names allow you to make distinctions between references in a query.

In ANSI 89, the following construction is not allowed:

select \* from table1, table1 [where\_clause]

In releases prior to 10.0, SQL Server handled the select by returning only the first table and ignoring the self-join. In release 10.0, this syntax returns Error 7364.

#### Action

The following syntax, which uses *t1* and *t2* as table correlation names, allows you to perform a self-join:

select \* from table1 t1, table1 t2 [where\_clause]

# **Threshold Errors**

This section contains error messages pertaining to thresholds. Refer to "Thresholds and Transaction Log Management" in the *SQL Server Troubleshooting Guide* and to the *System Administration Guide* for additional information about thresholds.

#### Severity Level

10

#### **Error Message Text**

Could not allocate memory for the threshold task's argument list; threshold task cannot be spawned for database %d, segment %d, free space %ld.

#### Explanation

This error occurs when a stored procedure associated with a threshold was not initiated because there was not enough memory available in procedure cache.

#### Action

Although Error 7401 indicates that the threshold task did not run because not enough memory was available in procedure cache at that time, the threshold task may have run later when enough memory did become available. Before performing the following steps, determine whether the threshold task ran at a later time. If the threshold task did run, the following steps are not necessary.

To perform the threshold task that did not run because of this error:

- 1. First, use the database displayed in the error message.
- 2. Run sp\_helpthreshold *segment\_name* for the segment named in the error message. This will display the stored procedure associated with the threshold.
- 3. Run the stored procedure manually.

To prevent this error from occurring in the future, make sure there is enough memory to run stored procedures associated with a threshold by testing them during the time when your system is most heavily loaded. If there is not enough memory, use sp\_configure to increase the SQL Server procedure cache configuration parameter (increases the percentage of total cache space that is used as procedure cache) or rewrite the stored procedure so it will execute in the procedure cache available for the SQL Server.

# **Additional Information**

Refer to "How SQL Server Uses Memory" in the *SQL Server Troubleshooting Guide* for information about SQL Server memory use.

Refer to the *System Administration Guide* for information about SQL Server thresholds.

#### **Severity Level**

10

#### **Error Message Text**

Could not spawn the threshold handler procedure for database %d, segment %d, when only %ld free pages were left in the segment.

#### Explanation

Thresholds access SQL Server as a detached user. This means that, although a threshold does not access SQL Server through the network, it still requires a user connection. Error 7402 occurs when there are no available user connections when a SQL Server process reaches a threshold and SQL Server tries to execute the associated threshold procedure.

#### Action

To perform the threshold task that did not run because of this error:

- 1. First, use the database displayed in the error message.
- 2. Run sp\_helpthreshold *segment\_name* for the segment named in the error message. This will display the stored procedure associated with the threshold.
- 3. Run the stored procedure manually.

To prevent this error from occurring in the future, use sp\_configure to make sure that there are enough user connections to allow for all clients and the threshold manager. Refer to "Increasing User Connections" in the *SQL Server Troubleshooting Guide* for information about changing the user connections configuration parameter.

#### **Additional Information**

Refer to the *System Administration Guide* for information about SQL Server thresholds.

# **Severity Level**

10

# **Error Message Text**

```
Threshold task could not use database %d, and so cannot execute the threshold procedure for segment %d, free space %ld.
```

### Explanation

A threshold is "owned" by the user who created it, or the last user to modify it with sp\_modifythreshold (see the *SQL Server Reference Manual* for details). Error 7403 occurs when the Threshold Manager tries to use a database and execute a stored procedure and access is denied. The procedure is not executed.

### Action

To perform the threshold task that did not run because of this error:

- 1. First, use the database displayed in the error message.
- 2. Run sp\_helpthreshold *segment\_name* for the segment named in the error message. This will display the stored procedure associated with the threshold.
- 3. Run the stored procedure manually using the account of the user that owns it.

To determine the owner of the procedure and user access to the database, type:

1> sp\_help procedure\_name
2> go
1> sp\_helpuser database\_name
2> go

To prevent this error from occurring in the future, make sure that the owner of each threshold has access to the database to which the threshold applies, and that the user has execute permission on the stored procedure associated with the threshold.

#### **Severity Level**

10

#### **Error Message Text**

```
Could not find row in systhresholds for database %.*s describing segment %d, free space %ld.
```

#### Explanation

This error can occur for either of the following reasons:

- The row in *database.dbo.systhresholds* describing the segment, free space, and procedure name for the segment named in the error message is incorrect. Therefore, the threshold manager was not able to execute the stored procedure associated with the threshold.
- The clustered index on *systhresholds* is corrupt and SQL Server was not able to retrieve the row.

#### Action

Run dbcc checktable (systhresholds) and dbcc tablealloc (systhresholds) to check consistency of the system table. If you have corruption on *systhresholds*, contact Sybase Technical Support for further instructions.

Perform the threshold task that did not run because of this error:

- 1. First, use the database displayed in the error message.
- 2. Run sp\_helpthreshold *segment\_name* for the segment named in the error message. This will display the stored procedure associated with the threshold.
- 3. Run the stored procedure manually.

If the stored procedure is not found, modify the row in *systhresholds* with sp\_modifythreshold to add a stored procedure (threshold action). See the *SQL Server Reference Manual* for information regarding sp\_modifythreshold.

# Additional Information

Refer to the material in this manual for errors reported by  $\ensuremath{\mathsf{dbcc}}.$ 

When calling Technical Support, have the following information available:

- Output of select @@version
- Output of all error messages
- Output of dbcc commands

#### Severity Level

10

#### **Error Message Text**

```
Threshold procedure name is missing from the systhresholds row in database %.*s for segment %d, free space %ld.
```

#### Explanation

This error occurs when a row is found in *database.dbo.systhresholds* for the specified segment, but no procedure name is available, and the threshold task is not able to take appropriate action.

#### Action

Run dbcc checktable (systhresholds) and dbcc tablealloc (systhresholds) to check consistency of the system table. If you have corruption on *systhresholds*, contact Sybase Technical Support for further instructions.

If there is no corruption, modify the row in *systhresholds* with sp\_modifythreshold to add a stored procedure (threshold action). See the *SQL Server Reference Manual* for information regarding sp\_modifythreshold.

Then perform the threshold task that did not run because of this error by running the stored procedure manually.

#### **Additional Information**

Refer to the material in this manual for errors reported by dbcc.

When calling Technical Support, have the following information available:

- Output of select @@version
- Output of all error messages
- Output of dbcc commands

# **Severity Level**

16

# **Error Message Text**

%s: unknown option: %.\*s

# Explanation

**Ict\_admin** is a function used to manage the last-chance threshold (lct) on the log segment of a database. Error 7406 occurs when an invalid option is passed to **Ict\_admin**.

# Action

Use one of the valid options listed below. Note that the options are all strings and require quotation marks.

- lct\_admin ("reserve", number\_of\_log\_pages) returns the number of
  pages required to do a successful dump transaction given the
  number of pages in a log segment.
- Ict\_admin ("lastchance", *database\_ID*) creates a last-chance threshold for a database with log on a separate device. This can be used to create a last-chance threshold for databases that were upgraded from a pre-10.0 SQL server.
- Ict\_admin ("logfull", *database\_ID*) determines if the last-chance threshold has been reached in a specified database. The procedure will return "0" if last-chance has not been reached, and "1" if it has.
- Ict\_admin ("unsuspend", database\_ID) unsuspends all tasks that were put to sleep by activation of last-chance threshold. This option should be used with caution, and users should be aware that additional changes to the database may not be saved in a transaction log due to insufficient space.

#### **Severity Level**

10

#### **Error Message Text**

```
Failed to execute the threshold procedure for database %d, segment %d, free space %ld.
```

#### Explanation

This error occurs when an attempt to execute a stored procedure defined in *systhresholds* fails.

#### Action

To perform the threshold task that did not run because of this error:

- 1. First, use the database displayed in the error message.
- 2. Run sp\_helpthreshold *segment\_name* for the segment named in the error message. This will display the stored procedure associated with the threshold.
- 3. Run the stored procedure manually.

To determine whether you can prevent this error from occurring in the future, review the SQL Server error log for additional messages regarding this error. If there is specific information as to what caused the procedure to fail, take whatever steps are necessary to prevent future occurrences.

### Severity Level

20

# **Error Message Text**

Could not find a dbtable for database %d.

# Explanation

Information about each database is kept in memory in a structure called a *dbtable*. Error 7408 occurs when an attempt to execute the built-in function lct\_admin fails because the database table (database ID) information was either not available or incorrect.

### Action

Verify that you have entered the correct database name (or database ID) in the threshold procedure. If it is not correct, modify the procedure.

This error can occur because the database is not open because the value of the open databases configuration parameter is too low. If the value for open databases is too low, use sp\_configure to raise it.

If the database name is correct, the value of the open databases variable is large enough to accommodate all the system and user databases, and the command still fails with this message, check the SQL Server error log for messages.

To perform the threshold task that did not run because of this error:

- 1. First, use the database displayed in the error message.
- 2. Run sp\_helpthreshold *segment\_name* for the segment for which the error occurred. This will display the stored procedure associated with the threshold.
- 3. Run the stored procedure manually.

#### Severity Level

20

#### **Error Message Text**

This task tried to sleep waiting for space to become available in the log, but could not. The current command is being aborted because no further database changes can be logged.

#### Explanation

The last-chance threshold has been reached in a database. All transactions trying to write to the transaction log should be put to sleep until space is free in the transaction log. SQL Server was unable to put the transaction to sleep, so it has been aborted.

#### Action

Verify that the threshold action defined for the last-chance threshold has executed successfully. If it did not complete, manually execute a **dump transaction** command to create free space in the log. Then rerun the transaction that was aborted. If the transaction still fails, check the SQL Server error log for additional information.

Refer to the *System Administration Guide* for information about lastchance thresholds.

#### **Severity Level**

26

#### **Error Message Text**

%s: function called with invalid argument.

### Explanation

This error occurs when an attempt to execute a threshold action fails because of a problem in a memory structure.

#### Action

If there are additional messages in the SQL Server error log that tell you which threshold was initiated, do the following steps:

1. Select the database:

```
1> use database_name
2> go
```

where *database\_name* is the database which contains the failed threshold.

2. Determine the stored procedure that is associated with the segment for which the error occurred:

```
1> sp_helpthreshold segment_name
2> go
```

3. Run the stored procedure manually.

If you were not able to determine which threshold was initiated by looking at the error log output, run curunreservedpgs in each database to determine which threshold action did not execute properly. If you can determine which threshold was reached (compare the output of curunreservedpgs with the free space for the segment), execute the threshold action manually. See the *SQL Server Reference Manual* for more information about curunreservedpgs.

To prevent this error from occurring in the future, review the SQL Server error log and look for other messages associated with this error. There may be other errors that will tell you why the threshold failed. If more information is available, take whatever steps are necessary to prevent the error from reoccurring.

### **Severity Level**

21

# **Error Message Text**

Table 'systhresholds' in database '%.\*s' is not in its correct sort order. Either the clustered index is missing or there is data corruption in the table.

#### Explanation

This error occurs when an attempt to execute a threshold action fails because of possible corruption on system table *systhresholds* or its clustered index.

#### Action

Run dbcc checktable (systhresholds) and dbcc tablealloc (systhresholds) to check consistency of the system table. If you find corruption in *systhresholds*, call Sybase Technical Support for further instructions.

Have the following information available when you call Sybase Technical Support:

- Output of select @@version
- Output of all error messages
- Output of dbcc commands

# **Severity Level**

10

# **Error Message Text**

Space available in the log segment has fallen critically low in database '%.\*s'. All future modifications to this database will be %S\_MSG until the log is successfully dumped and space becomes available.

### **Explanation**

This message notifies users that the last-chance threshold has been reached. All transactions requiring access to the transaction log will either be suspended or aborted until the appropriate action is taken to clear the transaction log.

#### Action

- 1. Ensure that the action defined by the last-chance threshold procedure was able to execute successfully. If the procedure failed for any reason, manually run dump transaction to create free space in the transaction log.
- 2. Look for errors in the SQL Server error log to determine why the procedure failed.
- 3. Restart any transactions that have been aborted.

### **Additional Information**

Refer to the *System Administration Guide* for information about lastchance thresholds.

#### **Severity Level**

10

#### **Error Message Text**

```
%d task(s) are sleeping waiting for space to
become available in the log segment for database
%.*s.
```

#### Explanation

This message is written to the SQL Server error log by the checkpoint process when the last-chance threshold has been exceeded, and the specified number of processes have been put to sleep.

#### Action

Ensure that the appropriate last-chance threshold action is executed to create free space in the log. If the procedure did not run for any reason, manually create free space in the transaction log with a dump transaction command and check on the cause of the failure by referring to the documentation in this manual.

#### **Additional Information**

Refer to the *System Administration Guide* for information about lastchance thresholds.

# **Severity Level**

10

# **Error Message Text**

```
Threshold procedure '%.*s' returned an error indication (@status=%d) when called for database '%.*s', segment '%.*s', free_space %ld.
```

### Explanation

This error occurs when a threshold procedure is activated but is not able to execute successfully.

#### Action

Check the SQL Server error log for additional information on the reason for the failure. Correct the situation that caused the failure.

Execute the procedure that failed:

1. Select the database:

```
1> use database_name
2> go
```

where *database\_name* is the database which contains the failed threshold.

2. Determine the stored procedure that is associated with the segment for which the error occurred:

1> sp\_helpthreshold segment\_name
2> go

3. Run the stored procedure manually.

#### **Severity Level**

10

#### **Error Message Text**

```
The transaction log in database %.*s is almost
full. Your transaction is being %S_MSG until
space is made available in the log.
```

#### Explanation

This error occurs when the last-chance threshold has been exceeded for the specified database. Each transaction accessing the transaction log will receive this message, indicating whether the transaction will be put to sleep or aborted until log space is available.

#### Action

This message does not ordinarily require action if the transaction was put to sleep and the threshold procedure frees log space. In that circumstance, the sleeping processes will awaken and continue. If the process was aborted, it will be necessary to restart it. If the process is not able to continue, determine why it cannot continue by looking at the error log and manually dump the transaction log.

# dbcc Errors (Continued)

This section contains error messages pertaining to dbcc commands.

#### **Severity Level**

16

#### **Error Message Text**

Data size mismatch occurred while checking TEXT/IMAGE values. The first page for this value is: %ld. There were %ld bytes found, which is different from the expected data length of %ld bytes.

#### **Explanation**

This error occurs when SQL Server detects a mismatch between the expected length of the text/image data of a text column and its actual length. The length of the text/image data is stored on the page header of the text/image data page.

This error usually occurs in one of the following situations:

- When you run dbcc checktable on a table whose text or image column was updated to NULL and then updated again to a non-NULL value via a writetext command which did not commit.
- When you run dbcc checktable or dbcc checkdb and your image or text data is corrupted. If this is the case, check your hardware error log and your operating system error log and determine the cause of the 7902 error.

#### Action for 4.9.1 SQL Server and Later

In order to eliminate the 7902 error, perform the following steps for each table encountering it:

- 1. Record the page numbers reported by Error 7902 for your table.
- 2. Select the offending rows from your table in a separate temporary table, giving the name of the text column for the textptr function:

```
1> select * into #newtable from mytable
2> where convert (int, textptr(text_colname))
3> in (list of page#'s)
4> go
```

The list of page numbers in the above query is the enumeration of all the page numbers specified by the 7902 errors. *text\_colname* is the name of the text column.

3. Update the corrupted text/image data from the old table with the text/image data from the new table:

```
1> begin transaction
2> go
1> update mytable
2> set mytable.text_col = #newtable.text_col
3> from #newtable
4> where mytable.other_col = #newtable.other_col
5> go
```

Make sure that only the required number of rows are updated by the above query and then commit the transaction. *other\_col* in the above query is the name of another column in your table which uniquely identifies a row (like a primary key). If you do not have any primary keys on your old table, run the above query with several search conditions in the where clause:

```
1> update mytable
2> set mytable.text_col = #newtable.text_col
3> from #newtable
4> where mytable.col1 = #newtable.col1
5> and mytable.col2 = #newtable.col2
6> go
```

4. Make sure that only the required number of rows are updated by the above query and then commit the transaction:

```
1> commit transaction
2> go
```

#### Example

You will get the following messages during dbcc checktable:

```
Checking mytable
Msg 7902, Level 16, State 1:
Server 'PRODUCTION', Line 1:
Data size mismatch occurred while checking
TEXT/IMAGE values. The first page for this value
is: 321. There were 12 bytes found, which is
different from the expected data length of 0 bytes.
Msg 7902, Level 16, State 1:
Server 'PRODUCTION', Line 1:
```

```
Data size mismatch occurred while checking
TEXT/IMAGE values. The first page for this value
is: 291. There were 17491 bytes found, which is
different from the expected data length of 19291
bytes.
```

Run the following query in order to select the erroneous rows from *mytable* into *#newtable*:

```
1> select * into #newtable from mytable
2> where convert (int, textptr(text_colname))
3> in (321, 291)
4> go
```

Another Method to Eliminate Error 7902

An alternate method is to select all data from the old table into a new table.

