SYBASE®

Troubleshooting and Recovery Guide

Sybase® IQ

12.7

DOCUMENT ID: DC36584-01-1270-01

LAST REVISED: June 2006

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.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book .... **v**

CHAPTER 1 **Troubleshooting Hints** **1**

Solutions for specific conditions 2

Server recovery and database repair 2

Server operational issues... 3

Database connection issues 13

Multiplex issues 17

dbisql/dbisqlc issues... 18

Resource issues 19

Processing issues 26

Performance issues... 31

Sybase Central issues... 32

Troubleshooting network communications.................................... 32

Ensuring that you are using compatible protocols.................. 33

Ensuring that you have current drivers................................... 33

Switching off your computer between reboots 33

Diagnosing your protocol stack layer by layer........................ 34

Testing a NetBIOS protocol stack 34

Testing a TCP/IP protocol stack... 34

Diagnosing wiring problems 36

Checking common network communications problems 36

Diagnostic tools... 38

The sp_iqstatus stored procedure.. 38

The sp_iqcheckdb stored procedure 40

Checking database and server startup option values 40

Finding the currently executing statement.............................. 41

Logging server requests... 41

Connection for collecting diagnostic information 43

Diagnosing communications issues 44

Reporting problems to Technical Support..................................... 44

Collecting diagnostic information using getiqinfo 44

Information collected by getiqinfo... 46

Correlating connection information.. 47

Another source of helpful information 49
Checklist: information for Technical Support 50

CHAPTER 2 **System Recovery and Database Repair 51**
Recovery and repair overview 51
Normal recovery 52
Database verification 52
The sp_iqcheckdb stored procedure 53
sp_iqcheckdb output 58
Resource issues running sp_iqcheckdb 61
Database repair 62
Analyzing index errors 63
Repairing index errors 66
Analyzing allocation problems 69
Repairing allocation problems 71
Forced recovery mode 73
Starting servers in forced recovery mode 74
Recovering leaked space 76
Recovering multiplex databases 78
Emergency recovery without a transaction log 79
-f recovery option 79
Handling unrepairable problems 80
Index problems that cannot be repaired 81
Dropping inconsistent indexes, tables, or columns 81
DBCC error messages 83

Index 87

About This Book

Subject

This book is a reference guide for troubleshooting Sybase® IQ. While other books provide more context on how to carry out particular tasks, this book offers suggestions for resolving various problems you may occasionally encounter, including server recovery and database repair.

Audience

This book is for all users of Sybase IQ. Use it in conjunction with other books in the documentation set.

How to use this book

The following table shows which sections fit a particular interest or need.

Table 1: Guide to using this book

To do this...	***Read this section...***
Troubleshoot specific problems	"Solutions for specific conditions" on page 2
Troubleshoot network problems	"Troubleshooting network communications" on page 32
Collect diagnostic information	"Diagnostic tools" on page 38 "Collecting diagnostic information using getiqinfo" on page 44
Find Technical Support information	"Reporting problems to Technical Support" on page 44
Recover databases under normal conditions	"Normal recovery" on page 52
Verify database consistency	"Database verification" on page 52
Repair index problems	"Analyzing index errors" on page 63
Repair allocation problems	"Analyzing allocation problems" on page 69
Recover databases under special conditions	"Forced recovery mode" on page 73
Resolve unrepairable problems	"Handling unrepairable problems" on page 80

Windows platforms

The Windows information in this book applies to all supported Windows platforms, unless noted otherwise. For supported Windows platforms, see the *Release Bulletin Sybase IQ for Windows*.

Related documents

Documentation for Sybase IQ:

- *Introduction to Sybase IQ*

  Read and try the hands-on exercises if you are unfamiliar with Sybase IQ or with the Sybase Central™ database management tool.

- *New Features in Sybase IQ 12.7*

  Read just before or after purchasing Sybase IQ for a list of new features.

- *Sybase IQ Performance and Tuning Guide*

  Read to understand query optimization, design, and tuning issues for very large databases.

- *Sybase IQ Reference Manual*

  Read for a full description of the SQL language, stored procedures, data types, and system tables supported by Sybase IQ.

- *Sybase IQ System Administration Guide*

  Read to understand administration issues such database creation and load operations, data security and integrity, server startup and connection, and multiplex operations.

- *Sybase IQ Error Messages*

  Refer to IQ error messages which are referenced by SQLCode, SQLState, and Sybase error code, and SQL preprocessor errors and warnings.

- *Sybase IQ Utility Guide*

  Read for Sybase IQ utility program reference material, such as available syntax, parameters, and options.

- *Sybase IQ Installation and Configuration Guide*

  Read the edition for your platform before and while installing Sybase IQ, when migrating to a new version of Sybase IQ, or when configuring Sybase IQ for a particular platform.

- *Sybase IQ Release Bulletin*

  Read just before or after purchasing Sybase IQ for last minute changes to the product and documentation. Read for help if you encounter a problem.

- *Large Objects Management in Sybase IQ*

  Read to understand storage and retrieval of Binary Large Objects (BLOBs) and Character Large Objects (CLOBs) within the Sybase IQ data repository. You need a separate license to install this product option.

- *Encrypted Columns in Sybase IQ*

  Read to understand the use of user encrypted columns within the Sybase IQ data repository. You need a separate license to install this product option.

Sybase IQ and Adaptive Server Anywhere
Because Sybase IQ is an extension of Adaptive Server® Anywhere, a component of SQL Anywhere® Studio, IQ supports many of the same features as Adaptive Server Anywhere. The IQ documentation set refers you to SQL Anywhere Studio documentation where appropriate.

Documentation for Adaptive Server Anywhere:

- *Adaptive Server Anywhere Programming Guide*

  Intended for application developers writing programs that directly access the ODBC, Embedded SQL™, or Open Client™ interfaces, this book describes how to develop applications for Adaptive Server Anywhere.

- *Adaptive Server Anywhere Database Administration Guide*

  Intended for all users, this book covers material related to running, managing, and configuring databases and database servers.

- *Adaptive Server Anywhere SQL Reference Manual*

  Intended for all users, this book provides a complete reference for the SQL language used by Adaptive Server Anywhere. It also describes the Adaptive Server Anywhere system tables and procedures.

You can also refer to the Adaptive Server Anywhere documentation in the SQL Anywhere Studio 9.0.2 collection on the Sybase Product Manuals Web site. To access this site, go to Product Manuals at http://www.sybase.com/support/manuals/.

Other sources of information

Use the Sybase Getting Started CD, the SyBooks CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

  Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

  Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.
- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

  To access the Sybase Product Manuals Web site, go to Product Manuals at http://www.sybase.com/support/manuals/.
- Infocenter is an online version of SyBooks that you can view using a standard Web browser. To access the Infocenter Web site, go to Sybooks Online Help at http://infocenter.sybase.com/help/index.jsp.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest information on product certifications**

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Click Certification Report.
3. In the Certification Report filter select a product, platform, and timeframe and then click Go.
4. Click a Certification Report title to display the report.

❖ Finding the latest information on component certifications

1. Point your Web browser to Availability and Certification Reports at http://certification.sybase.com/.
2. Either select the product family and product under Search by Product; or select the platform and product under Search by Platform.
3. Select Search to display the availability and certification report for the selection.

❖ Creating a personalized view of the Sybase Web site (including support pages)

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ Finding the latest information on EBFs and software maintenance

1. Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.
2. Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
3. Select a product.
4. Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

   Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.
5. Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Syntax conventions

This documentation uses the following syntax conventions in syntax descriptions:

- **Keywords** SQL keywords are shown in UPPER CASE. However, SQL keywords are case insensitive, so you can enter keywords in any case you wish; SELECT is the same as Select which is the same as select.
- **Placeholders** Items that must be replaced with appropriate identifiers or expressions are shown in *italics*.
- **Continuation** Lines beginning with ... are a continuation of the statements from the previous line.
- **Repeating items** Lists of repeating items are shown with an element of the list followed by an ellipsis (three dots). One or more list elements are allowed. If more than one is specified, they must be separated by commas.
- **Optional portions** Optional portions of a statement are enclosed by square brackets. For example:

  ```
  RELEASE SAVEPOINT [ savepoint-name ]
  ```

  It indicates that the *savepoint-name* is optional. The square brackets should not be typed.
- **Options** When none or only one of a list of items must be chosen, the items are separated by vertical bars and the list enclosed in square brackets. For example:

  ```
  [ ASC | DESC ]
  ```

  It indicates that you can choose one of ASC, DESC, or neither. The square brackets should not be typed.
- **Alternatives** When precisely one of the options must be chosen, the alternatives are enclosed in curly braces. For example:

  ```
  QUOTES { ON | OFF }
  ```

  It indicates that exactly one of ON or OFF must be provided. The braces should not be typed.

Typographic conventions

Table 2 lists the typographic conventions used in this documentation.

Table 2: Typographic conventions

Item	Description
`Code`	SQL and program code is displayed in a mono-spaced (fixed-width) font.
`User entry`	Text entered by the user is shown in bold serif type.
emphasis	Emphasized words are shown in italic.
file names	File names are shown in italic.
database objects	Names of database objects, such as tables and procedures, are shown in bold, san-serif type in print, and in italic online.

The sample database

Sybase IQ includes a sample database used by many of the examples in the IQ documentation.

The sample database represents a small company. It contains internal information about the company (employees, departments, and financial data), as well as product information (products), sales information (sales orders, customers, and contacts), and financial information (fin_code, fin_data).

The sample database is held in a file named *asiqdemo.db*, located in the directory *$ASDIR/demo* on UNIX systems and *%ASDIR%\demo* on Windows systems.

Accessibility features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Sybase IQ 12.7 and the HTML documentation have been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

For information about accessibility support in the Sybase IQ plug-in for Sybase Central, see “Using accessibility features” in *Introduction to Sybase IQ*. The online help for this product, which you can navigate using a screen reader, also describes accessibility features, including Sybase Central keyboard shortcuts.

Configuring your accessibility tool

You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool and see "Using screen readers" in *Introduction to Sybase IQ*.

For information about how Sybase supports accessibility, see Sybase Accessibility at http://www.sybase.com/accessibility. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

For a Section 508 compliance statement for Sybase IQ, go to Sybase Accessibility at http://www.sybase.com/products/accessibility.

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

CHAPTER 1 Troubleshooting Hints

About this chapter

This chapter offers suggestions for resolving various problems you may occasionally encounter in running Sybase IQ.