If the index ID is 0 or 255 and you do not have clean backups of your database, copy the data from the corrupted table into a new (dummy) table or into a file. Then rename your old, corrupted table and copy the data back into a new table using the original name. For more information about doing this, refer to "How to Rescue Data from a Corrupted Table" in the *SQL Server Troubleshooting Guide*.

```
♦ WARNING!
```

Some data on this page might be lost if you recover your table using bcp or select into (that is, the corrupted row and rows following it might be truncated and contain the wrong keys). Compare the two tables (old and new) row by row (by joining them on a primary key, for example) to see which rows are different (corrupted).

Before dumping your database, make sure it is working correctly. More specifically, the following commands should be run prior to each dump:

- 1. dbcc checkdb.
- 2. dbcc checkalloc or dbcc checkalloc with the fix option. Refer to the *SQL Server Reference Manual* for information about dbcc checkalloc.

## Action for 4.9.0 SQL Server and Earlier

In order to eliminate the 7902 error, perform the following steps for each table reporting the error:

- 1. Record the page numbers reported by Error 7902 for your table.
- 2. Select the corrupted rows from your table:

```
1> select * from mytable
2> where convert (int, textptr(text_colname))
3> in (list of page#'s)
4> go
```

The list of page numbers in the above query is the enumeration of all the page numbers specified by the 7902 error. *text\_colname* is the name of the text/image column.

Record what is in the text/image column for each row of the output generated by the above query. You need this data in order to update the keys from your text/image column to their correct values.

3. Update the corrupted text/image data from your table with the values of the text/image data from the output generated by the above query:

```
1> begin transaction
2> go
1> update mytable
2> set text_col = value_in_the_above_output
3> where other_col = key_other_col
4> go
```

Make sure that only the required number of rows are updated by the above query and then commit the transaction. *other\_col* in the above query is the name of another column in your table, which uniquely identifies a row (like a primary key). If you do not have any primary keys on your old table, run the above query with several search conditions in the where clause.

*value\_in\_the\_above\_output* is the value your text/image column had in the above output. It can be a character string, a pointer (address number), or NULL.

```
1> update mytable
2> set mytable.text_col = value_in_the_above_output
3> where col1 = key_col1
4> and col2 = key_col2
5> go
```

Make sure that only the required number of rows are updated by the above query and then commit the transaction.

## Prevention

Consider using writetext with log or update instead of writetext with no log. This way, you will not have unlogged changes to your database and will not get 7902 errors if writext with no log did not commit.

#### **Additional Information**

For more information, see select into, convert, and writetext in the *SQL Server Reference Manual* and in the *Transact-SQL User's Guide.* 

## **Severity Level**

16

## **Error Message Text**

```
Table Corrupt: keys in left child is not less than the parent key; check left child page %ld.
```

### Explanation

This serious error is returned by dbcc checktable or dbcc checkdb (which calls dbcc checktable). dbcc checktable checks a table and its indexes to determine whether:

- Pages are linked correctly
- Indexes are properly sorted
- Pointers are consistent
- Page offsets are consistent
- sysindexes pages are correct
- The data row count is equal to the leaf row count for nonclustered indexes

During index level checking, for each key in the parent page, dbcc checktable checks whether the last key in the left child page is less than the parent key, and whether the first key in the right child page is greater than or equal to the parent key. If the last key in the left child is not less than the parent key, Error 7930 occurs.

## Action

- 1. Use the procedure in "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* to identify which table and index correspond to the page number from the error message text.
- 2. If the object encountering the error is **not** a system table (a system table's object ID is less than 100), continue with step 3.

If the object with the error is a system table and the index ID is **not** 0, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index. Then go to step 5.

If the object with the error is a system table and the index ID is 0, contact Sybase Technical Support. They may be able to help you repair the corruption but it may be necessary to restore from clean backups.

3. If the object with the error is a user table and the index ID is **not** 0, determine whether there is a clustered index on the table:

```
1> sp_helpindex table_name
2> go
```

where *table\_name* is the name you determined in step 1.

If there is a clustered index on the table, rebuild it. Then go to step 5.

4. If there is a nonclustered index on the table, rebuild the index using the following steps.

Translate the index ID into an index name:

```
1> use database_name
2> go
1> select name from sysindexes
2> where id = object_ID and indid = index_ID
3> go
```

Drop the index. To ensure that the information needed to re-create the index is available, you may need to run the sp\_helpindex procedure on the index prior to dropping it.

Re-create the index. This clears the corruption in most cases.

5. Run dbcc checktable on the table to verify that the corruption is gone. If corruption still exists, call Sybase Technical Support.

## **Additional Information**

See the *System Administration Guide* and *SQL Server Reference Manual* for information about dropping and re-creating indexes.

#### **Severity Level**

16

#### **Error Message Text**

Table Corrupt: The entry is missing from the OAM for object id %ld indid %d for allocation page %ld.

#### Explanation

This error is raised when SQL Server detects that the allocation pages associated with an object are not recorded in the object allocation map (OAM) for the object. This problem is detected by dbcc checkalloc, dbcc tablealloc, and dbcc indexalloc.

Usually, this error will not cause any operations to fail at run time, so it is acceptable to wait to correct this problem until nonpeak hours. However, do not drop a table with the 7939 error, as subsequent dbcc checkalloc commands may produce 2540 and 2546 errors.

#### ► Note

The instructions below are for fixing 7939 errors once they have occurred. Two easy-to-use strategies exist for **preventing** this error from occurring in the future. Refer to "Preventing Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about these strategies.

#### Action

Occasionally dbcc checkalloc reports this error when no real error condition exists. You can either check to see if these errors are real, or continue with this section and take action to correct them, whether or not they reflect a real allocation error.

Because the process used to discover whether or not the errors are real can be time-consuming, you may want to go now to the "Action" section appropriate for your release level:

- "Action for 10.0 SQL Server and Later" on page 2-449
- "Action for 4.9.1 SQL Server and Later" on page 2-449
- "Action for 4.9.0 SQL Server and Earlier" on page 2-452

### Verifying That the Error Is Real

Run dbcc checkalloc in single-user mode if you suspect the 7939 error messages are incorrect. If the error is in *master*, use the section "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide* to start SQL Server in single-user mode. See the *System Administration Guide* for information about dbcc checkalloc.

### Action for 10.0 SQL Server and Later

If many of these errors are occurring, it is possible to clear them all at once by using the dbcc checkalloc with fix option command. Refer to "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about using dbcc checkalloc.

Action for 4.9.1 SQL Server and Later

### Make Sure Object Exists

Execute the following query to make sure the object exists and is correctly referred to in the system catalog:

```
1> use database_name
2> go
1> select object_name ( objid_from_error_msg )
2> go
```

If an object name is returned, then the error is on an existing, correctly referenced object. If this is the case, go now to the section "Identify Table: User or System Table" on page 2-449.

If a number, or something other than an object name is returned, use the procedure in "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide*.

## Identify Table: User or System Table

Look at the value for the object ID in the error message. If it is 100 or greater, continue with the next section, "Action for User Tables." If the object ID is below 100, it is a system table and requires a different procedure described in the section "Action for System Tables" on page 2-450.

## **Action for User Tables**

If the object ID in the error message is 100 or greater, follow these steps to correct the error:

- 1. Check the value of the index ID in the error message to determine whether it is a table (value is 0) or an index (value is greater than 0).
- 2. Run dbcc tablealloc or dbcc indexalloc, depending on whether the object named in the 7939 error message is a table or an index. Before you run either command, keep the following in mind:
  - dbcc tablealloc corrects this problem on a table or an index, but if the problem is on an index, you can avoid affecting the entire table by using dbcc indexalloc. If you need to minimize the amount of time the table is unavailable, it may be most practical to dbcc indexalloc.
  - These commands can correct the error only when run in the full or optimized mode, and with the nofix option not specified, the default for user tables.

Use the command appropriate for your situation:

For Tables (index ID = 0)	For Indexes (0 < index ID < 255)
1> dbcc tablealloc	<pre>1&gt; dbcc indexalloc (object_ID,</pre>
2> ( <i>object_ID</i> )	2> index_ID)
3> go	3> go

Refer to the *SQL Server Reference Manual* and *System Administration Guide* for additional details on the dbcc tablealloc and dbcc indexalloc commands.

## **Action for System Tables**

If the object ID in the error message is less than 100, follow these steps to correct the error:

- 1. Put the affected database in single-user mode:
  - If the database is *master*, use the procedure in "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide*, and then go to step 2.

- If the database is not *master*, use the sp\_dboption stored procedure to put the affected database in single-user mode:

```
1> use master
2> go
1> sp_dboption database_name, single, true
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

- 2. Check the value of the index ID in the error message to determine whether it is a table (value = 0) or an index (value > 0).
- 3. Run dbcc tablealloc or dbcc indexalloc, depending on whether the object named in the 7939 error message is a table or an index. Keep these facts in mind before executing either command:
  - dbcc tablealloc corrects either a table or an index, but if the problem is on an index, you can avoid affecting the entire table by using dbcc indexalloc. If the table is large or heavily used, if may be most practical to use dbcc indexalloc.
  - These commands correct the error only when run in the full or optimized mode, with the fix option specified, because the default value is nofix on system tables.

Use the command appropriate for your situation:

```
For tables (index ID = 0):
1> dbcc tablealloc (object_ID, full, fix)
2> go
For indexes (0 < index ID < 255):</li>
1> dbcc indexalloc (object_ID, index_ID, full, fix)
2> go
4. Turn off single-user mode in the database:
If the database is master, use "Returning SQL Server to Multi-User Mode" in the SQL Server Troubleshooting Guide.
If the database is not master, use the following procedure:
1> use master
```

```
1> use master
2> go
1> sp_dboption database_name, single, false
2> go
```

```
1> use database_name
2> go
1> checkpoint
2> go
```

Refer to the *SQL Server Reference Manual* and *System Administration Guide* for additional details on dbcc tablealloc and dbcc indexalloc.

## Action for 4.9.0 SQL Server and Earlier

Use either of the following methods to correct 7939 errors:

- Recover the database from a known, clean backup, using your usual SYBASE system administration procedures (see the *System Administration Guide* for more information).
- Call Sybase Technical Support.

#### Severity Level

22

#### Error Message Text

#### Release 4.9.1 and Later

The counts in the OAM are incorrect. This implies that there are entries missing. Run tablealloc utility with the FIX option on the table with the inaccurate OAM counts.

#### Release 4.9.0 and Earlier

Table Corrupt: The counts in the OAM are incorrect. This implies that there are entries missing.

#### Explanation

This error is raised when SQL Server detects that the total page count in the object allocation map (OAM) is different from the actual number of pages in the chain. This problem is detected by dbcc checkalloc, dbcc tablealloc, and dbcc indexalloc.

This error can result in various run-time failures and should be corrected; however, if no other errors are occurring, you can wait to correct the problem until nonpeak hours.

#### ► Note

The instructions below are for fixing 7940 errors once they have occurred. Two easy-to-use strategies exist for **preventing** this error from occurring in the future. Refer to "Preventing Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about these strategies.

#### Action for All SQL Server Versions

Occasionally dbcc checkalloc reports this error when no real error condition exists. You can either check to see if these errors are real, or continue with this section and take action to correct them, whether or not they reflect a real allocation error.

Because the process used to discover whether or not the errors are real can be time-consuming, you may want to go now to the "Action" section appropriate for your release level:

- "Action for 10.0 SQL Server and Later" on page 2-454
- "Action for 4.9.1 SQL Server and Later" on page 2-454
- "Action for 4.9.0 SQL Server and Earlier" on page 2-456

## Verifying That the Error Is Real

Run dbcc checkalloc in single-user mode if you suspect the 7940 error messages are incorrect. If the error is in *master*, use the section "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide* for instructions about how to invoke SQL Server in single-user mode. See the *SQL Server Reference Manual* for information about dbcc checkalloc.

## Action for 10.0 SQL Server and Later

If many of these errors are occurring, it is possible to clear them all at once by using the dbcc checkalloc with fix option command. Refer to "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about using dbcc checkalloc.

Action for 4.9.1 SQL Server and Later

Identify Table: User or System Table

Get the table name that the error occurred on from the original output which indicated this error. If it is a user table, continue with the next section, "Action for User Tables." If it is a system table, it requires a different procedure described in the section "Action for System Tables".

## **Action for User Tables**

Run dbcc tablealloc. This command can correct the error only when run in the full or optimized mode, and with the nofix option not specified (the default for user tables):

1> dbcc tablealloc (table\_name)
2> go

Refer to the *SQL Server Reference Manual* and *System Administration Guide* for additional details on dbcc tablealloc.

#### Action for System Tables

- 1. Put the affected database in single-user mode:
  - If the database is *master*, use the procedure in "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide*, and then go to step 2.
  - If the database is not *master*, use the sp\_dboption stored procedure to put the affected database in single-user mode:

```
1> use master
2> go
1> sp_dboption database_name, single, true
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

2. Run dbcc tablealloc. This command corrects the error only when run in the full or optimized mode, with the fix option specified, because the default value is nofix on system tables:

```
1> dbcc tablealloc (table_name, full, fix)
2> go
```

- 3. Turn off single-user mode in the database:
  - If the database is *master*, use "Returning SQL Server to Multi-User Mode" in the *SQL Server Troubleshooting Guide*.
  - If the database is not *master*, use the following procedure:

```
1> use master
2> go
1> sp_dboption database_name, single, false
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

Refer to the *SQL Server Reference Manual* and *System Administration Guide* for additional details on dbcc tablealloc.

## Action for 4.9.0 SQL Server and Earlier

Use either of the following methods to correct 7940 errors:

- Recover the database from a known, clean backup, using your usual SYBASE system administration procedures (see the *System Administration Guide* for more information).
- Call Sybase Technical Support.

#### **Severity Level**

16

#### **Error Message Text**

```
Page %ld, object %ld, index %d, level %d, was
found with fixed-row status improperly set. That
status has been cleared by DBCC.
```

#### Explanation

This error is reported by dbcc checktable or dbcc checkdb (which calls dbcc checktable) when a page status bit in a page header is set on when it should not be. This bit is normally set on when there are no variable-length columns for this object (the rows are all fixed-length). If dbcc checktable finds that this bit is set on and there **are** variable display rows, the bit is turned off and the 7948 message is displayed.

This is an informational message.

#### Action

dbcc checktable (or dbcc checkdb) corrects this problem so no further action is required.

### **Severity Level**

16

## **Error Message Text**

The number of pages used and unused for %ld index %d on allocation page %ld do not match the counts in the OAM entry.

#### Explanation

This error occurs when SQL Server detects that the allocation page count in the Object Allocation Map (OAM) is different from the actual number of pages in the chain. This problem is detected by dbcc checkalloc, dbcc tablealloc, and dbcc indexalloc.

Usually this error will not cause any operations to fail at run time. Therefore, it is acceptable to wait to correct this problem until nonpeak hours.

```
► Note
```

The instructions below are for fixing 7949 errors once they have occurred. Two easy-to-use strategies exist for **preventing** this error from occurring in the future. Refer to "Preventing Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about these strategies.

## Action for All SQL Server Versions

Occasionally dbcc checkalloc reports this error when no real error condition exists. You can either check to see if these errors are real, or continue with this section and take action to correct them, whether or not they reflect a real allocation error.

Because the process used to discover whether or not the are errors are real can be time-consuming, you may want to go now to the "Action" section appropriate for your release level:

- "Action for 10.0 SQL Server and Later" on page 2-459
- "Action for 4.9.1 SQL Server and Later" on page 2-459
- "Action for 4.9.0 SQL Server and Earlier" on page 2-462

### Verifying That the Error Is Real

Run dbcc checkalloc in single-user mode if you suspect the 7949 error messages are incorrect. If the error is in *master*, use the section "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide* for instructions about how to invoke SQL Server in single-user mode. See the *SQL Server Reference Manual* for information about dbcc checkalloc.

## Action for 10.0 SQL Server and Later

If many of these errors are occurring, it is possible to clear them all at once by using the dbcc checkalloc with fix option command. Refer to "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide* for information about using dbcc checkalloc.

Action for 4.9.1 SQL Server and Later

Make Sure Object Exists

Execute the following query to make sure the object exists and is correctly referred to in the system catalog:

```
1> use database_name
2> go
1> select object_name ( objid_from_error_msg )
2> go
```

If an object name is returned, then the error is on an existing, correctly referenced object. If this is the case, go now to the section "Identify Table: User or System Table" on page 2-459.

If a number, or something other than an object name is returned, use the procedure in "How to Fix and Prevent Allocation Errors" in the *SQL Server Troubleshooting Guide*.

## Identify Table: User or System Table

Look at the value immediately before the "index" value in the error message (the object ID). If it is 100 or greater, the object is a user table and you should continue with the section "Action for User Tables" on page 2-460. If it is below 100, the object is a system table and requires a different procedure described in the section "Action for System Tables" on page 2-460.

## **Action for User Tables**

If the object ID from the error message is 100 or greater, follow these steps to correct the error:

- 1. Check the value of "index" (index ID) in the error message to determine whether it is a table (value is 0) or an index (value is greater than 0).
- 2. Run dbcc tablealloc or dbcc indexalloc, depending on whether the object is a table or an index as determined in step 1. Before you run either command, keep these facts in mind:
  - dbcc tablealloc corrects this problem on a table or an index, but if the problem is on an index, you can avoid affecting the entire table by using dbcc indexalloc. If the table is large or heavily used, if may be most practical to use dbcc indexalloc.
  - These commands can correct the error only when run in the full or optimized mode, and with the nofix option not specified, the default for user tables.

Use the command appropriate for your situation:

For Tables (index ID = 0)	For Indexes (0 < index ID < 255)
<pre>1&gt; dbcc tablealloc (object_ID) 2&gt; go</pre>	<pre>1&gt; dbcc indexalloc (object_ID, 2&gt; index_ID) 3&gt; go</pre>

Refer to the *SQL Server Reference Manual* and *System Administration Guide* for additional details on dbcc tablealloc and dbcc indexalloc.

## Action for System Tables

If the object ID is less than 100, follow these steps to correct the error:

- 1. Put the affected database in single-user mode:
  - If the database is *master*, use the procedure in "How to Start SQL Server in Single-User Mode" in the *SQL Server Troubleshooting Guide*, and then go to step 2.
  - If the database is not *master*, use the sp\_dboption stored procedure to put the affected database in single-user mode:

```
1> use master
2> go
1> sp_dboption database_name, single, true
2> go
```

```
1> use database_name
2> go
1> checkpoint
2> go
```

- 2. Check the value of the "index" (index ID) in the error message to determine whether it is a data page (value is 0) or an index page (value is greater than 0).
- 3. Run dbcc tablealloc or dbcc indexalloc, depending on whether the object named in the 7949 error message is a data page or an index page. Then execute the appropriate command, using the object ID from the error message. Before you run the appropriate command, keep these facts in mind:
  - dbcc tablealloc corrects either a table or an index, but if the problem is on an index, you can avoid affecting the entire table by using dbcc indexalloc. If you need to minimize the amount of time the table is unavailable, it may be most practical to use dbcc indexalloc.
  - These commands correct the error only when run in the full or optimized mode, with the fix option specified, because the default value is nofix on system tables.

Use the command appropriate for your situation:

For Tables (index ID = 0)	For Indexes (0 < index ID < 255)
<pre>1&gt; dbcc tablealloc (object_ID,</pre>	<pre>1&gt; dbcc indexalloc (object_ID,</pre>
2> full, fix)	2> index_ID, full, fix)
3> go	3> go

- 4. Turn off single-user mode in the database:
  - If the database is *master*, use "Returning SQL Server to Multi-User Mode" in the *SQL Server Troubleshooting Guide*.
  - If the database is not *master*, use the following procedure:

```
1> use master
2> go
1> sp_dboption database_name, single, false
2> go
1> use database_name
2> go
1> checkpoint
2> go
```

See the *SQL Server Reference Manual* and *System Administration Guide* for additional details on dbcc tablealloc and dbcc indexalloc.

## Action for 4.9.0 SQL Server and Earlier

Use either of the following methods to correct 7940 errors:

- Recover the database from a known clean backup, using your usual SYBASE system administration procedures (see the *System Administration Guide* for more information).
- Call Sybase Technical Support.

## **Descriptor Manager Errors**

This section contains error messages pertaining to the Descriptor Manager.

## **Severity Level**

26

## **Error Message Text**

```
Keep count of descriptor was expected to be %d. Instead %d was found
```

### Explanation

Session descriptors are the data structures used to manage access to SQL Server objects. The Descriptor Manager controls and synchronizes access to descriptors. It uses a keep count to determine whether a descriptor is in use. This guarantees that a descriptor is protected while it is in use. Error 8201 occurs if the Descriptor Manager determines that the keep count is incorrect.

Error 8201 is caused by a SQL Server problem.

### Action

Shut down and restart SQL Server. This should clear Error 8201.

If the 8201 error persists, contact Sybase Technical Support.

## **Additional Information**

Have the following information ready when you call Sybase Technical Support:

- SQL Server release and EBF rollup level
- SQL Server error log
- Text of all error messages

#### **Severity Level**

26

#### **Error Message Text**

Expected to find the descriptor in %s state

#### Explanation

Session descriptors are the data structures used to manage access to SQL Server objects. As it is used, a descriptor passes through a number of different states:

- **free** The descriptor is on the free resource list. It can be used only by the Descriptor Manager which can turn it into an active descriptor.
- **create** This is the stage of a descriptor between free and active, while the fields are being filled in.
- **active** The descriptor is accessible from outside the Descriptor Manager. This state does not necessarily mean that the descriptor is being used.
- drop An object is in the process of being dropped.
- **destroy** A descriptor is being released so it can be used for another object.
- **clean** The descriptor is between the active and free states of the chain; in this state the descriptor cannot be found. The next state after this is either free or create.

When the SQL Server Descriptor Manager tries to access a descriptor and it finds it in an incorrect state, Error 8203 occurs.

Error 8203 is caused by a SQL Server problem.

#### Action

Shut down and restart SQL Server to clear the cache corruption.

If the 8203 error persists, call Sybase Technical Support.

## Additional Information

Have the following information ready when you call Sybase Technical Support:

- SQL Server release and EBF rollup level
- SQL Server error log
- Text of all error messages

#### **Severity Level**

20

#### **Error Message Text**

```
Expected to receive system catalog id. Instead \mathrm{\%} was received.
```

#### Explanation

Session descriptors are the data structures used to manage access to SQL Server objects. SQL Server tracks individual descriptors by database table and system catalog ID. Error 8204 occurs if an invalid system catalog ID is returned while SQL Server is trying to locate a descriptor.

Error 8204 is caused by incorrect system catalogs or a SQL Server problem.

#### Action

Call Sybase Technical Support.

#### **Additional Information**

Have the following information ready when you call Sybase Technical Support:

- SQL Server release and EBF rollup level
- SQL Server error log
- Text of all error messages
- Output of dbcc checkcatalog if you know which database had the error

### **Severity Level**

17

## **Error Message Text**

```
Couldn't find an available descriptor for an object. Configure your system for more descriptors.
```

### Explanation

This message is displayed when SQL Server is configured for fewer open objects (also known as descriptors) than are currently needed.

The configuration parameter **open objects** sets the maximum number of objects that can be open at one time on SQL Server.

#### Action

1. Use the system procedure **sp\_configure** in order to display the run value of the configuration parameter you need to change:

```
1> sp_configure "open objects"
2> go
```

- 2. Setting the number of open objects higher does not have a significant impact on performance or storage requirements. However, you should be careful when changing SQL Server configuration parameters. Refer to "SQL Server Memory Issues" in the *SQL Server Troubleshooting Guide* to select an appropriate value for *open objects*.
- 3. Increase the number of open objects to a value larger than the current run value of the open objects variable:

```
1> sp_configure "open objects", new_value
2> go
1> reconfigure
2> go
```

4. Restart SQL Server so that the configuration change will take effect.

## **Additional Information**

For more information, see:

- open objects in the System Administration Guide
- "How SQL Server Uses Memory" in the SQL Server Troubleshooting Guide
- "Open Objects Parameter May Be Too Low" on page 2-517 of this manual

## **Severity Level**

26

## **Error Message Text**

```
Mismatch found between the name and id descriptor hash table. Descriptor hashed by name = 0x%0x and hashed by id = 0x%0x.
```

### Explanation

Session descriptors are the data structures used to manage access to SQL Server objects. ID and name hash tables are used to maintain session descriptors. Every session descriptor should have a corresponding entry in the ID and name hash tables. Error 8211 occurs when SQL Server finds two descriptors for one object or when a descriptor for an object exists in only one of the hash tables (ID or name table).

Error 8211 is caused by a SQL Server problem.

#### Action

Shut down and restart SQL Server to clear the cache corruption.

If the 8211 error persists, call Sybase Technical Support.

#### **Additional Information**

Have the following information ready when you call Sybase Technical Support:

- SQL Server release and EBF rollup level
- SQL Server error log
- Text of all error messages

## sysindexes Manager Errors

This section contains error messages pertaining to the *sysindexes* Manager.

#### **Severity Level**

21

#### Error Message Text

```
Index row for object %ld (index id %d) was not found in database %d.
```

#### Explanation

Each database has a *sysindexes* table. The *sysindexes* Manager maintains and provides access to the rows of the *sysindexes* table. It handles inserting and deleting of *sysindexes* rows and manages the abort and recovery of individual operations performed on the rows. It also provides read/write access to the database information structure.

sysindexes contains one row for each:

- Clustered index
- Nonclustered index
- Table that has no clustered index
- Table that contains *text* or *image* columns

Error 8402 occurs when one of these rows is expected but not found for the object specified in the error message.

#### Action

1. Determine the name of the database mentioned in the error message:

```
1> use master
2> go
1> select db_name(database_ID)
2> go
```

where *database\_ID* is the database ID in the error message.

2. Search the *sysindexes* table for the object displayed in the error message:

```
1> use database_name
2> go
```

```
1> select * from sysindexes
2> where id = object_ID
3> go
```

where *object\_ID* is the object ID in the error message.

3. If the object is found in *sysindexes*, continue with the step 4.

If the object is **not** found in *sysindexes*, call Sybase Technical Support.

4. If the object specified in the error message is **not** a system table (a system table's object ID is less than 100), continue with step 5.

If the object specified in the error message is a system table, refer to "How to Fix a Corrupted Index on System Tables" in the *SQL Server Troubleshooting Guide* for instructions on how to repair the system table index. Then go to step 7.

- 5. Drop the index for the object specified in the error message.
- 6. Re-create the index. This clears the corruption in most cases.
- 7. Run dbcc checktable on the table to verify that the corruption is gone. If corruption still exists, call Sybase Technical Support.

#### **Additional Information**

See the *System Administration Guide* and *SQL Server Reference Manual* for information about dropping and re-creating indexes.

### **Severity Level**

21

## **Error Message Text**

```
No object id specified in sysindexes Manager call to: %s.
```

### Explanation

Each database has a *sysindexes* table. The *sysindexes* Manager handles the inserting and deleting of *sysindexes* rows and manages the abort and recovery of individual operations performed on the rows. It also provides read/write access to the database information structure.

The *sysindexes* Manager sets up the session descriptor and object ID for the object whose index row is to be changed. If no session descriptor is given, then *sysindexes* is used and the object ID must be specified. In that case, if no object ID is specified, Error 8412 occurs. *%s* is a character string identifying the caller of the *sysindexes* Manager.

This error is caused by a SQL Server problem.

#### Action

Call Sybase Technical Support.

## **Additional Information**

Have the following information ready when you call Sybase Technical Support:

- SQL Server release and EBF rollup level
- SQL Server error log
- Text of all error messages

## Site Buffer Manager Errors

This section contains error messages pertaining to the Site Buffer Manager.

## **Severity Level**

26

## **Error Message Text**

```
Could not get the address lock. address=0x%1x, status=%d.
```

## Explanation

Address locks are exclusive locks that control memory addresses. They are used for:

- Object Allocation Map (OAM) pages
- Allocation pages
- Updating the disk maps in the database table memory structure
- Updating values for descriptors and site buffers

The Site Buffer Manager is one of the managers that control (SQL) Server-to-Server communications.

Error 8704 occurs when the Site Buffer Manager fails to get an address lock for an active site buffer and fails to attach a logical connection to it.

This error is caused by a SQL Server problem

## Action

Shut down and restart SQL Server. If the 8704 error persists, call Sybase Technical Support.

## **Additional Information**

Have the following information available before calling Sybase Technical Support:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all error messages
- sp\_configure, sp\_who, and dbcc lock output

## Threshold Errors (Continued)

This section contains additional error messages pertaining to thresholds.

### **Severity Level**

0

## **Error Message Text**

```
Maximum file descriptors or FILLM process quota
too low to support requested number of user
connections. Configuration variable 'user
connections' will not be modified.
```

### **Explanation**

This error occurs when you attempt to increase the value of the configuration parameter user connections using sp\_configure and the new value is too large for your current operating system parameters.

Although most of the open file descriptors are available for user connections, a few are used by SQL Server for opening files and devices. The number of user connections allowed by SQL Server is limited by the number of open file descriptors allowed by the operating system. The number allowed is platform- and versionspecific. For example:

• **SunOS 4.1.1** (Sun Solaris 1.x) has 256 file descriptors available. It has 1,024 available if DBE, the Database Excelerator, is installed.

**SunOS 4.1.3** has both "hard" and "soft" limits as settable parameters. The soft limit determines the number of open file descriptors available to SQL Server up to the maximum of 1,024 on a 4.1.3 operating system. A system running 4.1.3 has 256 file descriptors available, 2,048 if the Sun Database Excelerator (DBE) is installed.

• **SunOS 5.x** (Sun Solaris 2.x) sets both "hard" and "soft" limits: the soft limit can be increased by the user up to the hard limit, but the hard limit can be changed only by someone with "superuser" permissions. The soft limit determines the number of open file descriptors available to SQL Server, up to a maximum of 10,000 on a SunOS 5.x operating system. The SunOS 5.x operating system has a maximum of 3,072 file descriptors on a 4.9.1 SQL Server.

#### ► Note

To set the hard limit to a value greater than 1,024, you must be logged in as "root."

To verify the maximum number of user connections you are allowed to configure, run the following query in isql:

```
1> select @@max_connections
2> go
```

The value returned is the maximum number of user connections configurable, based on the number of available file descriptors. If the number of connections is not sufficient for your needs, alter the kernel on your operating system.

### Action for SunOS and Sun Solaris

1. Verify the configured descriptor soft and hard limits:

	C Shell	Bourne Shell						
Soft	<pre>% limit descriptors</pre>	\$ ulimit -n						
Hard	<pre>% limit -h descriptors</pre>	\$ ulimit -Hn						

2. To increase the soft limit, use one of the following commands:

C Shell	Bourne Shell
% limit descriptors $x$	\$ ulimit -Sn <i>X</i>

where *x* is the value to which you want to increase the soft limit.

These commands can be used in your runserver file to increase the soft limit every time SQL Server is started. Because the runserver file is a Bourne Shell script in the Sun Solaris operating system, use the Bourne Shell version of these commands in the runserver file.

3. Refer to the *System Administration Guide Supplement* for instructions for changing the hard limit.

### ► Note

For SunOS 4.1.3, since the number of user connections cannot exceed the number of file descriptors available, you cannot increase user connections to a higher value than the hard-coded number of file descriptors.

## Action for OpenVMS

Recalculate the necessary user connections. Refer to your *System Administration Guide Supplement* and recalculate your operating system process quotas.

## **Additional Information**

For information about user connection limits for your operating system, refer to the appropriate *System Administration Guide Supplement* and to your operating system manuals.

### **Severity Level**

0

### **Error Message Text**

```
The last-chance threshold for database %1! is now %2! pages.
```

### Explanation

A last-chance threshold has been established on a newly created log device (sp\_logdevice), or an existing one has been adjusted after a procedure which alters the size of a log device was executed (sp\_dropsegment, sp\_extendsegment). The threshold is established automatically by SQL Server.

### Action

No action is necessary.

### **Severity Level**

0

### **Error Message Text**