Contents

Topic	Page
Solutions for specific conditions	2
Troubleshooting network communications	32
Diagnostic tools	38
Reporting problems to Technical Support	44
Checklist: information for Technical Support	50

For information on resolving issues related specifically to Sybase IQ multiplex servers, see the section “Multiplex server migration and failover” in Chapter 14, “Data Backup, Recovery, and Archiving” of the *Sybase IQ System Administration Guide*.

If you are unable to resolve the problem using the methods described here, you may find additional help from the Sybase online support Web site, MySybase. MySybase lets you search through closed support cases, latest software bulletins, and resolved and known problems, using a view customized for your needs. You can even open a Technical Support case online. (See the section “Reporting problems to Technical Support” on page 44 for a list of the information to collect before opening a technical support case.)

MySybase can be used from most Internet browsers. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/ and click MySybase for information on how to sign up for and use this free service.

Solutions for specific conditions

This section describes types of conditions that may occur, where to get more information to diagnose the problem, and actions to try to resolve the problem. The issues described in this section are grouped in the following categories:

- “Server recovery and database repair”
- “Server operational issues”
- “Database connection issues”
- “Multiplex issues”
- “dbisql/dbisqlc issues”
- “Resource issues”
- “Processing issues”
- “Performance issues”
- “Sybase Central issues”

See the section “Diagnostic tools” on page 38 for instructions on how to obtain information you can use in diagnosing various conditions, including those described in the following sections.

Server recovery and database repair

If you have trouble starting a server or database, if the database starts but you are unable to connect to it, or if problems are found during database verification, this section helps you determine the action you should take to resolve the problem.

❖ Decision flow for server recovery and database repair

1. Does the server start?

   If the server starts, go to step 2.

   If the server does not start, refer to the section “Server operational issues” on page 3. If you cannot start the server after following the suggestions in this section, then refer to the section “Starting servers in forced recovery mode” on page 74 and start the server in forced recovery mode.

   If the server does not start in forced recovery mode, call Technical Support. A restore of the database from backup may be necessary.

2. Can you connect to the database?

   If you cannot connect to the database, refer to the section "Database connection issues" on page 13 for troubleshooting suggestions.

   If you can connect to the database and you previously started the server with forced recovery, refer to the section "Analyzing allocation problems" on page 69 for information on verifying database allocation and recovering leaked blocks.

   If you can connect to the database, but suspect the database may be inconsistent, refer to the section "Database verification" on page 52 for information on checking the consistency of your database.

3. The server is running and you can connect, but you want to verify the consistency of your database.

   If you previously started the server with forced recovery or you suspect database inconsistency, you should run DBCC checks to validate the database. Refer to the section "Database verification" on page 52 for information on checking both index consistency and database allocation.

4. The server is running, you can connect, you have run DBCC checks, and you need to repair the index inconsistencies or allocation problems detected by DBCC.

   If sp_iqcheckdb reports errors in the Index Summary and Index Statistics sections of the results, refer to the section "Repairing index errors" on page 66 for the procedure to repair index problems using DBCC.

   If sp_iqcheckdb reports errors in the Allocation Summary and Allocation Statistics sections of the results, refer to the section "Repairing allocation problems" on page 71 for the procedure to repair allocation problems using DBCC.

Server operational issues

This section contains information about problems with the operation of the server, including startup, shutdown, unresponsiveness, and abnormal termination.

Sybase IQ will not start

If there is a problem starting the server, start_asiq returns a non-zero value. If you did not specify a log file after the -o switch on startup, the error is written to the first one of the following that is defined:

- *$ASLOGDIR/<servername>.xxxx.srvlog*
- *$ASDIR/logfiles/<servername>.xxxx.srvlog*
- *$ASLOGDIR/start_asiq.log*
- *$ASDIR/logfiles/start_asiq.log*
- the Systems applications log file

Possible causes

- Transaction log file does not match the database.
- Server cannot find the transaction log.
- Operating system is not at proper patch level.
- Network connections are not working.
- Server name is not unique on your network.
- Server port number is not unique on the machine.
- Server is already running as a Windows service (Windows systems only).
- Not enough available memory.
- Environment variables are not set correctly.
- You cannot run start_asiq.

Action

Transaction log file does not match the database The following messages appear in the server log file (*.srvlog*) and in the window where you are starting the server:

```
Starting database "dbname" (/dbdir/dbname.db) at Fri Jun 27 2006 10:53
Transaction log: dbname.log
Error: Cannot open transaction log file -- Can't use log file "dbname.log"
since the database file has been used more recently
Cannot open transaction log file -- Can't use log file "dbname.log" since
the database file has been used more recently
Database server stopped at Fri Jun 27 2006 10:53
```

If these errors are reported when you are starting the server, check to be sure the server is using the correct transaction log file. If you cannot find the correct transaction log file, the safest way to recover from this situation is to restore from the last valid backup.

If you cannot find the correct transaction log and restoring from backup is not an option, then use the emergency recovery method described in “Emergency recovery without a transaction log” on page 79.

Server cannot find the transaction log If the server fails to start because it cannot find the transaction log, the following messages appear in the server log file:

```
Transaction log: /dbdir/dbname.log...
Error: Cannot open transaction log file -- No such file or directory
Cannot open transaction log file -- No such file or directory
```

If this error is reported when you attempt to start the server, find the transaction log file and copy the file to the same directory as the database *.db* file. If you cannot find the correct transaction log file, then restore from the last valid backup.