```
Could not update the last-chance threshold for database %1!
```

#### Explanation

Procedure lct\_admin is run automatically when any attempt is made to add or modify the log segment of a database with sp\_logdevice, sp\_dropsegment, or sp\_extendsegment. Error 17716 occurs when an attempt to run such a command has failed.

### Action

Check other messages displayed near the attempted command. They may give you additional information regarding the failure of lct\_admin or may indicate the original command was not successful. If the command appeared to run successfully, except for this error message, contact Sybase Technical Support about evaluating the current status of the last-chance threshold in this database.

When calling Technical Support, have the following information ready to fax or otherwise provide to Sybase Technical Support:

- Output of select @@version
- Output of all error messages
- Output of dbcc commands

### **Severity Level**

0

### **Error Message Text**

```
Table '%1!' does not exist in database '%2!' -- cannot add thresholds.
```

#### Explanation

This error occurs when the system table *systhresholds* does not exist in the current database, and **sp\_addthreshold** or **sp\_modifythreshold** was not able to add or modify the requested threshold.

#### Action

*systhresholds* should exist in every SQL Server 10.0 and later database. Run dbcc checkalloc and dbcc checkdb to determine if there is corruption in the database. Contact Sybase Technical Support if *systhresholds* is missing from a release 10.0 or later database.

When calling Technical Support, have the following information ready to fax or otherwise provide to Sybase Technical Support:

- Output of select @@version
- Output of all error messages
- Output of dbcc commands

#### **Additional Information**

Refer to the *SQL Server Reference Manual* entry for sp\_addthreshold for additional information.

### **Severity Level**

0

## **Error Message Text**

There is no segment named '%1!'.

### Explanation

This error occurs when an attempt is made to add, drop, or modify (using sp\_addthreshold, sp\_dropthreshold, or sp\_modifythreshold) a segment that does not exist in the current database.

### Action

Run sp\_helpsegment to display segment information about the current database. If the segment does exist, check that you have spelled it correctly in your procedure. If the segment does not exist and you believe it has been dropped, use sp\_addsegment to add it back.

If the requested segment does not exist and you are certain no one has dropped it, run dbcc commands to check the integrity of the database.

### **Additional Information**

Refer to the *SQL Server Reference Manual* entry for sp\_addthreshold for additional information.

#### **Severity Level**

0

### **Error Message Text**

This threshold is too close to one or more existing thresholds. Thresholds must be no closer than 128 pages to each other.

#### Explanation

This error occurs when an attempt is made to add or modify thresholds (sp\_addthreshold or sp\_modifythreshold) that are too close together.

### Action

Thresholds for a database cannot be closer than (*size\_of\_threshold* + (2 \* @@thresh\_hysterisis)). Ensure that the free space allocation allows enough pages. Rerun the command with the appropriate value for free space. See the *System Administration Guide* for more information about threshold size and hysteresis value.

## **Severity Level**

0

## Error Message Text

Adding threshold for segment '%1!' at '%2!' pages.

## Explanation

This message is displayed after the successful completion of sp\_addthreshold.

## Action

No action is required.

### **Severity Level**

0

### **Error Message Text**

```
A threshold at %1! pages is logically impossible
for segment '%2!'. Choose a value between %3! and
%4! pages.
```

### Explanation

This error occurs when an attempt is made to either add or modify a threshold (sp\_addthreshold or sp\_modifythreshold) for a number of pages that is greater than the segment named in the command.

### Action

Run sp\_helpdb *database\_name* to display the device fragments used by the specified database and the size of the device fragment. Then rerun the command with an appropriate page value for *free\_space*.

## **Severity Level**

0

## **Error Message Text**

```
Only the DBO of database '%1!' or a user with
System Administrator (SA) role may add, delete, or
modify thresholds in that database.
```

### Explanation

This error occurs when you try to run sp\_addthreshold, sp\_dropthreshold, or sp\_modifythreshold and you are not the owner of the current database and do not have "sa" role.

### Action

If you are not the database owner, run sp\_displaylogin *login* to display roles currently assigned to you (you must have "sso" or "sa" role to display this information about another user).

Ask your system administrator to give you "sa" role or ask another user to execute the command.

### **Severity Level**

0

### **Error Message Text**

```
Table '%1!' does not exist in database '%2!' -- cannot drop thresholds.
```

#### Explanation

This error occurs when the system table *systhresholds* does not exist in the current database, and **sp\_dropthreshold** is unable to drop the requested threshold.

### Action

*systhresholds* should exist in every SQL Server 10.0 and later database. Run dbcc checkalloc and dbcc checkdb to determine if there is corruption in the database. Contact Sybase Technical Support if *systhresholds* is missing from a release 10.0 or later database.

When calling Technical Support, have the following information ready to fax or otherwise provide to Sybase Technical Support:

- Output of select @@version
- Output of all error messages
- Output of dbcc commands

## **Severity Level**

0

## **Error Message Text**

```
Segment '%1!' does not have a threshold at '%2!' pages.
```

### Explanation

This error occurs when an attempt is made to either drop or modify a threshold (sp\_dropthreshold or sp\_modifythreshold) that does not exist for the free space page allocation given in the command.

### Action

Run sp\_helpthreshold without any options to display all the thresholds and their free page allocations for the current database. Then rerun the stored procedure with corrected parameters.

### **Severity Level**

0

### **Error Message Text**

You may not drop the log's last-chance threshold.

### Explanation

The last-chance threshold is created and maintained by the SQL Server and cannot be dropped by a user. Error 17905 occurs when an attempt is made to drop the database last-chance threshold with sp\_dropthreshold.

### Action

No action is required.

## **Severity Level**

0

## **Error Message Text**

```
Dropping threshold for segment '%1!' at '%2!' pages.
```

## Explanation

This message is issued after successful completion of the sp\_dropthreshold command.

### Action

No action is required.

#### **Severity Level**

0

### **Error Message Text**

```
Database '%1!' has no thresholds -- table '%2!' does not exist.
```

#### Explanation

This error occurs when the system table *systhresholds* does not exist in the current database and **sp\_helpthreshold** is unable to output information from that table.

### Action

*systhresholds* should exist in every SQL Server 10.0 and later database. Run dbcc checkalloc and dbcc checkdb to determine if there is corruption in the database. Contact Sybase Technical Support if *systhresholds* is missing from a release 10.0 or later database.

When calling Technical Support, have the following information ready to fax or otherwise provide to Sybase Technical Support:

- Output of select @@version
- Output of all error messages
- Output of dbcc commands

## **Severity Level**

0

## **Error Message Text**

This procedure can only affect thresholds in the current database. Say 'USE %1!', then run this procedure again.

### Explanation

Thresholds for a specified database may only be created or modified while in the database. Error 18031 occurs when you try to run sp\_addthreshold, sp\_dropthreshold, or sp\_modifythreshold when you are not in the database for which you are running the command.

### Action

- 1. Run select db\_name to determine the name of the current database.
- 2. Issue the use command to select the correct database.
- 3. Rerun the sp\_addthreshold, sp\_dropthreshold, or sp\_modifythreshold command.

## **Severity Level**

0

#### **Error Message Text**

```
You may not alter the free space or segment name of the log's last-chance threshold.
```

### Explanation

The only parameter you can change for the last-chance threshold is the name of the procedure that will execute when the threshold is crossed. Error 18032 occurs when you try to modify the last-chance threshold name or free space allocation with sp\_modifythreshold.

### Action

No action is required.

## **Severity Level**

0

## **Error Message Text**

```
You cannot drop user because user '%1!' owns thresholds in database.
```

### Explanation

The user who created a threshold or last modified a threshold is the threshold owner. A user who owns a threshold cannot be dropped. Error 18033 occurs when you try to drop a user who owns a threshold.

### Action

You have three options:

- Drop the threshold procedure owned by the user. Then you can drop the user.
- Have another user with appropriate privileges modify the threshold. Then you can drop the user.
- If you want to inactivate the user SQL Server login while leaving the threshold active, use sp\_locklogin to prevent use of the login.

## **Unnumbered Errors**

This section includes error messages that have a text description but no number. These errors are displayed only in the SQL Server error log. They are presented in alphabetical order.

# **Buffer Mismatch Error**

### **Error Message Text**

server: Buffer "%d" from database "%s" has page
number "%d" in the page header and page number
"%d" in the buffer header.

### Explanation

This error only appears in the error log, and it means that a cache management problem occurred. This can be a very serious error because it is often followed by database corruption, such as 605 errors.

Although often a result of hardware failure, this error can also be caused by operating system or SQL Server problems.

### Action

If possible, shut down and restart SQL Server immediately after this error occurs. This may prevent the buffer cache error from being flushed to disk.

Run complete diagnostics on the machine running SQL Server as well as on all disk drives and controllers attached to the SQL Server machine. Run complete dbcc checks on any databases involved, including dbcc checkalloc and dbcc checkdb. Repair or replace faulty hardware. Then, shut down SQL Server and restart it.

If you do not find any hardware problems, call Sybase Technical Support immediately.

Before calling Technical Support for assistance, have the following available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Hardware error log
- Text of all the error messages
- Reproducible case (if possible)

## **Character Set Error**

## **Error Message Text**

```
SQL Server doesn't support use of character_set_name character set.
```

## Explanation

This error occurs when the client software default character set has not been installed on the SQL Server to which the client is trying to connect. This happens most often when the client is running on a different operating system from SQL Server.

### Action

To allow the client and SQL Server to communicate, do the following:

1. Check to see if the client software character set is installed on SQL Server by executing this query:

```
1> use master
2> go
1> select name, id, csid from syscharsets
2> go
```

If the client software's default character set appears in the output, have your primary site contact call Sybase Technical Support. If it does not appear, go to step 2.

If the client software's default character set does not appear in the output, you may be able to set the client software to the character set which SQL Server is using. For example, using the isql -J flag (see the *SQL Server Reference Manual* entry for isql for more information). If this is not possible, go to step 2.

## ♦ WARNING!

Familiarize yourself with all the instructions in the *Installation Guide* before completing step 2, in order to make the character set addition a smooth process.

2. Run the installation utility sybinit (or the equivalent for your platform). Choose the additional character set item from the character set menu. If the particular character set you need is not one of the options listed, you must purchase that character set by

purchasing a Language Module from Sybase, and install it with sybinit or its equivalent.

3. Try again to connect to SQL Server with the client software. Call Sybase Technical Support if you are still unable to connect to SQL Server.

# **Could Not Create Shared Memory Error**

## **Error Message Text**

kistartup: could not create shared memory

## Explanation

kistartup is the SQL Server kernel initialization routine. It performs the following tasks:

- Reads the configuration parameters
- Allocates kernel tables and buffer pools
- Sizes memory
- Initializes devices
- Spawns the service processes
- Starts the scheduler

When **kistartup** is unable to create shared memory for kernel initialization, the error "kistartup: could not create shared memory" is written to the SQL Server error log.

kistartup errors occur after os\_create\_region errors.

### Action

Refer to "os\_create\_region Errors" on page 2-520 for information about recovering from the os\_create\_region errors which precede the kistartup error. Once those errors are corrected, the kistartup error will not occur.

# Current Process Infected with %d

## **Error Message Text**

Kernel: current process (0x%x) infected with %d

## Explanation

SQL Server reports this error when it detects a UNIX signal specifying an error. The signal values ("%d") displayed in the above message vary by platforms and SQL Server release versions; the most common are either 10 or 11.

### Current process infected with 10

A value of 10 [SIGBUS] means that the operating system detected an address alignment error or a miscellaneous hardware error (for example, bus timeout).

A timeout can occur when the CPU issues a request across the bus for the contents of a memory location, and that request is not answered within that CPU's timeout period (usually a few nanoseconds).

### Current process infected with 11

A value of 11 [SIGSEGV] means that the operating system detected a segment violation error.

Sometimes this error occurs in conjunction with stack overflow or data corruption. For more information on stack overflow, see "Stack Guardword Error" on page 2-526.

The message appears in the SQL Server error log followed by a stack trace. The *SQL causing error* or the *lasterror* displayed in the SQL Server error log may be the underlying cause for this error. But they can also be just the last data SQL Server had in its cache space.

In order to identify the *lasterror* (except in the cases where the *lasterror* is 0), get the number displayed by SQL Server in the *lasterror* field from the SQL Server error log and consult this manual for more information on this error number.

### In the following example, the value for *lasterror* is 614.

00: 94/02/14 11:32:26.02 kernel: current process (0x1fb001d) infected with 11  $\,$ 

#### Action

In order to correct this error, try to eliminate the *lasterror* first, as this might be one of the causes for this error (except when *lasterror* is 0). Also, try rerunning the command referenced in the *SQL causing error* to see if the problem reoccurs.

Check your hardware error log as this error can be caused by hardware failure as well.

#### **Additional Information**

If this error occurs frequently, contact Sybase Technical Support. It would be useful to have a reproduction case (for example, a query which always causes the "infected" message) before calling Technical Support.

Before calling Technical Support for assistance, have the following available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Operating system error log
- Text of all the error messages
- Reproducible case (if possible)

# Dstart I/O Request Repeatedly Delayed

## **Error Message Text**

## Explanation

This error message is not dangerous and there is no data corruption connected with it unless it is reported together with other kernel messages in the error log which indicate corruption.

## **Error Message**

If this is an error message, it is a sign of inconsistencies or corruption relating to the asynchronous I/O capabilities of SQL Server. In this case, the above error message will be reported along with other error messages in the SQL Server error log.

If this message is accompanied by other error or kernel messages in the SQL Server error log, refer to the appropriate section of this manual (by error number or name) for corrective measures regarding those other messages.

## Warning Message (Informational)

This can be an informational message (occurring by itself, any number of times). If this is a warning message, it is displayed by itself in the SQL Server error log. In this case, the message is displayed in the SQL Server error log when:

- SQL Server is configured for a different rate of concurrent asynchronous I/O requests than the operating system is configured for. Every time an I/O request has to be requeued in SQL Server's I/O queues, there is a decrease in overall performance due to requeueing overhead.
- SQL Server is configured for a lower throughput rate of concurrent asynchronous I/O than it is capable of (that is, the value of the configuration parameter cnmaxaio\_server is less than cnblkmax or cnmaxaio\_engine is less than cnblkmax; these configuration parameters are described below in the "Action" section).

Asynchronous disk I/O allows SQL Server to initiate a disk operation to continue processing other data while the disk operation is completed. This can improve performance and recoverability

especially when the master device, the user databases, and transaction logs are on raw disk partitions. For more information on raw disk partitions, see "Device Administration Issues" in the *SQL Server Troubleshooting Guide*.

#### Action

The following SQL Server configuration parameters and operating system parameters play a role when this error occurs:

- cnblkio the initial number of block I/O structures at SQL Server boot time (this is a SQL Server configuration parameter)
- cnblkmax the maximum number of block I/O structures available to SQL Server (this is a SQL Server configuration parameter)
- cnmaxaio\_server the number of asynchronous I/O requests outstanding at one time for SQL Server (this SQL Server configuration parameter applies only to multiprocessor machines)
- cnmaxaio\_engine the number of asynchronous I/O requests outstanding at one time for one engine (in the case of multiprocessor machines) and the maximum number of asynchronous I/O requests outstanding at one time for SQL Server (in the case of single processor machines) (this is a SQL Server configuration parameter)
- asynchronous I/O operating system parameters

### ♦ WARNING!

The buildmaster commands below must be run while SQL Server is down.

Review the current configuration settings of SQL Server by running the following command:

```
% buildmaster -yall -dmaster_device | grep cn
```

where *master\_device* includes the path and name of the master device. If no rows are returned, make sure you entered the correct master device path. If still no rows are returned, run the command again without the " | grep cn" part.

The above SQL Server variable values can be adjusted by running the following command while SQL Server is shut down:

```
% buildmaster -yvariable_name=new_value -dmaster_device
```

For Sun 4.1

Check the UNIX kernel */usr/share/sys/sys/asynch.h* file and record the value of the kernel parameter *MAXASYNCHIO* (maximum rate of concurrent asynchronous I/O requests, usually equal to 200 or more).

Check the values of the Sybase configuration parameters cnblkio, cnblkmax, and cnmaxaio\_server and adjust them while SQL Server is shut down in order to match the value of *MAXASYNCHIO*. Use the buildmaster utility to do this:

```
% buildmaster -ycnblkio=value_of_MAXASYNCHIO -dmaster_device
```

```
% buildmaster -ycnblkmax=value_of_MAXASYNCHIO -dmaster_device
```

```
% buildmaster -ycnmaxaio_server=value_of_MAXASYNCHIO
-dmaster_device
```

where *master\_device* includes the path and name of the master device. If you are running on SunOS 4.1.3 (Sun Solaris 1.x), the UNIX kernel parameter *perproc\_maxasyncio* (per-process limit on concurrent asynchronous I/O, found in */usr/share/sys/conf.common/param.c*) defaults to 50. Increase this parameter to 200, to match the value of the Sybase configuration parameter cnmaxaio\_server. (Rebuild the UNIX kernel after making the change to the UNIX kernel parameter.)

The above recommendations are made to help you clear the error message. For asynchronous I/O tuning, contact your hardware vendor to find out what are the operating system parameters that improve asynchronous I/O performance on your platform.

For SunOS 5.x (Sun Solaris 2.x)

Check whether there is a hard coded value for *MAXASYNCHIO* (usually set to 200 and stored in */usr/include/sys/asynch.h* or in */etc/system* on newer operating system versions such as Solaris 2.4). If there is a hard coded value for *MAXASYNCHIO*, set the values for SQL Server's asynchronous I/O configuration parameters (cnblkio, cnblkmax, and cnmaxaio\_server) to the same value *MAXASYNCHIO* is set to. If you have the KAIO patch 102020-03 for Solaris 2.4, you can raise the values of cnblkio, cnblkmax, and cnmaxaio\_server to as high as 500.

## For AIX

Configure the operating system for a higher number of outstanding asynchronous I/Os using smit. Bring up the smit aio screen by typing smit aio at the AIX prompt. Then, go to the "Change/Show

Characteristics of Asynchronous I/O" screen and raise the "maximum number of REQUESTS" parameter to 4096.

This change will improve SQL Server performance. You will **not** need to reboot the operating system for the change to take effect. For more information on asynchronous I/O, search for "asynchronous" using the **info** utility.

Also, raise the values of the SQL Server configuration parameters cnblkio, cnblkmax, and cnmaxaio\_engine to 4096 to match the "maximum number of REQUESTS" value in smit.

### For DEC OSF/1

The number of outstanding asynchronous I/O's is hardcoded in the operating system kernel in the *AIO\_MAX* parameter (stored in a header file in /sys). Its default value is 64. The SQL Server configuration parameters cnblkmax, cnblkio, and cnmaxaio\_engine should be set to 64 so that SQL Server can match the operating system's rate of concurrent I/O requests.

#### For HP-UX

Depending on the operating system kernel, the maximum number of outstanding asynchronous I/O requests can be either 50 (on older HP-UX 8.x kernels) or 500 (on patched HP-UX 8.x kernels and default on HP-UX on 9.x kernels). this value is stored in the *MAX\_CONCURRENT* parameter in */usr/include/sio/asynch*. Adjust SQL Server's configuration parameters cnblkmax, cnblkio, and cnmaxaio\_server to match *MAX\_CONCURRENT*.

The error log example below shows the SQL Server startup messages associated with the older 8.x asynchronous I/O driver and its limit of 50 asynchronous requests at one time. Once a later asynchronous patch has been applied or the new operating system version has been installed, this limit increases to 500.

```
00: 92/06/04 14:38:07.10 server: Number of blocks left for proc headers: 351.
```

00: 92/06/04 14:38:07.36 kernel: basis\_dioinit: Using older version of asynch I/O driver with **50** concurrent requests

00: 92/06/04 14:38:07.38 server: Opening Master database ...

For other operating systems, check your operating system manuals for information about asynchronous configuration issues.

If problems persist, refer to the error documentation in this manual or contact Sybase Technical Support. Have the following ready to fax or otherwise provide to Technical Support:

- Output from buildmaster -yall -dmaster\_device
- SQL Server release and EBF Rollup level
- SQL Server error log
- Operating system error log
- Text of all the error messages

## **Additional Information**

Refer to the "raw partition" entry in *SYBASE Installation and Operations* and "Device Administration Issues" in the *SQL Server Troubleshooting Guide* and to your operating system documentation on asynchronous I/O.

# Failed to Open Device Error

## **Error Message Text**

kdopen: failed to open device %s for vdn %d

## Explanation

Before using a database device, SQL Server attempts to perform some initialization tasks. When mirroring is being used, if SQL Server is unable to open the secondary side of a mirrored device for these initialization tasks, the message "kdopen: failed to open device %s for vdn %d" is written to the error log. %s is the physical path to the secondary side of the mirrored device and %d is the virtual disk number of the device.

This error can be raised for the following reasons:

- A database device is offline
- A database device has been removed or renamed
- A database file or device is inaccessible to SQL Server for any other reason such as insufficient permissions

### Action

- 1. Examine the SQL Server error log for other messages that may explain why this error has occurred. Refer to the instructions in this manual for resolving other errors associated with this error.
- 2. Identify the device involved in the error:

```
1> select name from master..sysdevices
2> where low/power(2,24) = vdevno
```

where *vdevno* is the virtual disk number of the device as shown in the error message.

3. Examine the availability and condition of the device involved using your standard operating system procedures and make sure the device is accessible.

# File Already in Use Error

## **Error Message Text**

```
basis_dlock: file '%s' already in use by a SQL
Server
```

## Explanation

During startup, SQL Server tries to lock the master device. If SQL Server finds that the master device is already locked, the message "basis\_dlock: file '%s' already in use by a SQL Server" is written to the error log and SQL Server does not start. %s is the name of the master device file.

The most common reasons for this error are:

- You are trying to start a SQL Server with a master device that is already being used by another SQL Server.
- The SQL Server you are trying to start is already running.

## Action

At the operating system level, check the processes that currently exist and determine whether a SQL Server using the master device you specified is already running.

If a SQL Server using the master device you specified is already running, and this is not the Server you want to start, determine the correct master device name for the Server you want to start.

If you cannot determine whether the master device is already in use, reboot the operating system.

## ► Note

If the master device resides on an operating system file system, you can copy the file to another name, modify your runserver file, and start SQL Server with that new file.

## Memory Too Fragmented Error

## **Error Message Text**

```
Physical memory on this machine may be too fragmented
```

### Explanation

At startup, SQL Server allocates and initializes the memory needed for its structures and cache. This error occurs when SQL Server is not able to initialize memory needed for its structures and static cache.

The most common cause of this error is that the total memory required for the configurable variables is more than the memory configured for SQL Server.

### Action

1. At the operating system level, check memory use.

On UNIX machines, there are several commands for checking memory use. Following are examples of two of these commands:

#### % /etc/pstat -s

```
44076k allocated + 8920k reserved = 52996k used, 40960k available
```

#### % vmstat

procs	memory			page				disk					faults			cpu			
r b w	avm	fre	re	at	pi	ро	fr	de	sr	sO	d1	d2	d3	in	sy	CS	us	sy	id
4 0 0	0	336	0	1	0	0	0	16	0	1	0	0	0	181	260	399	84	2	14

If processes other than SQL Server are leaving memory fragmented, rebooting the operating system may resolve this error.

2. Reset your SQL Server configuration parameters to their default values so you can start SQL Server. Refer to "How to Reset SQL Server to Its Default Configuration" in the *SQL Server Troubleshooting Guide* for instructions.

Start your SQL Server.

3. If you still cannot start SQL Server, determine the differences between a SQL Server default configuration and your current

configuration so you can modify your configuration and start SQL Server. To do this:

- 1. Create a new physical master device. Refer to "Master Device Completely Lost" in the *SQL Server Troubleshooting Guide* for instructions.
- 2. With the SQL Servers still down, get a listing of the current configuration for both Servers and compare the output:
- % buildmaster -yall -dmaster\_device\_name
- 3. Change the configuration values for your original SQL Server to the values of the default SQL Server (while SQL Server is still down):

### % buildmaster -yparameter\_name=new\_value -dmaster\_device\_name

4. Start your SQL Server.

4. Once your SQL Server is running, use sp\_configure either to reduce the value of configurable variables (such as user connections and locks) or to increase the value of the memory variable. The memory variable controls the size of memory, in 2K units, that SQL Server allocates from the operating system.

Refer to "How SQL Server Uses Memory" in the *SQL Server Troubleshooting Guide* for information about memory-related SQL Server configuration parameters.

## **Additional Information**

Before calling Technical Support for assistance, have the following available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Text of all the error messages
- sp\_configure or buildmaster -yall output

# Network Information Message

#### Message Text

kernel: Using '%s' for network information.

#### Explanation

SQL Server uses information from the network addressing (interfaces) file located in the *\$SYBASE* directory when it starts up and when it tries to establish a connection to a remote server. At those times, SQL Server displays this message, including the complete path of the interfaces file which was used.

Following are examples of this message for UNIX and OpenVMS:

#### UNIX

```
kernel: Using '/usr/u/sybase/interfaces' for
network information.
```

#### **OpenVMS**

```
kernel: Using '/SYBASE_SYSTEM:[SYBASE]interfaces'
for network information.
```

If a remote procedure call (RPC) is executed and no connection is currently established for that remote Server, SQL Server must first establish a connection to it, using a site handler process, before the RPC can be executed. Each time SQL Server establishes a site handler process, this informational message is written to the Server error log.

If the site handler process times out, then this message can occur many times for each remote Server. If the site handler for a remote Server does not time out, then SQL Server will write this message for the first RPC to that remote Server, and the message will occur only once for each remote Server.

Use sp\_helpserver to check the status of the remote SQL Server. If the *status* column of the sp\_helpserver output is "timeouts", then you can use the following action to keep the network information message to one occurrence.

#### Action

To limit this message to one occurrence for a remote Server, disable the "timeouts" option to that remote SQL Server. Log into isql as "sa" and use sp\_serveroption to set the "timeouts" option to "no timeouts":

```
1> sp_serveroption remote_sql_server_name,
2> timeouts, false
3> go
```

This will keep the connection to the remote Server established. It will be maintained until either the local or remote Server is shut down. If you have few available user connections, remember that one connection per remote SQL Server connection will be in constant use.

Refer to the *SQL Server Reference Manual* for more information about sp\_serveroption.

# No More Alarms Available Error

#### **Error Message Text**

uasetalarm: no more alarms available

#### Explanation

The Transact-SQL command waitfor defines a specific time, time interval, or an event for the execution of a statement block, stored procedure, or transaction. SQL Server uses alarms to carry out waitfor commands correctly.

The number of alarms needed by SQL Server is related to the requirements of an application and the number of instances of that application being run simultaneously. The maximum number of alarms available for use by SQL Server is controlled by the cnalarm configuration parameter.

When SQL Server needs more than the number of alarms available, the message "uasetalarm: no more alarms available" is written to the error log.

#### Action

- 1. Shut down SQL Server. (SQL Server must be shut down before any buildmaster commands are run.)
- 2. Determine the current value of cnalarm (this example shows syntax for UNIX; refer to the examples below for syntax for other operating systems):

% buildmaster -yall -dmaster\_device | grep cnalarm

where *master\_device* is the name (including path) of the master device for this SQL Server.

- 3. Change the value of cnalarm to a value larger than the current value:
  - UNIX:

% buildmaster -ycnalarm=new\_value -dmaster\_device

- VMS:

```
$ buildmaster /alter="cnalarm=new_value" -
   /disk="master_device"
```

- OS/2, Microsoft Windows NT:

```
bldmastr -ycnalarm=new_value -dmaster_device
```

- Novell NetWare:

```
LOAD BLDMASTR -ycnalarm=new_value -dmaster_device
```

- Stratus VOS:

buildmaster.pm -ycnalarm=new\_value -dmaster\_device

4. Restart SQL Server.

#### ► Note

Each alarm structure uses 20 bytes of memory. If you raise the value of *cnalarm* significantly, readjust SQL Server's memory use accordingly.

# **Open Objects Parameter May Be Too Low**

#### Error Message Text

server: Warning: OPEN OBJECTS parameter may be too low; attempt was made to free up descriptor in desfree().

Run sp\_configure to increase parameter value.

or

server: Warning: OPEN OBJECTS or OPEN DATABASES
parameter may be too low; attempt was made to free
up descriptors in release\_dbtable().

Run sp\_configure to increase parameter value.

#### Explanation

This message is a warning indicating that you do not have enough object descriptors or open databases available in the chain containing all the free descriptors.

This message is displayed in the SQL Server error log when SQL Server reaches the end of its array of allocated object descriptors, tries to recycle previously-used object descriptors and fails. If can also be displayed when SQL Server has fewer open databases configured than the number of databases on the Server.

#### Action

To recover from this error, increase the values of the configuration parameters open objects and open databases.

The configuration parameter open databases sets the maximum number of databases that can exist at any given time on SQL Server.

The configuration parameter open objects sets the maximum number of objects that can be open at one time on SQL Server.

Setting the number of open databases or open objects higher does not have a significant impact on performance or storage requirements but does require some memory. For more information on how SQL Server allocates memory, see "SQL Server Memory Issues" in the *SQL Server Troubleshooting Guide*.

Use the system procedure sp\_configure in order to display the run values of the configuration parameters you need to change:

```
1> sp_configure "open databases"
2> go
```

If the value for open databases is less than the value displayed by select count (\*) from sysdatabases, increase open databases. Otherwise, increase open objects only.

# os\_attach\_region Error

#### **Error Message Text**

os\_attach\_region: shmat(%d): %s

#### Explanation

SQL Server uses the function os\_attach\_region to attach to a shared memory region based on a shared memory identifier. This error occurs when SQL Server fails to attach to a region. In this message, *%d* is the shared memory identifier and *%s* is an operating system error message.

The message "os\_attach\_region: shmat(%d): %s" is raised on UNIX systems only. Other operating systems raise slightly different errors.

#### Action

An os\_attach\_region error can occur when the value of the operating system parameter *shmseg* is too small. Refer to the *System Administration Guide Supplement* for your platform for information about configuring shared memory properly.

This message can also be raised by SQL Monitor Server. Refer to the *SQL Monitor Server Supplement* for your platform for details.

#### **Additional Information**

Refer to the operating system man pages for the shmat() system call.

# os\_create\_region Errors

#### **Error Message Text**

```
os_create_region: shmget (0x%x): %s
os_create_region: Shared memory segment %d is in
the way
os_create_region: uninitialized shared memory
descriptor
os_create_region: shmat (%d): %s
os_create_region: can't allocate %d bytes
```

#### Explanation

SQL Server uses the following functions to manage shared memory:

- os\_get\_shmid create a shared memory identifier
- os\_create\_region create a region based on a shared memory identifier
- os\_attach\_region attach to a region based on a shared memory identifier
- os\_detach\_region detach from (and delete) the shared region
- os\_format\_shmid format a shared memory identifier for printing

When os\_create\_region errors occur, SQL Server will not start.

The messages listed in this writeup are raised on UNIX systems only. Other operating systems raise slightly different errors.

os\_create\_region: shmget (0x%x): %s

This message is written to the error log when SQL Server fails to get a shared memory segment. In this message, *%x* is a shared memory key based on the shared memory identifier and *%s* is an operating system error message.

#### os\_create\_region: Shared memory segment %d is in the way

This error follows the shmget message and is also written to the SQL Server error log. A value of -1 for *%d* means the region does not exist.

os\_create\_region: uninitialized shared memory descriptor

During creation of a shared memory region, SQL Server attempts to validate the descriptor for the memory region. This message is written to the error log if the descriptor is found to be invalid.

os\_create\_region: shmat (%d): %s

This message is written to the error log when SQL Server fails to attach at an address. In this message, *%d* is the shared memory identifier and *%s* is an operating system error message.

os\_create\_region: can't allocate %d bytes

SQL Server was unable to allocate the number of bytes it requested for the shared memory region.

Action

1. At the operating system level, check which shared memory processes are using and whether shared memory segments are being used by SQL Server.

To check this on UNIX, run this command as the "sybase" user:

% ipcs -m

IPC status from workstationl as of Fri May 26 14:08:25 1995 T ID KEY MODE OWNER GROUP Shared Memory: m 256 0x699b7e24 --rw----- sybase sybase m 257 0x699b7e25 --rw----- sybase sybase

> If shared memory segments are being used by SQL Server, reboot the operating system to clear shared memory or remove them using the ipcrm operating system command.

#### ► Note

Before removing the shared memory segments, identify the process that created them using the command "ipcs -ma" to make sure you only remove the appropriate segments.

- 2. Check the *\$SYBASE* directory to see whether there are any *\*.krg* or *\*.srg* files left from an abnormal SQL Server exit. If any such files exist, delete them.
- 3. os\_create\_region errors can occur when shared memory is not configured properly on your operating system. Refer to the

*System Administration Guide Supplement* for your platform for information about configuring shared memory properly.

Additional Information

Refer to the operating system man pages for the shget() and shmat() system calls.

Refer to the operating system man pages for ipcs and ipcrm.

# Read/Write Error

#### **Error Message Text**