If no other option for starting the server is available, you may be able to start the server using the method discussed in “Emergency recovery without a transaction log” on page 79. Contact Sybase Technical Support for assistance, if necessary.

Warning! This procedure is highly risky and is not recommended except in extreme cases.

Server name is not unique on your network If the server name is not unique on your network, i.e., multiple systems have a server with the same name, the following messages appear in the server log file (**.srvlog* or the name specified in the -o startup option) when you attempt to start the server using start_asiq:

```
DBSPAWN ERROR:  -82
Unable to start specified database

Server failed to start
```

If you see these errors in the server log file and the server will not start, try to start the server using the asiqsrv12 command. The asiqsrv12 command returns a more specific error message:

```
A database server with that name has already started
```

Once you have verified that the problem is a duplicate server name on your network, start the server with a name that is different from the names of servers that are already running.

Server port number is not unique on the machine If a Sybase IQ server is running and you attempt to start another Sybase IQ server on the same machine using the same port number, the following messages appear in the server log file (**.srvlog*):

```
Trying to start TCPIP link ...
TCPIP communication link not started
Unable to initialize requested communication links
...
DBSPAWN ERROR:  -85
Communication error

Server failed to start
```

If you see these messages in the server log file and the server will not start, run the stop_asiq command to display the names and port numbers of Sybase IQ servers already running on the machine. Then try to start your server, specifying either a port number that is not in use or no port number. When you start a server and do not provide a port number (and the default port number is already in use), Sybase IQ generates an available port number.

Here are the messages you see in the server log file, when you start the server and do not specify a port number:

```
Trying to start TCPIP link ...
Unable to start on default port; starting on port
49152 instead
TCPIP link started successfully
Now accepting requests
...
Server started successfully
```

Not enough memory on Windows If the Sybase IQ server will not start on a 32-bit Windows system, make sure you have enabled the Microsoft 4GT RAM Tuning feature, if appropriate for your version of Windows server. The 4GT option configures the Windows operating system at boot time to allow the allocation of up to 3GB of dynamic memory for a user process. See “System requirements” in the chapter “Installing Sybase IQ” in the *Sybase IQ Installation and Configuration Guide for Windows* for a list of supported Windows platforms and details on enabling the 4GT feature.

Environment variables are not set correctly If your database configuration file parameters differ from those used by start_asiq, make sure the correct parameters are used to start the server. See "Setting environment variables" in Chapter 1, "File Locations and Installation Settings," in the *Sybase IQ Reference Manual*.

You cannot run start_asiq If you cannot run the start_asiq command and you normally use a configuration file or other command line switches, try starting the server using only start_asiq with the server name and database name. If the server starts with this simple command, then the problem is probably caused by one or more of the switches or parameters entered on the command line or in the configuration file. Try to isolate which parameter or switch is preventing the server from starting.

If the server does not start with the most basic start_asiq command, try starting the asiqdemo demo database using your configuration file and command line switches. If the server starts with the asiqdemo database, there may be a problem with your database. Refer to the section "Database connection issues" on page 13.

If you still cannot run the start_asiq command, use the Start Database Server utility in Sybase Central or the asiqsrv12 command.

Before running asiqsrv12, you must perform the following tasks (which start_asiq normally does for you):

- Remove all limits, and then set limits on the stack size and descriptors. To do so, go to the C shell and issue these commands:

  ```
  % unlimit
  % limit stacksize 8192
  % limit descriptors 4096
  ```

  Note Be aware that unlimit affects soft limits only. You must change any hard limits by setting kernel parameters.

- Be careful to set all server options appropriately for your platform. For details about appropriate options and how to set them in a configuration file, see the *Sybase IQ Installation and Configuration Guide*.

For any database created with a relative pathname, you must start the database server from the directory where the database is located.

Note what directory you are in when you start the server. The server startup directory determines the location of any new database files you create with relative pathnames. If you start the server in a different directory, Sybase IQ cannot find those database files.

Any server startup scripts should change directory to a known location before issuing the server startup command.

Syntax for asiqsrv12 is as follows:

> **asiqsrv12 -n** *server-name* **-gm** *number*
> [*other-server-switches*] [*database-file* [*database-switches*]]

Note On the asiqsrv12 command line, the last option specified takes precedence, so if you want to override your configuration file, list any options you want to change *after* the configuration file name. For example:

```
asiqsrv12 @asiqdemo.cfg -x 'tcpip{port=1870}' asiqdemo
```

The –x parameter here overrides connection information in the *asiqdemo.cfg* file.

When you start the server with the asiqsrv12 command, it does not run in the background, and messages do not automatically go to the server log. However, if you include the -o filename server switch, messages are sent to the named file in addition to the server window.

If the server fails to start when you run the asiqsrv12 command, then attempt to start again using the asiqsrv12 utility with minimal switches and parameters. For example:

```
asiqsrv12 -n <servername> <dbname>.db -c 32m
-gd all -gl all
```

If the server starts with the minimum parameters and switches, then one of the parameters or switches normally used to start the server may be causing a problem. Try to isolate which parameter or switch is preventing the server from starting.