```
kernel: write error on virtual disk %d, block %ld
or:
kernel: read error on virtual disk %d, block %ld
```

#### Explanation

This error occurs when SQL Server failed to read or write to the specified location on disk. This failure is usually a result of a physical disk problem.

#### Action

1. Identify the device with the problem by selecting the row from *sysdevices* which has the virtual disk number indicated in the error message:

```
1> select name, phyname from master..sysdevices
2> where low/power(2,24) = virtual_disk_number
3> go
```

The output from this query should provide the physical name of the damaged disk. Shut down SQL Server and examine the disk as soon as feasible and correct any problems found to prevent further corruption. Then restart SQL Server.

► Note

The sp\_diskblock procedure described below also identifies the device.

2. Use the procedure described below to determine which database (if there is more than one database on the device), table, or index is affected by this error. Use this information to assess the potential severity of the problem quickly and decide on an action which is appropriate for your operating environment.

Use the sp\_diskblock procedure supplied in "How to Gather Information About Read/Write Errors" in the *SQL Server Troubleshooting Guide*. The syntax of this procedure is:

sp\_diskblock virtual\_disk, block\_number

For example, if the read/write error message displays a virtual disk number 4 and a block number 871, the procedure command is:

```
1> sp_diskblock 4, 871
2> go
Virtual disk 4, block 871 corresponds to:
Logical page 1895 in the "production" database
(dbid=4) on device "main".
```

Use the dbcc page command to determine which object is using that page. Refer to "How to Find an Object Name from a Page Number" in the *SQL Server Troubleshooting Guide* for details on this procedure.

# Server Is Unnamed

#### Message Text

server: server is unnamed

#### **Explanation**

This informational message is displayed during SQL Server start-up if the SQL Server has not been given a name via the sp\_addserver system procedure. The SQL Server name referred to here is stored in the *sysservers* system table as well as in the *@@servername* global variable. This name is used primarily for managing Server-to-Server communication and does not have to match the Server name used either in the interfaces file or in the runserver file.

#### Action

No action is required for this informational message. However, the SQL Server administrator can prevent this message by assigning a name to the SQL Server via the sp\_addserver procedure. (See the SQL Server Reference Manual for information about restrictions on valid SQL Server names with sp\_addserver.)

To eliminate the message, assign a name to SQL Server using the sp\_addserver command:

```
1> sp_addserver sql_server_name, local
2> go
```

The *@@servername* global variable will not reflect the name until SQL Server is restarted. All subsequent starts of the SQL Server display the new name during SQL Server start-up. For example, for a SQL Server named "MY\_SERVER", the message would be:

server: server name is 'MY\_SERVER'

Although this SQL Server name is associated with Server-to-Server communication, specifying a name does not in itself allow this communication and therefore does not represent a security risk. Naming all SQL Servers in this way makes the SQL Server error logs easy to identify, because the SQL Server name is contained within the log.

# Stack Guardword Error

#### **Error Message Text**

kernel: \*\*\* Stack Guardword corrupted

#### Explanation

This error occurs when the stack area used by a SQL Server process was corrupted. SQL Server usually shuts itself down after this error to avoid problems that could arise from this stack corruption.

#### ♦ WARNING!

# If this error occurs and SQL Server remains up, shut it down immediately!

SQL Server allocates one stack area for every configured user connection when it starts. These stacks are located in a contiguous area of memory. Stack corruption is usually a result of one process's stack overflowing its boundaries and writing on top of another process's stack. In general, a SQL Server process writes past the end of its stack area for one of two reasons:

- The stack size is too small. The information placed on the stack is valid but its total size exceeds the specified stack size. This is often caused by a single, complex query which requires more than the average amount of stack space.
- An incorrectly handled error results in recursive calls which eventually fill the stack, regardless of its size.

SQL Server detects that a stack has been corrupted by checking a known pattern, or "guardword," at the end of each stack area. SQL Server periodically checks whether this pattern has changed and whether the stack pointer is outside of its stack. If either of these conditions is true, this error displays and SQL Server shuts down.

#### Action for 4.9.1 SQL and Later

If this error occurs because a complex query has run out of stack area, correct the error using one of two the two following methods.

Solution One: Break Up the Complex Query

Break up the query into smaller pieces, if possible.

Solution Two: Increase Stack Size

Use the following information to select an appropriate size for the stack size configuration parameter:

- The stack size should always be an even increment of 2K (2,048 bytes). If a stack size is specified that is not divisible by 2K, SQL Server rounds it up to the next 2K increment.
- Increase the stack size by no more than 2K at a time. If the stack size is too large, you may be unable to start SQL Server.
- Increasing the stack size results in SQL Server requiring more memory for each configured user connection and device. Therefore, increasing the stack size without increasing the *memory* configuration parameter will result in less total cache space for SQL Server and could affect performance. See "SQL Server Memory Issues" in the *SQL Server Troubleshooting Guide* for more information.

Use sp\_configure to increase the stack size configuration parameter as follows:

```
1> sp_configure "stack size", new_value
2> go
1> reconfigure
2> go
```

Once this change is made, shut down and restart SQL Server so the change will take effect.

#### Action for Pre-4.9.1 SQL Server

If this error occurs because a complex query has run out of stack area, correct the error using one of the two following methods.

Solution One: Break Up the Complex Query

Break up the query into smaller pieces, if possible.

Solution Two: Increase Stack Size

Contact Sybase Technical Support for information on how to increase the stack size on SQL Server.

# t\_rcvconnect Error

#### **Error Message Text**

nconnect: t\_rcvconnect, %s

#### Explanation

This error occurs when a SQL Server using a TLI-based interface tries to establish a client connection and an operating system error message is returned.

When this error occurs, the connection is not established and the message "nconnect: t\_rcvconnect, %s" is written to the error log. *%s* is the text of the operating system error that caused the failure.

This is not a serious error unless it occurs frequently.

#### Action

Resolve the problem that caused the operating system error contained in the SQL Server error message. You may need assistance from an operating system or network administrator. Refer to the operating system man pages for information about the t\_rcvconnect() system call.

Check the SQL Server error log. If other errors exist, refer to the writeups in this manual for information about recovering from them.

If, after resolving the operating system problem and the problems found in the error log, you still get the message "nconnect: t\_rcvconnect, %s," call Sybase Technical Support.

#### **Additional Information**

Before calling Technical Support, have the following information available:

- SQL Server release and EBF Rollup level
- SQL Server error log
- Operating system error log
- sp\_helpserver and sp\_configure output
- Contents of interfaces file

# Glossary

#### allocation pages

Database-wide pages SQL Server uses to maintain the information about how each unit of 256 contiguous database pages is used.

#### clustered index

An index in which the physical order of the data and the logical (indexed) order of the data is the same. Only one clustered index can exist on a table. See also "leaf level."

#### error log

The SQL Server error log file includes:

- SQL Server restart messages (including database recovery messages)
- Fatal error messages
- Error messages from the kernel

The name of this file varies; see the *System Administration Guide Supplement* for your platform or the *SYBASE Utility Commands* manual for details.

The Backup Server has its own error log file where Backup Server error messages are written. Refer to the *System Administration Guide* for details.

The SQL Monitor Server has its own error log file where SQL Monitor Server error messages are written. Refer to the *SQL Monitor Server Supplement* for your platform for details.

The Replication Server has its own error log file where Replication Server error messages are written. Refer to the *Replication Server Administration Guide* for details.

The Navigation Server has its own error log file where Navigation Server error messages are written. Refer to the *Navigation Server Reference Manual* for details.

#### extent

Every 256-page allocation unit is divided into 32 structures called extents, each containing 8 contiguous pages. Each extent contains information about those 8 contiguous pages, including a bitmap showing pages in use and a bitmap showing pages to be deallocated.

#### foreign key

In a table, a column whose data values correspond to the values of a key column in another table.

#### index ID

The number SQL Server uses to uniquely identify (within a database) an index. The following table shows the index types:

Index ID	Meaning
0	Actual table data
1	Clustered index
2-250	Nonclustered indexes
255	Text or image page

#### leaf level

The bottom of a clustered or nonclustered index. In a clustered index, the leaf level contains the actual data pages of the table.

#### logical page

The unique address for pages in the context of each database.

#### nonclustered index

An index that stores key values and points to data. The leaf level points to data rows rather than containing the data itself.

#### **Object Allocation Map (OAM)**

An Object Allocation Map (OAM) contains information about how space is allocated for SQL Server objects. A single OAM page holds information about up to 250 allocation pages (248 if the object is an IDENTITY column).

Each allocation page for an object has an entry in the OAM pages for that table or index. The OAM entry for each allocation page stores the number of used and unused data and index pages on that allocation page.

There is a maximum of 263 OAM pages allowed, supporting a maximum of 65,535 allocation units for an object.

#### object ID

The number SQL Server uses to uniquely identify (within a database) a SQL Server object. If you know an object ID and you know the database the object is in, you can determine the object name. System tables have an object ID that is less than 100 and user tables have an object ID that is 100 or greater.

#### primary key

The column or combination of columns that uniquely identify a row.

#### remote procedure call (RPC)

A procedure executed on a different SQL Server from the SQL Server the user is logged into.

#### session descriptors

The data structures used to manage access to SQL Server objects. In SQL Server 10.0.1 and earlier versions, session descriptors exist in only two forms: **user** session descriptors (including worktable session descriptors) and **system** session descriptors. SQL Server 10.0.2 includes a an additional type, **worktable** session descriptors (separate from user session descriptors).

#### system table

One of the data dictionary tables. System tables have an object ID that is less than 100. 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. The *System Administration Guide* includes a "SYBASE SQL Server System Tables Diagram" which shows which system tables are in *master* only and which system tables are in all databases.

#### transaction log

A system table (*syslogs*) in which all changes to the database are recorded.

#### virtual page

Actual offset on a device for a page within a database.

# Index

Symbols	632 error 2-60
@@servername	644 error 2-61
and 7214 error 2-405	701 error 2-64
@@servername global variable 2-525	702 error 2-66
8	703 error 2-68
	706 error 2-70
Numerics	707 error 2-71
<b>102</b> error 2-2	803 error 2-73
207 error 2-5	804 error 2-76
208 error 2-7	806 error 2-77
and 229 error 2-9	813 error 2-80
and 230 error 2-9	821 error 2-82
213 error 2-10	822 error 2-84
<b>225 error</b> 2-11	823 error 2-88
	840 error 2-90
<b>226</b> error 2-12 <b>230</b> error 2-14	903 error 2-94
<b>229</b> error 2-14	905 error 2-96
<b>232</b> error 2-15	906 error 2-98
<b>259 error</b> 2-16	911 error 2-99
<b>268</b> error 2-18	<b>913</b> error 2-101
and sp_help 2-18	<b>916</b> error 2-102
and sp_helpsort 2-18	<b>921</b> error 2-104
403 error 2-22	924 error 2-106
404 error 2-23	925 error 2-107
and 3626 error 2-23	<b>926</b> error 2-110
<b>428</b> error 2-24 <b>420</b> error 2-25	930 error 2-112
430 error 2-25 511 error 2-27	933 error 2-115
511 error 2-27 and 1709 array 2-27	945 error 2-117
and 1708 error 2-27	1105 error 2-120
515 error 2-29	and 2558 error 2-274
601 error 2-32	<b>1108</b> error 2-135, 2-250
603 error 2-34	1117 error 2-137
<b>605</b> error 2-36, 2-498	1120 error 2-139
and 2503 error 2-217	1129 error 2-141
611 error 2-40	1133 error 2-143
614 error 2-41	1134 error 2-146
623 error 2-45	<b>1142</b> error 2-147
624 error 2-47	1143 error 2-150
625 error 2-51	1203 error 2-153
629 error 2-54	<b>1204</b> error 2-154
<b>631 error</b> 2-56	1205 error 2-157

1207 error 2-160 1265 error 2-161 1501 error 2-163 1505 error 2-165 1508 error 2-167 and 1916 error 2-167 1509 error 2-169 1510 error 2-171 and 1105 error 2-171 1520 error 2-173 1530 error 2-175 1531 error 2-176 1601 error 2-179 1602 error 2-181 1605 error 2-183 1608 error 2-187 1613 error 2-189 1623 error 2-190 1702 error 2-193 1808 error 2-194 1809 error 2-196 1810 error 2-197 1811 error 2-198 1813 error 2-199 1902 error 2-200 1903 error 2-201 1904 error 2-202 1916 error 2-203 and 1508 error 2-168 2401 error 2-206 2402 error 2-208 2409 error 2-210 2501 error 2-213 2502 error 2-215 and 605 error 2-215 2503 error 2-217 and 605 error 2-217 2506 error 2-219 2507 error 2-222 2509 error 2-225 2510 error 2-228 2511 error 2-230 2513 error 2-232 2514 error 2-234

2517 error 2-237 2519 error 2-239 2520 error 2-240 2521 error 2-241 2524 error 2-247 2525 error 2-250 2526 error 2-253 2529 error 2-254 2540 error 2-256 and 7939 error 2-448 2541 error 2-261 2543 error 2-263 2544 error 2-265 2546 error 2-267 and 7939 error 2-448 2550 error 2-272 2558 error 2-274, 2-316 cause 2-275 prevention 2-278 2559 error 2-282 2571 error 2-284 2572 error 2-286 2573 error 2-288 2596 error 2-289 2601 error 2-293 2610 error 2-294 2615 error 2-296 and 1916 error 2-296 2620 error 2-298 2714 error 2-301 **2811 error** 2-303 and 1105 error 2-303 2812 error 2-304 2824 error 2-307 3002 error 2-309 3004 error 2-311 3101 error 2-313 3104 error 2-314 3105 error 2-316 3120 error 2-317 **3201** error 2-319 3202 error 2-322 3203 error 2-325 3225 error 2-327

3307 error 2-329 **3403** error 2-332 3414 error 2-334 3418 error 2-335 3425 error 2-336 3604 error 2-338 3621 error 2-339 **3626** error 2-340 3701 error 2-343 and 2714 error 2-343 3702 error 2-345 3703 error 2-347 3904 error 2-350 4002 error 2-353 4204 error 2-358 4207 error 2-362 4219 error 2-364 4305 error 2-367 4306 error 2-368 4806 error in bcp 2-371 5006 error 2-374 5013 error 2-376 5115 error 2-380 5123 error 2-383 5808 error 2-387 5813 error 2-389 6103 error 2-391 6902 error 2-393 7130 error 2-396 7205 error 2-399 7211 error 2-401 7212 error 2-403 7214 error 2-405 7215 error 2-406 7218 error 2-408 7220 error 2-409 7221 error 2-413 7223 error 2-416 7227 error 2-418 7364 error 2-421 7401 error 2-423 7402 error 2-425 7403 error 2-426

7404 error 2-427 7405 error 2-429 7406 error 2-430 7407 error 2-431 7408 error 2-432 7409 error 2-433 7410 error 2-434 7411 error 2-435 7412 error 2-436 7413 error 2-437 7414 errors 2-438 7415 errors 2-439 7902 error 2-441 7930 error 2-446 7939 error 2-448 and 2540 error 2-448 and 2546 error 2-448 7940 error 2-453 **7948** error 2-457 7949 error 2-458 8201 error 2-464 8203 error 2-465 8204 error 2-467 8207 error 2-468 8211 error 2-470 8402 error 2-472 8412 error 2-474 8704 error 2-476 17417 errors 2-478 17715 error 2-481 17716 error 2-482 17870 errors 2-483 17871 error 2-484 17872 error 2-485 17873 error 2-486 17874 error 2-487 17875 error 2-488 17903 error 2-489 17904 error 2-490 17905 error 2-491 17906 error 2-492 17910 error 2-493 18031 error 2-494 18032 error 2-495

18033 error 2-496

## A

Access errors 2-31 Error 601 2-32 Error 603 2-34 Error 605 2-36 Error 611 2-40 Error 614 2-41 Error 623 2-45 Error 624 2-47 Error 625 2-51 Error 629 2-54 Error 631 2-56 Error 632 2-60 Error 644 2-61 Account locked, and 4002 error 2-353 Ad-hoc updates 2-16 Adjust table and 2506 error 2-219 Allocation 605 error 2-36 1108 error 2-135 2502 error 2-215 2521 error 2-241 2525 error 2-250 2540 error 2-256 2541 error 2-261 2543 error 2-263 2544 error 2-265 2546 error 2-267 2559 error 2-282 3403 error 2-332 7939 error 2-448 7940 error 2-453 7949 error 2-458 and 2572 error 2-286 space allocation and 1105 error 2-120 structure 2-263, 2-267 Allocation pages and 1120 error 2-139 and 1129 error 2-141

and 1134 error 2-146 allow updates and 259 error 2-16 and 5808 error 2-387 allow\_dup\_row and 1508 error 2-167 and 1916 error 2-203 create index option 2-167 allows nulls by default and 515 error 2-29 alter database and 1808 error 2-194 and 2558 error 2-278 and 3225 error 2-327 for load 2-113 when database full 2-374 wrong dump device type 2-198 alter database errors 2-373 Error 5006 2-374 Error 5013 2-376 alter table and 3703 error 2-347 and expression limit 2-23 Arithmetic overflow and 232 error 2-15 Asynchronous I/O Sun systems 2-88, 2-382

### В

bcp and 4207 error 2-362 Error 4207 2-371 fast mode 2-371 use to shrink database 2-316 Buffer messages 2-498 mismatch 2-498 Buffer Manager errors 2-72 Error 803 2-73 Error 804 2-76 Error 806 2-77 Error 813 2-80 Error 821 2-82 Error 822 2-84 Error 823 2-88 Error 840 2-90 Buffer mismatch error 2-498 buildmaster and 5013 error 2-377

### С

Cache management problems 2-498 Character set conversion errors 2-205 Error 2401 2-206 Error 2402 2-208 Error 2409 2-210 Character set error 2-499 Character set support error 2-499 checkpoint and 623 error 2-45 and 624 error 2-47 and 631 error 2-59 and 822 error 2-85, 2-120 and 1207 error 2-160 and 2519 error 2-239 and 7413 error 2-437 Client process unexpected stop 2-187 Clustered index 2-280 and duplicate values 2-296 one per table 2-200 wrong page 2-54 cnblkmax configuration parameter. See sp\_configure cnmaxaio\_server configuration parameter. See sp\_configure cntrltype and 3201 error 2-319 Column limit on tables 2-193 name rules 2-193 **Command aborted** and 3621 error 2-339 Comments xx Compile errors 2-21 Error 403 2-22

Error 404 2-23 Error 428 2-24 Error 430 2-25 Composite index column limit 2-202 definition 2-201 config\_value. See sp\_configure configuration parameter 2-504 **Configured sockets** and 1605 error 2-183 Corrupt index 2-61, 2-228 Could not create shared memory error 2-501 create database and 906 error 2-98 and 1808 error 2-194 and 1809 error 2-196 and 1810 error 2-197 and 1813 error 2-199 and 2558 error 2-278 and 3225 error 2-327 and size of model 2-197 wrong dump device type 2-198 create index and 1505 error 2-165 and 1510 error 2-171 and 1530 error 2-175 and 1531 error 2-176 clustered index 1508 error 2-167 command failed 2-173 error with clustered index 2-171 exclusive option error 2-203 failure on duplicate key 2-165 failure on duplicate row 2-167 ignore\_dup\_key option 2-338 options and Error 1916 2-168 create table and 1702 error 2-193 Create utilities errors 2-192, 2-300 Error 1702 2-193 Error 1808 2-194 Error 1809 2-196 Error 1810 2-197 Error 1811 2-198

Error 1813 2-199 Error 1902 2-200 Error 1903 2-201 Error 1904 2-202 Error 1916 2-203 Error 2714 2-301 Creating error messages 1-2 csortbufsize configuration parameter 2-163 csortpgcount configuration parameter 2-163 Current process infected with %d error 2-502

# D

Data and log mapping 2-275 cache, sizing 2-65 page 2-230 Data on log error 2-275 Data Server errors 2-352 Error 4002 2-353 data\_pgs 2-147 Database configured size 2-316 dump Error 3201 2-319 failed recovery and suspect status 2-334, 2-336 failure during recovery 2-49, 2-334 full 2-122, 2-130 load Error 3101 2-313 loading after corruption 2-49 maintenance 2-69 model and 1105 error 2-125, 2-128 naming conventions 2-7 no chkpt on recovery option 2-128, 2-131, 2-133 no performance benefit on a single device 2-374 owner. See dbo recovery 2-47 select into/bulkcopy option 2-362

shrinking 2-316 single-user option 2-128, 2-131, 2-133 sort order 2-317 suspect 2-49, 2-104, 2-110 dataserver and 1623 error 2-190 Datatype storage size. See System Administration Guide dbcc checkalloc and 605 error 2-36 and 2525 error 2-250 and 2541 error 2-261 and 2543 error 2-263 and 2544 error 2-265 and 2546 error 2-267 and 2558 error 2-274 and 2571 error 2-284 and 2572 error 2-286 and 7939 error 2-448 and 7940 error 2-453 and 7949 error 2-458 dbcc checkcatalog and 2513 error 2-232 and 2514 error 2-234 and 2517 error 2-237 and 2519 error 2-239 and 2520 error 2-240 and 2550 error 2-272 dbcc checkdb and 605 error 2-36 and 2506 error 2-219 and 2507 error 2-222 and 2509 error 2-225 and 2520 error 2-240 and 2524 error 2-247 and 7902 error 2-441 and 7930 error 2-446 and 7948 error 2-457 dbcc checktable and 2506 error 2-219 and 2507 error 2-222 and 2509 error 2-225 and 2524 error 2-247

and 2596 error 2-289 and 7902 error 2-441 and 7930 error 2-446 and 7948 error 2-457 dbcc dbrepair and 2520 error 2-240 and 2571 error 2-284 and 2573 error 2-288 and 3702 error 2-345 dbcc errors 2-212, 2-440 Error 2501 2-213 Error 2502 2-215 Error 2503 2-217 Error 2506 2-219 Error 2507 2-222 Error 2509 2-225 Error 2510 2-228 Error 2511 2-230 Error 2513 2-232 Error 2514 2-234 Error 2517 2-237 Error 2519 2-239 Error 2520 2-240 Error 2521 2-241 Error 2524 2-247 Error 2525 2-250 Error 2526 2-253 Error 2529 2-254 Error 2540 2-256 Error 2541 2-261 Error 2543 2-263 Error 2544 2-265 Error 2546 2-267 Error 2550 2-272 Error 2558 2-274 Error 2559 2-282 Error 2571 2-284 Error 2572 2-286 Error 2573 2-288 Error 2596 2-289 Error 7902 2-441 Error 7930 2-446 Error 7939 2-448 Error 7940 2-453

Error 7948 2-457 Error 7949 2-458 dbcc indexalloc and 7939 error 2-448 and 7940 error 2-453 and 7949 error 2-458 dbcc memusage and 2571 error 2-284 dbcc page and 806 error 2-77 and 2571 error 2-284 dbcc tablealloc and 7939 error 2-448 and 7940 error 2-453 and 7949 error 2-458 dbcc traceoff and 2571 error 2-284 dbcc traceon and 2571 error 2-284 dbo and 208 error and 268 error 2-18 and 2812 error 2-304 and 4806 error 2-371 and select/into bulkcopy option 2-371 dbwritetext and 268 error 2-18 and 7130 error 2-396 Deadlock and 1205 error 2-157 and long-running transaction 2-157 **DECnet** protocol and user connections 2-180 **Delimited identifiers** 2-5 **Descriptor Manager errors** 2-463 Error 8201 2-464 Error 8203 2-465 Error 8204 2-467 Error 8207 2-468 Error 8211 2-470 Device offline 2-84, 2-90 removed 2-84, 2-90 renamed 2-84, 2-90

dinit kernel error message 2-86 Disk errors 2-379 Error 5115 2-380 Error 5123 2-383 disk init failure 2-380 failure and Error 5123 2-383 size 2-381, 2-384 vdevno 2-381 Distributed database network errors 2-398 Error 7205 2-399 Error 7211 2-401 Error 7212 2-403 Error 7214 2-405 Error 7215 2-406 Error 7218 2-408 Error 7220 2-409 Error 7221 2-413 Error 7223 2-416 Error 7227 2-418 Distribution page 2-251 dpages 2-173 drop errors 2-342 Error 3701 2-343 Error 3702 2-345 Error 3703 2-347 drop index and 3703 error 2-347 drop trigger and 3703 error 2-347 Dstart I/O request repeatedly delayed error 2-504 dump and load errors 2-308, 2-357 Error 3002 2-309 Error 3004 2-311 Error 3101 2-313 Error 3104 2-314 Error 3105 2-316 Error 3120 2-317 Error 3201 2-319 Error 3202 2-322 Error 3203 2-325 Error 3225 2-327

Error 4204 2-358 Error 4207 2-362 Error 4219 2-364 Error 4305 2-367 Error 4306 2-368 dump database and 3004 error 2-311 and 3104 error 2-314 and uninitialized pages 2-311 failure and Error 3002 2-309 dump transaction and 2519 error 2-239 and 4204 error 2-358 and 4207 error 2-362 and 4219 error 2-364 not allowed 2-362 with no\_log 2-82 **Duplicate key** and create index 2-165 Duplicate row and create index 2-167

## E

Error creating error messages 1-2 dinit kernel error message 2-86 hard 605 2-36 read Error 822 2-84 reporting to Sybase 1-2 threshold 2-477 transient 605 2-36 variables in error message text 1-2 write Error 822 2-84 Error 102 2-2 Error 207 2-5 Error 208 2-7 and 229 error 2-9 and 230 error 2-9 Error 213 2-10 Error 225 2-11 Error 226 2-12 Error 229 2-14 Error 232 2-15

Error 259 2-16 Error 268 2-18 and sp\_help 2-18 and sp\_helpsort 2-18 Error 403 2-22 Error 404 2-23 and 3626 error 2-23 Error 428 2-24 Error 430 2-25 Error 511 2-27 and 1708 error 2-27 Error 515 2-29 Error 601 2-32 Error 603 2-34 Error 605 2-36 and 2502 error 2-215 and 2503 error 2-217 Error 611 2-40 Error 614 2-41 Error 623 2-45 Error 624 2-47 Error 625 2-51 Error 629 2-54 Error 631 2-56 Error 632 2-60 Error 644 2-61 Error 701 2-64 Error 702 2-66 Error 703 2-68 Error 706 2-70 Error 707 2-71 Error 803 2-73 Error 804 2-76 Error 806 2-77 Error 813 2-80 Error 821 2-82 Error 822 2-84 Error 823 2-88 Error 840 2-90 Error 903 2-94 Error 905 2-96 Error 906 2-98 Error 911 2-99 Error 913 2-101

Error 916 2-102 Error 921 2-104 Error 924 2-106 Error 925 2-107 Error 926 2-110 Error 930 2-112 Error 933 2-115 Error 945 2-117 Error 1105 2-120 and 2558 error 2-274 Error 1108 2-135, 2-250 Error 1117 2-137 Error 1120 2-139 Error 1129 2-141 Error 1133 2-143 Error 1134 2-146 Error 1142 2-147 Error 1143 2-150 Error 1203 2-153 Error 1204 2-154 Error 1205 2-157 Error 1207 2-160 Error 1265 2-161 Error 1501 2-163 Error 1505 2-165 Error 1508 2-167 and 1916 error 2-167 Error 1509 2-169 Error 1510 2-171 and 1105 error 2-171 Error 1520 2-173 Error 1530 2-175 Error 1531 2-176 Error 1601 2-179 Error 1602 2-181 Error 1605 2-183 Error 1608 2-187 Error 1613 2-189 Error 1623 2-190 Error 1702 2-193 Error 1808 2-194 Error 1809 2-196 Error 1810 2-197 Error 1811 2-198

Error 1813 2-199 Error 1902 2-200 Error 1903 2-201 Error 1904 2-202 Error 1916 2-203 and 1508 error 2-168 Error 2401 2-206 Error 2402 2-208 Error 2409 2-210 Error 2501 2-213 Error 2502 2-215 Error 2503 2-217 and 605 error 2-217 Error 2506 2-219 Error 2507 2-222 Error 2509 2-225 Error 2510 2-228 Error 2511 2-230 Error 2513 2-232 Error 2514 2-234 Error 2517 2-237 Error 2519 2-239 Error 2520 2-240 Error 2521 2-241 Error 2524 2-247 Error 2525 2-250 Error 2526 2-253 Error 2529 2-254 Error 2540 2-256 and 7939 error 2-448 Error 2541 2-261 Error 2543 2-263 Error 2544 2-265 Error 2546 2-267 and 7939 error 2-448 Error 2550 2-272 Error 2558 2-274, 2-316 cause 2-275 prevention 2-278 Error 2559 2-282 Error 2571 2-284 Error 2572 2-286 Error 2573 2-288 Error 2596 2-289

Error 2601 2-293 Error 2610 2-294 Error 2615 2-296 and 1916 error 2-296 Error 2620 2-298 Error 2714 2-301 Error 2811 2-303 and 1105 error 2-303 Error 2812 2-304 Error 2824 2-307 Error 3002 2-309 Error 3004 2-311 Error 3101 2-313 Error 3104 2-314 Error 3105 2-316 Error 3120 2-317 Error 3201 2-319 Error 3202 2-322 Error 3203 2-325 Error 3225 2-327 Error 3307 2-329 Error 3403 2-332 Error 3414 2-334 Error 3418 2-335 Error 3425 2-336 Error 3604 2-338 Error 3621 2-339 Error 3626 2-340 Error 3701 2-343 and 2714 error 2-343 Error 3702 2-345 Error 3703 2-347 Error 3904 2-350 Error 4002 2-353 Error 4204 2-358 Error 4207 2-362 Error 4219 2-364 Error 4305 2-367 Error 4306 2-368 **Error 4806** in bcp 2-371 Error 5006 2-374 Error 5013 2-376 Error 5115 2-380

Error 5123 2-383 Error 5808 2-387 Error 5813 2-389 Error 6103 2-391 Error 6902 2-393 Error 7130 2-396 Error 7205 2-399 Error 7211 2-401 Error 7212 2-403 Error 7214 2-405 Error 7215 2-406 Error 7218 2-408 Error 7220 2-409 Error 7221 2-413 Error 7223 2-416 Error 7227 2-418 Error 7364 2-421 Error 7401 2-423 Error 7402 2-425 Error 7403 2-426 Error 7404 2-427 Error 7405 2-429 Error 7406 2-430 Error 7407 2-431 Error 7408 2-432 Error 7409 2-433 Error 7410 2-434 Error 7411 2-435 Error 7412 2-436 Error 7413 2-437 Error 7414 2-438 Error 7415 2-439 Error 7902 2-441 Error 7930 2-446 Error 7939 2-448 and 2540 error 2-448 and 2546 error 2-448 Error 7940 2-453 Error 7948 2-457 Error 7949 2-458 Error 8201 2-464 Error 8203 2-465 Error 8204 2-467 Error 8207 2-468

Error 8211 2-470 Error 8402 2-472 Error 8704 2-476 Error 8412 2-474 Error 17417 2-478 Error 17715 2-481 Error 17716 2-482 Error 17870 2-483 Error 17871 2-484 Error 17872 2-485 Error 17873 2-486 Error 17874 2-487 Error 17875 2-488 Error 17903 2-489 Error 17904 2-490 Error 17905 2-491 Error 17906 2-492 Error 17910 2-493 Error 18031 2-494 Error 18032 2-495 Error 18033 2-496 Error and exception handling errors 2-337 Error 3604 2-338 Error 3621 2-339 Error 3626 2-340 Error log 1-1 Error severity levels 1-1 Errors access 2-31 alter database 2-373 Buffer Manager 2-72 character set conversion 2-205 compile 2-21 create utilities 2-192, 2-300 Data Server 2-352 dbcc 2-212, 2-440 **Descriptor Manager** 2-463 disk 2-379 distributed database network 2-398 drop 2-342 dump and load 2-308, 2-357 error and exception handling 2-337 initialization 2-178

insert 2-292 Lock Manager 2-152 Memory Manager 2-63 **Open Database Manager** 2-93 Page Manager 2-119 Parser 2-1, 2-420 Procedure Manager 2-302 process kill 2-390 query processor 2-26 reconfigure 2-386 recovery 2-331 Sequencer 2-4 Site Buffer Manager 2-475 Sort Manager 2-162 sysindexes Manager 2-471 Text Manager 2-395 threshold 2-422 timestamp 2-392 Transaction 2-349 undo and redo 2-328 unnumbered 2-497 Execution plan and 225 error 2-11 Expression limit on queries 2-23 Extent 2-263, 2-267, 2-274

# F

Failed to open device error 2-509 Fast bcp method 2-371 File already in use error 2-510 Foreign key 2-24

### G

Global variable @@servername 2-525 Guardword 2-526

#### Η

Hard 605 2-36 Help xxii holdlock and 1205 error 2-157

### I

I/OError 822 2-84 Error 823 2-88 Error 5115 2-380 **Identifiers** delimited 2-5 ignore\_dup\_key 2-338 and 1916 error 2-203 and 3604 error 2-338 ignore\_dup\_row and 1508 error 2-167 and 1916 error 2-203 create index option 2-167 Index clustered 2-54, 2-280 clustered index and sorted\_data 2-175 column limit on composite index 2-202 composite index 2-201 corrupt 2-47, 2-61, 2-228 deleting duplicate index values 2-165 deleting duplicate rows 2-297 determining type 2 distribution page 2-251 duplicate values 2-293, 2-296 index ID 2-61 nonclustered 2-61 page 2-230 page number 2-61 single clustered index limit 2-200 table corrupt error 2-230 unique index and 2601 error 2-293 unique index and duplicate values 2-338 Initialization errors 2-178 Error 1601 2-179 Error 1602 2-181 Error 1605 2-183 Error 1608 2-187 Error 1613 2-189