See also

Chapter 2, “Running Sybase IQ” and Chapter 3, “Sybase IQ Connections” in the *Sybase IQ System Administration Guide* for more information on server startup, including the section “Troubleshooting startup, shutdown, and connections.”

Sybase IQ stops processing or stops responding

Possible causes

The following are the two most common causes of server unresponsiveness, which can be detected by looking in the Sybase IQ message file:

- Insufficient disk space. See the section “Insufficient disk space” on page 19 for actions to take.
- Insufficient room in main or temp buffer cache. See “Managing buffer caches” in Chapter 5, “Managing System Resources” of the *Sybase IQ Performance and Tuning Guide*.

Action

If your server seems to be prone to unresponsiveness, either while processing or during shutdown, use the start_asiq command line option -z and the Sybase IQ database option QUERY_PLAN = 'ON' to log useful information in the Sybase IQ message (*.iqmsg*) and server log (*.srvlog*) files. In addition to logging this information, there are other steps you can take to determine the cause of the problem:

- Check both the Sybase IQ message file and the server log file for `"You have run out of … dbspace"` messages. If you have run out of IQ STORE (main) or IQ TEMPORARY STORE, add the appropriate dbspace with the CREATE DBSPACE command. See the section “Insufficient disk space” on page 19 for more information on resolving out of space issues.

  Setting the database options MAIN_RESERVED_DBSPACE_MB and TEMP_RESERVED_DB_SPACE_MB to large enough values to handle running out of space during a DDL COMMIT or CHECKPOINT is also important. A few hundred MB should be enough, but these options can be set higher for a large database. For more information, see the section “Reserving space to handle out-of-space conditions” in Chapter 5, “Working with Database Objects” of the *Sybase IQ System Administration Guide*.

- Determine if the Sybase IQ server process (asiqsrv12) is consuming CPU cycles by monitoring the CPU usage for a few minutes at the operating system level. Record this information. If the CPU usage changes, then the Sybase IQ server process should be processing normally.

  If the Sybase IQ server CPU usage is normal, you can examine what the server is doing, i.e., what statement the server is currently executing. For details on capturing this information and logging server requests, see the sections “Finding the currently executing statement” on page 41 and “Logging server requests” on page 41.

- If there are no out of space indications, use dbisql on a new or existing connection to gather the information listed in the following table (in this order).

  Table 1-1: Information to gather for server unresponsiveness

Command	Informational purpose
select db_name()	database name
checkpoint	checkpoint can succeed
select db_property('FileVersion')	version of catalog
set temporary option truncation_length=100	increase output line length
sa_conn_properties >># sa_conn_properties.out	connection information
sa_conn_info >># sa_conn_info.out	connection information
sa_db_properties >># sa_db_properties.out	database property information
sa_eng_properties >># sa_eng_properties.out	server property information
sp_iqstatus >># sp_iqstatus.out	database status information
sp_iqconnection >># sp_iqconnection.out	connection information
sp_iqtransaction >># sp_iqtransaction.out	transaction information

  If you cannot resolve this issue, contact Sybase Technical Support for assistance. The information you have just gathered can be used by Technical Support to help diagnose the problem. See the section “Reporting problems to Technical Support” on page 44.

- When the server is unresponsive, you can generate a stack trace for each Sybase IQ thread by creating a file named *DumpAllThreads* or *dumpallthreads* in the *$ASDIR/logfiles* directory (the *%ASDIR%\logfiles* folder on Windows platforms).

  Starting Sybase IQ as recommended, using the Program Manager or start_asiq command, sets the ASDIR variable automatically. If the ASDIR variable is not set, create the *DumpAllThreads* file in the directory in which *asiqsrv12* was started.

  The Sybase IQ server detects the presence of the *DumpAllThreads* file and writes a stack trace for each IQ thread in the stack trace file *stktrc-YYYYMMDD-HHNNSS_#.iq*. After the stack traces are written to the stack trace file, the *DumpAllThreads* file is deleted.

  This stack trace information can be used by Sybase Technical Support to help diagnose the problem. See the section “Reporting problems to Technical Support” on page 44.

- If you can connect to the database, run the IQ UTILITIES buffer cache monitor on the main and temp (private) buffer caches for 10 minutes with a 10 second interval:

  a Connect to the database or use the existing connection.

  b
  ```
  CREATE TABLE #dummy_monitor(c1 INT);
  ```

  c
  ```
  IQ UTILITIES MAIN INTO #dummy_monitor START MONITOR
  '-append -debug -interval 10 -file_suffix iqdbgmon';
  ```

  d
  ```
  IQ UTILITIES PRIVATE INTO #dummy_monitor
  START MONITOR '-append -debug -interval 10
  -file_suffix iqdbgmon';
  ```

  Let the process run for 10 minutes, then stop the buffer cache monitor:

  e
  ```
  IQ UTILITIES MAIN INTO #dummy_monitor STOP MONITOR;
  ```

  f
  ```
  IQ UTILITIES PRIVATE INTO #dummy_monitor STOP
  MONITOR;
  ```

  For more information on monitoring buffer caches, see the section “Monitoring the buffer caches” in Chapter 6, “Monitoring and Tuning Performance” of the *Sybase IQ Performance and Tuning Guide*.