```
Error 1623 2-190
insert errors 2-292
  Error 2601 2-293
  Error 2610 2-294
  Error 2615 2-296
  Error 2620 2-298
installmaster
  and 511 error 2-27
  and 2812 error 2-305
  and permissions error 2-306
Interfaces file
  and 1602 error 2-181
  and 1605 error 2-183
  and kernel message 2-513
isql
  -J flag 2-499
```

## J

Join operation limits 2-66

# K

Kernel message device full 2-374 network information 2-513 stack guardword 2-526 and 3626 error 2-340 Keys 2-230

### L

Ict\_admin and 7406 error 2-430 and 7408 error 2-432 and 17716 error 2-482 Leaf page 2-228, 2-230 Limit and or or expressions 2-23 on joins 2-66 on search conditions 2-66 Ioad database and 631 error 2-56 and 3105 error 2-316

and 3225 error 2-327 and 3425 error 2-336 do not use while loading 2-313 failure 2-112 when sort orders differ 2-317 load transaction and 631 error 2-56 and 3425 error 2-336 and 4306 error 2-368 Loading a database 2-275 after corruption 2-49 Lock on an object 2-157 Lock Manager errors 2-152 Error 1203 2-153 Error 1204 2-154 Error 1205 2-157 Error 1207 2-160 Error 1265 2-161 Locking and 1510 error 2-171 Locks and 1265 error 2-161 and holdlock 2-157 running out of 2-154 running out of locks and Error 1510 2-172 Locksleep and 1207 error 2-160 Log and data mapping 2-275 full 2-122, 2-130 "Log on data" 2-276 Logical page 2-77, 2-245, 2-277 Logical Volume Manager 2-89 Login request rejected, 4002 error 2-353 LVM 2-89

#### Μ

Mail address tsg@sybase.com xx Mapping

user-defined segments 2-278 master database and 1809 error 2-196 error when extending 2-376 master..sysmessages 1-1 Maximum bytes per index 2-201 clustered indexes on a table 2-200 columns in a table 2-193 columns per index 2-202 Memory lack of 2-389 maximum amount 2-68 too fragmented 2-511 memory configuration parameter. See sp\_configure Memory Manager errors 2-63 Error 701 2-64 Error 702 2-66 Error 703 2-68 Error 706 2-70 Error 707 2-71 Memory too fragmented error 2-511 Minimally logged operation and dumping to transaction log 2-20 model database how to check size 2-197 log full 2-133 MULTARG and 3202 error 2-322

### Ν

Naming conventions 2-7, 2-304 Network 1602 error 2-181 1605 error 2-183 1608 error 2-187 1613 error 2-189 information message 2-513 No more alarms available error 2-515 Non-logged operation 2-371 and 268 error 2-18 nowait and 1207 error 2-160

### 0

OAM entry 2-453, 2-458 OAM page 2-147 and 1133 error 2-143 **Object Allocation Map** and 1120 error 2-139 and 1129 error 2-141 and 1133 error 2-143 and 1134 error 2-146 and 1142 error 2-147 and 1143 error 2-150 and 7939 error 2-448 Object does not exist 2-7 Object name 2-7 Offset table and 2507 error 2-222 **Open Database Manager errors** 2-93 Error 903 2-94 Error 905 2-96 Error 906 2-98 Error 911 2-99 Error 913 2-101 Error 916 2-102 Error 921 2-104 Error 924 2-106 Error 925 2-107 Error 926 2-110 Error 930 2-112 Error 933 2-115 Error 945 2-117 open databases configuration parameter. See sp\_configure open objects configuration parameter. See sp\_configure Open objects parameter may be too low error 2-517 or expression limit 2-23 os\_attach\_region error 2-519 os\_create\_region errors 2-520

### Ρ

Page See also Allocation chain 2-217, 2-241, 2-267 distribution 2-251 empty page error 2-282 incorrect free byte offset 2-289 linkage error 2-217 logical 2-245 not found 2-254 uninitialized 2-311 Page allocation error. See Allocation Page Manager errors 2-119 Error 1105 2-120 Error 1108 2-135 Error 1117 2-137 Error 1120 2-139 Error 1129 2-141 Error 1133 2-143 Error 1134 2-146 Error 1142 2-147 Error 1143 2-150 Parser errors 2-1, 2-420 Error 102 2-2 Error 7364 2-421 Password incorrect, and 4002 error 2-353 Permissions and 229 error 2-14 **Procedure cache** and 701 error 2-64 shortage 2-64 sizing 2-64 procedure cache configuration parameter. See sp\_configure Procedure Manager errors 2-302 Error 2811 2-303 Error 2812 2-304 Error 2824 2-307 Process killed, cleanup and Error 6103 2-391 sleeping process and 1207 error 2-160 Process kill errors 2-390 6103 2-391

Pss structure error 2-179

### Q

Query plan 2-64, 2-68 Query processor errors 2-26 Error 511 2-27 Error 515 2-29 Query tree 2-68

### R

Read Error 822 2-84 Error 823 2-89 kernel error 2-523 Read/write error 2-523 Reboot problems 2-334 reconfigure failure 2-389 reconfigure errors 2-386 Error 5808 2-387 Error 5813 2-389 Recovery and 631 error 2-56 and 906 error 2-94, 2-98 and 933 error 2-115 and 3403 error 2-332 and 3425 error 2-336 errors 2-47 failure 2-334 Recovery errors 2-331 Error 3403 2-332 Error 3414 2-334 Error 3418 2-335 Error 3425 2-336 Recursive calls 2-526 Remote procedure call 2-513 and 7205 error 2-399 and 7211 error 2-401 no site name received error 2-405 Reporting errors to Sybase 1-2 reserved\_pgs 2-147 Resvar 2-25

RID. See Row ID Row maximum length 2-27 Row compare and 1509 error 2-169 Row ID 2-45, 2-47 rowcnt 2-147 RPC 2-513 can't open connection 2-418 can't send to site error 2-406 login to site failed error 2-413 site not found error 2-408, 2-409 run\_value. See sp\_configure

# S

S MSG explanation 2-230 sa role and 2571 error 2-284 sdes 2-73, 2-76 Search condition limits 2-66 Segment default 2-274 full 2-120 log 2-274 mapping 2-274 system 2-274 select into and 4207 error 2-362 select into/bulkcopy 2-362 how to activate 2-19 Sequencer errors 2-4 Error 207 2-5 Error 208 2-7 Error 213 2-10 Error 225 2-11 Error 226 2-12 Error 229 2-14 Error 232 2-15 Error 259 2-16 Error 268 2-18 Server is unnamed error 2-525 Session descriptors 2-73, 2-76

and 603 error 2-34 system 2-34 user 2-34 worktable 2-34 Severity levels 1-1 Shrinking a database 2-316 shutdown and 631 error 2-59 and 1207 error 2-160 Single-user mode and 924 error 2-106 and 2572 error 2-286 Site Buffer Manager errors 2-475 Error 8704 2-476 Sleeping processes 2-160 Sort failure 2-173 and Error 1510 2-171 and Error 1520 2-173 Sort Manager errors 2-162 Error 1501 2-163 Error 1505 2-165 Error 1508 2-167 Error 1509 2-169 Error 1510 2-171 Error 1520 2-173 Error 1530 2-175 Error 1531 2-176 Sort order and 2714 error 2-301 and dump/load 2-317 changing 2-317 sorted\_data and 1530 error 2-175 and 1531 error 2-176 sp\_addmessage 1-2 sp\_addsegment and 2558 error 2-278 and 2571 error 2-284 sp\_addserver 2-525 and 7214 error 2-405 sp addthreshold and 2571 error 2-284 and 17870 error 2-483 and 17871 error 2-484

and 17872 error 2-485 and 17874 error 2-487 and 17875 error 2-488 and 18031 error 2-494 sp\_addumpdevice and 3201 error 2-319 sp\_configure and 1510 error 2-172 and 2812 error 2-305 cnblkmax configuration parameter 2-504 cnmaxaio\_engine configuration parameter 2-504 cnmaxaio\_server configuration parameter 2-504 config value 2-154 locks configuration parameter 2-154 memory configuration parameter 2-64, 2-341, 2-389 memory configuration parameter and stack size 2-527 open databases configuration parameter 2-96 open objects configuration parameter 2-11, 2-468 procedure cache configuration parameter 2-64 run value 2-154 running out of locks 2-154 stack size configuration parameter 2-341 user connections configuration parameter 2-179, 2-478 sp\_dboption and 226 error 2-13 and 268 error 2-18 and 515 error 2-29 and 924 error 2-106 and 4207 error 2-363 and 4806 error 2-371 and *dbo* 2-18 enabling select into/bulkcopy 2-19 sp\_diskblock (user must add) 2-523 sp\_diskdefault

and 1808 error 2-194 sp dropsegment and 2550 error 2-272 and 2558 error 2-278 and 2571 error 2-284 sp\_dropthreshold and 2571 error 2-284 and 17871 error 2-484 and 17875 error 2-488 and 17903 error 2-489 and 17904 error 2-490 and 17905 error 2-491 and 18031 error 2-494 sp\_extendsegment and 2558 error 2-278 and 2571 error 2-284 sp\_help and 268 error 2-18 and 2812 error 2-304 sp helpdb and 924 error 2-106 and sysusages 2-279 sp\_helpdevice valid dump device type 2-198 sp\_helpserver checking SQL Server status 2-513 sp\_helpsort and 268 error 2-18 sp helpthreshold and 17910 error 2-493 sp\_logdevice and 2558 error 2-278 and 2571 error 2-284 sp\_modifythreshold and 2571 error 2-284 and 7403 error 2-426 and 17870 error 2-483 and 17871 error 2-484 and 17872 error 2-485 and 17874 error 2-487 and 17875 error 2-488 and 17904 error 2-490 and 18031 error 2-494 and 18032 error 2-495

sp\_placeobject and 2571 error 2-284 sp\_remap and 2571 error 2-284 sp\_resetstatus (user must add) 2-78, 2-85, 2-91, 2-115 sp\_who and 924 error 2-106 SQL Server character set support error 2-499 stack space 2-340 Stack guardword 2-340 Stack guardword error 2-526 Stack size 2-340, 2-341 and stack guardword error 2-526 Stored procedures and 913 error 2-101 compilation 2-68 creating system procedures 2-304 maintenance 2-69 query plan 2-64, 2-68 query tree 2-68 resolution 2-68 Style conventions xxii Sun systems asynchronous I/O 2-88, 2-382 Suspect database 2-104, 2-110 and 624 error 2-49 and 822 error 2-84 and 840 error 2-91 and 3414 error 2-334, 2-336 and dbcc dbrepair 2-288 resetting suspect status 2-78, 2-85, 2-91, 2-115 sybload and 803 error 2-74 **Syntax** style conventions xxi to xxii sysalternates and 916 error 2-102 syscharsets and 605 error 2-37 syscolumns and 2514 error 2-234

sysconfigures and 1204 error 2-154 sysconstraints and 3703 error 2-347 sysdatabases and 911 error 2-99 and 913 error 2-101 and 1813 error 2-199 and 2520 error 2-240 and 3418 error 2-335 sysindexes and 629 error 2-54 and 903 error 2-94 and 906 error 2-98 and 1133 error 2-143 and 2541 error 2-261 and 8402 error 2-472 and 8412 error 2-474 and allocation Error 2525 2-250 sysindexes Manager errors 2-471 Error 8402 2-472 Error 8412 2-474 syslogins and 4002 error 2-353 syslogs and 933 error 2-115 and 1105 error 2-122, 2-130 and 2519 error 2-239 and 3403 error 2-332 and 4219 error 2-364 sysobjects and 906 error 2-98 and 2517 error 2-237 and 3703 error 2-347 finding system tables with ID **value** 2-16 sysprocedures and 2517 error 2-237 and 2811 error 2-303 sysreferences and 3703 error 2-347, 2-348 sysremotelogins and 4002 error 2-353 sysservers

and 7214 error 2-405 System Administrator 1-1 System catalog. See System table or individual system table name System memory 2-64 System table 2-230 ad-hoc updates 2-16 errors 2-228 syscolumns 2-234 sysconfigures 2-154 sysmessages 1-1 sysusages 2-278 systhresholds and 601 error 2-32 and 7404 error 2-427 and 7405 error 2-429 and 7407 error 2-431 and 7411 error 2-435 and 17870 error 2-483 and 17903 error 2-489 and 17910 error 2-493 systypes and 2514 error 2-234 sysusages 2-278 and 2550 error 2-272 sysusers and 916 error 2-102

# Т

t\_rcvconnect error 2-528 Table column limit 2-193 column name rules 2-193 corrupt error 2-234 index corruption 2-230 *master..sysmessages* 1-1 *sysconfigures* 2-154 Table corrupt error 2503 error 2-217 2510 error 2-228 2511 error 2-230 2521 error 2-241 2525 error 2-250

2540 error 2-256 2541 error 2-261 2543 error 2-263 2546 error 2-267 tempdb and 806 error 2-77 and select/into bulkcopy option 2-18 log full 2-124 **Temporary tables** and 208 error 2-7 and 2714 error 2-301 Text Manager errors 2-395 Error 7130 2-396 Threshold errors 2-422, 2-477 Error 7401 2-423 Error 7402 2-425 Error 7403 2-426 Error 7404 2-427 Error 7405 2-429 Error 7406 2-430 Error 7407 2-431 Error 7408 2-432 Error 7409 2-433 Error 7410 2-434 Error 7411 2-435 Error 7412 2-436 Error 7413 2-437 Error 7414 2-438 Error 7415 2-439 Error 17417 2-478 Error 17715 2-481 Error 17716 2-482 Error 17870 2-483 Error 17871 2-484 Error 17872 2-485 Error 17873 2-486 Error 17874 2-487 Error 17875 2-488 Error 17903 2-489 Error 17904 2-490 Error 17905 2-491 Error 17906 2-492 Error 17910 2-493 Error 18031 2-494

Error 18032 2-495 Error 18033 2-496 Time out process times out 2-513 Timestamp and 933 error 2-115 and 4305 error 2-367 Timestamp errors 2-392 Error 6902 2-393 Trace flag 3601 1-5 Transaction long transactions and deadlocks 2-157 Transaction errors 2-349 Error 3904 2-350 Transaction log and 3403 error 2-332 Transact-SQL syntax conventions xxi to xxii Transient 605 2-36 tsg@sybase.com mail address xx Type ID mismatch problems 2-234

# U

udstartio 2-86 Undo and redo errors 2-328 Error 3307 2-329 Uninitialized page 2-311 unique and create index 2-165 UNIX asynch I/O and 5115 error 2-382 Unnumbered errors 2-497 buffer mismatch 2-498 character set 2-499 could not create shared memory 2-501 current process infected with %d 2-502 dstart I/O request repeatedly delayed 2-504 failed to open device 2-509

file already in use 2-510 memory too fragmented 2-511 network information 2-513 no more alarms available 2-515 open objects parameter may be too low 2-517 os\_attach\_region 2-519 os\_create\_region 2-520 read-write error 2-523 server is unnamed 2-525 stack guardword 2-526 t\_rcvconnect 2-528 used\_pgs 2-147 User connections and 701 error 2-64 and 1605 error 2-183 and DECnet protocol 2-180 running out of 2-179 user connections configuration parameter. See sp\_configure **User-defined segments** mapping 2-278 usertype problems 2-234

# V

Variables in error messages 1-1 Virtual device number 2-381, 2-383, 2-384 Virtual disk number 2-523 Virtual page 2-77 Virtual sockets and 1605 error 2-183

#### W

where clause limitations 2-23 Write Error 822 2-84 kernel error 2-523 writetext and 268 error 2-18 and 7130 error 2-396 writetext error 2-396