- Check near the end of the Sybase IQ message file for the message `"Resource count 0"`, which may be followed by an `"Open Cursor"` message. These messages indicate a resource depletion, which can cause a deadlock. The immediate solution is to reduce the number of active connections using CTRL-C or the DROP CONNECTION command.

  The long term solution to avoid a deadlock due to resource depletion is one or a combination of the following:

  - Restrict the number of users on the server by reducing the value of the -gm server startup option
  - Add another query server to a multiplex
  - Increase the processing capacity of the hardware by adding CPUs

Server fails to shut down

Normally you should be able to shut down the server by running the dbstop utility or stop_asiq, by typing `q` in the server window on UNIX, or by clicking Shutdown on the server window on Windows. If none of these methods works, see the Actions section below.

Possible causes

Various.

Actions

On UNIX systems:

1 Capture ps operating system utility output, so you can submit this output to Sybase Technical Support. On Sun Solaris two different ps options are available. Use both.

```
ps -aAdeflcj|egrep "PPID|asiqsrv12"

/usr/ucb/ps -awwwlx|egrep "PPID|asiqsrv12"
```

2 Try to kill the process at the operating system level to generate a core dump.

```
kill -6 pid
```

A small core file is created in the directory where start_asiq was run. If you are able to kill the server process in this way, skip to step 5.

3 If the server process still does not exit, capture ps output as in step 1. Retain the output from both times you run ps (before and after trying to kill the process). Then kill the process with a stronger signal:

```
kill -9 pid
```

4 If this method does not cause the process to exit, capture yet another set of ps output, and then reboot your system.

5 Submit all ps output, the core file (if generated in step 2), and the stack trace in *stktrc-YYYYMMDD-HHNNSS_#.iq* to Sybase Technical Support.

On Windows systems:

1 Start the Task Manager by right-clicking the Task Bar and clicking Task Manager.

2 In the Processes tab, select asiqsrv12.exe and then click the End Process button to stop the database server.

3 If necessary, restart Windows.

Refer to the section “Reporting problems to Technical Support” on page 44 for a full list of information to provide to Sybase Technical Support.

System failure/Sybase IQ failure

Possible causes

Various.

Actions

- Copy or rename the message log file (*dbname.iqmsg*) before trying to restart the database. This ensures that any useful information in the file will not be lost.
- On UNIX, send a copy of the stack trace to Sybase Technical Support, along with the additional information listed in the section “Reporting problems to Technical Support” on page 44. The stack trace should be in the directory where you started the database server, in a file named *stktrc-YYYYMMDD-HHNNSS_#.iq*. If the database was open when the failure occurred, the stack trace should also be in the Sybase IQ message log (default name *dbname.iqmsg*). This information helps Sybase Technical Support determine why the failure occurred.
- Restart the server with the start_asiq command. When the database restarts, recovery occurs automatically.
- Try to start the server without starting a database. If you are able to start the server but not the database, check that database parameters are specified correctly on the startup line and/or in the connection profile.
- If you query Catalog Store tables extensively, restart the server and make sure that the TEMP_SPACE_LIMIT_CHECK option is ON. With this option setting, if a connection exceeds its quota of Catalog Store temporary file space it receives a non-fatal error.

See also

- Chapter 2, “System Recovery and Database Repair”
- “System recovery” in Chapter 10, “Transactions and Versioning” of the *Sybase IQ System Administration Guide*

Database connection issues

This section contains information on issues you may encounter when attempting to connect to a database.

Cannot connect to a database

Possible causes

- Data source is not defined, or you have entered or defined it incorrectly. A data source is a set of connection parameters, stored in the registry (on Windows) or in a file (Windows and UNIX).

- An incorrect user name or password is specified. The error messages returned are:

  ```
  Unable to connect
  ```

  or

  ```
  Could not connect to the database.
  ```

  followed by the message:

  ```
  Invalid user ID or password.
  ```

  Try connecting again with the correct user ID and password.

- User may not have permission to use the database.
- You provide an incorrect database file name. The error messages returned are:

  ```
  Unable to connect
  ```

  or

  ```
  Could not connect to the database.
  ```

  followed by the message:

  ```
  Specified database not found.
  ```

  Try connecting again with the correct database file name.

  You must supply the DBF parameter and the database file name to connect when you use dbisqlc or dbisql and you have restored the database from backup while connected to *utility_db*. For details, see “Reconnecting after you restore” in *Sybase IQ System Administration Guide*.

- Database files may be missing. The files *dbname.db*, *dbname.iq*, *dbname.iqmsg*, and *dbname.iqtmp* (where dbname is the name of the database) must all exist.
- A limit on the number of connections or other DBA-defined login restrictions may be exceeded. The error messages returned are:

  ```
  Unable to connect
  Database server connection limit exceeded.
  ```

- You have run out of disk space. Check the Sybase IQ message file for messages related to disk space.
- The server name specified is not correct. The error messages returned are:

  ```
  Connection failed.
  Database server not running.
  ```

Check the name of the server and try connecting again with the correct server name.

- The server machine name or address has changed.
- When connecting from a client for the first time and the server name is not specified, providing the wrong port number can cause a failure to connect to the database. The error messages returned are:

  ```
  Connection failed.
  Database server not running.
  ```

  Either provide the server name when connecting, or use the correct port number. To determine the server name and the number of the port on which the server is listening, run the command stop_asiq, which displays this information.
- Port number may be out of correct range or in use by another process.
- If you receive the message

  ```
  Unable to start — server not found
  ```

  or

  ```
  Database server not running.
  ```

  when trying to start the client, the client cannot find the database server on the network. The connection string may be incorrect or the server name cache may contain incorrect or old connection information. For example, if the server is started with a different port number, even if the client application specifies the new port number at connect time, the connection information is still taken from the server name cache.

  The server name cache is a file named *asasrv.ini* in the Sybase IQ executable directory that contains server connection information and is used for faster connections over a network. You can modify the server name cache to contain the new port number or you can completely remove the server entry from the file. For more information on the server name cache, see the section “Server name caching for faster connections” in Chapter 3, “Sybase IQ Connections” of the *Sybase IQ System Administration Guide*.
- You used Component Integration Services (CIS) in certain geographic regions, where connection attempts return the error No Suitable Driver. Java Developer Kits used with Sybase IQ 12.7 support time zone codes shown in Table 1-1 and Table 1-2 in Chapter 1, “File Locations and Installation Settings,” in the *Sybase IQ Reference Manual*.

  - For databases using default JDK 1.1.8:

    Substitute JST for unsupported time zone KST, which gives the same GMT+9 time, as follows:

    ```
    setenv TZ JST
    ```

  - For databases using JDK 1.3:

    ```
    setenv TZ Asia/Seoul
    ```

  Set the time zone environment variable to a supported setting, start the server, and CIS works as expected. To ensure that the correct setting is always used, you can set the time zone in the start_asiq script.

- You specified a character set in the CharSet connection parameter and tried to connect to a server that does not support that character set. If the server does not support the requested character set, the connection fails.

  Try reconnecting without specifying CharSet. If the client's local character set is unsupported by the server, the connection succeeds, but with a warning that the character set is not supported.

Note Do not confuse an inability to connect to a database with a Sybase IQ server-level error while Sybase IQ is trying to open a database.

Action

If you suspect that you cannot connect because there is a problem with the database, you can look in the *dbname.iqmsg* file to determine where the problem occurred. If the message “Open Database Completed” appears, then the database opened without error and the problem is related to the clients connecting. If the message does not appear, then the database may have failed while opening or recovering.

See also

- Chapter 3, “Sybase IQ Connections” in the *Sybase IQ System Administration Guide* for more information on creating and editing data sources, how Sybase IQ makes connections, specifying a port number, and troubleshooting database connection problems.
- Chapter 12, “Managing User IDs and Permissions” in the *Sybase IQ System Administration Guide* for information on database permissions.
- “LOGIN_PROCEDURE option” on page 105 in the *Sybase IQ Reference Manual*
- “Insufficient disk space” on page 19
- Chapter 2, “System Recovery and Database Repair”

- "TZ environment variable" in Chapter 1, "File Locations and Installation Settings," in the *Sybase IQ Reference Manual*.

Multiplex issues

This section contains information on troubleshooting issues related to multiplex functionality.

Main dbspace count differs on write and query servers

Possible causes

Sybase Central may fail executing a CREATE DBSPACE command on a query server, causing the main dbspace count from a query server to differ from the count on the write server.

Trying to start the query server where the CREATE DBSPACE failed returns the error:

```
Query Server dbspace count differs from Write Server
```

In this case, executing sp_iqmpxvalidate on the write server returns a message like the following:

```
ERROR: Main dbspace counts for server UXIQIP393, write
server mismatch in SYS.SYSIQFILE
```

CREATE DBSPACE may also fail on a query server if the file system permissions for a shared raw disk device are wrong at the query server, or if the pathname does not resolve to the same device where the write server created the dbspace.

Actions

The following actions resolve the problem:

- Use Sybase Central to drop the query server and add it to the multiplex again, or
- Use the stored procedure sp_iqmpxaliasdbspace on the write server interactively to create the necessary dbspace definitions for the query server, and then synchronize the query server.

See also

- Chapter 10, "System Procedures" in *Sybase IQ Reference Manual*
- "Adding dbspaces" on page 221 in *Sybase IQ System Administration Guide*

Multiplex synchronize “Not enough space” error

A multiplex synchronize may fail with a “Not enough space” error in the Agent log file:

```
java.io.IOException: Not enough space
```

This error indicates an out of memory condition, not a physical out of disk space or dbspace situation.

dbisql/dbisqlc issues

This section contains information on troubleshooting issues related to the operation of dbisql and dbisqlc.

Data truncation or data conversion error

Possible causes

A data truncation error or conversion error occurs when a procedure calls another procedure with a dynamic result set and all of the following are true:

- The Sybase IQ server is version 12.5
- dbisql Java connects through iAnywhere JDBC driver
- dbisql Java version is higher than 7.04.

The problem does not happen if dbisql Java connects through jConnect5 or the ODBC driver or if Sybase IQ 12.6 is used with dbisql 9.0.1.

- Differences in display characteristics between your terminal and the expectations of Sybase IQ.
- Differences in function key support between your terminal and the expectations of Sybase IQ.

Action

There are several ways to avoid the problem:

- Connect dbisql Java through jConnect5 or ODBC driver.
- Use Sybase IQ 12.7 with dbisql version 9.0.2, Sybase IQ 12.6 with dbisql version 9.0.1, or Sybase IQ 12.5 with dbisql version 7.0.4
- Add a statement like the following to the start of the procedure, to keep the server from adding a result set:

  ```
  IF 1 = 0 THEN
  SELECT 1 AS a FROM nosuchtable;
  END IF;
  ```

dbisqlc window does not work on UNIX

Possible causes

- Differences in display characteristics between your terminal and the expectations of Sybase IQ.
- Differences in function key support between your terminal and the expectations of Sybase IQ.

Action

Install, and if necessary edit, the terminfo extension (.*tix*) file provided with Sybase IQ. This file contains the definitions of function keys and special key sequences. See the section “Connecting to databases from DBISQL” in Chapter 3, “Running and Connecting to Servers” of the manual *Introduction to Sybase IQ* for more information on installing the terminfo extension file.

Directories remain after exiting dbisql

Note This issue affects users of NFS file systems only.

Possible causes

The ASTMP environment variable is not set to point to a local directory.

Each client connection creates several directories and files in a temporary directory. Sybase IQ deletes these files when the connection ends. If ASTMP does not point to a local directory, it cannot find the .*nfs** files that NFS creates.

Action

Set ASTMP to a local directory and restart the server.

Resource issues

This section contains information on troubleshooting resource issues, including insufficient disk space, insufficient number of threads, thread stack overflow, and unused system resources.

Insufficient disk space

Warning! If Sybase IQ holds certain system locks or is performing a checkpoint when you run out of disk space, you may not be able to add disk space. For this reason, recognizing when you are low on disk space and adding a new dbspace *before* you run out of space are important.

For an example of using an event handler to monitor disk space usage, see the section “Monitoring disk space usage” on page 23.

Actions

- Check recent messages in the Sybase IQ message log (*dbname.iqmsg*). If you see an “out of dbspace” message, you must add another dbspace. The message in the Sybase IQ message file indicates which dbspace has run out of space and the minimum number of megabytes to add. If the problem occurs while you are inserting data, you probably need more room in the IQ Store. If the problem occurs during queries with large sort-merges joins, you probably need more room in the Temporary Store.

  Check the Sybase IQ message log for the following message:

```
You have run out of { IQ STORE | IQ TEMPORARY STORE } dbspace
in database <dbname>. In another session, please issue a
CREATE DBSPACE ... { IQ STORE | IQ TEMPORARY STORE } command and
add a dbspace of at least nn MB.
[EMSG_IQSTORE_OUTOFDISK_HEADER: 'QSB31',1009131].
```

- Try to connect to the database from a new connection. If this works, you know that the database server is running, even though the query is waiting. Run sp_iqstatus to get more information.
- If you cannot connect to the database, check if Sybase IQ is in an unusable state by monitoring the CPU usage for that processor. If the CPU usage does not change over a small time interval, then Sybase IQ is probably not operational. If the CPU usage does change, Sybase IQ is operational.
- Check the sp_iqstatus output for the following two lines:

```
Main IQ Blocks Used:,10188 of 12288, 82%, Max Block#: 134840
Temporary IQ Blocks Used:,163 of 6144, 2%, Max Block#: 97
```

  If the percentage of blocks used is in the nineties, you need to add more disk space with the CREATE DBSPACE command. In this example, 82% of the Main IQ Blocks and 2% of the Temporary IQ Blocks are used, so more space will soon be needed in the Main IQ Store.

- If out-of-space conditions occur or sp_iqstatus shows a high percentage of main blocks in use on a multiplex server, run sp_iqversionuse to find out which versions are being used and the amount of space that can be recovered by releasing versions. For details, see “sp_iqversionuse procedure,” *Sybase IQ Reference Manual*.

Running out of space during checkpoint

If you run out of space during a checkpoint:

- For a multiplex server, try starting in single-node mode.

  Start the write server in single-node mode using the -iqmpx_sn 1 switch. *Note that if you use a server name different from the current write server name, you must also use the override switch,* -iqmpx_ov 1.

  For example:

  ```
  start_asiq @params.cfg -n <write_server>
  -iqmpx_sn 1 -x 'tcpip{port=<writer_port>}' <dbfile>
  ```

  You may use the write server's normal TCPIP port.

- For a non-multiplex server, try starting in forced recovery mode. See "Starting servers in forced recovery mode" on page 74.

Add a dbspace as soon as possible. You must add a dbspace before any new checkpoints can succeed.

Effect of checkpoints on out of disk space conditions

If Sybase IQ has already run out of space when a checkpoint is requested, the checkpoint command fails with the error:

```
You have run out of space during the CHECKPOINT
operation.
[EMSG_IQSTORE_OUTOFSPACE_CHECKPOINT:'QSB33', 1009133].
```

You must add a dbspace before any new checkpoints can succeed.

Adding space if you cannot connect to a server

If you run out of space during an operation and are unable to add space because you cannot connect to the server, you must:

1. Shut down the database server using any of these methods:

   - On any platform, run dbstop.
   - On Windows, click the correct server icon on the Windows task bar to display the Sybase IQ window, and then click the Shutdown button.
   - On UNIX, run stop_asiq or type `q` in the window where the server was started.

   If the server does not shut down, see "Server fails to shut down" below.

2. Restart the engine with the start_asiq command.
3. Connect to the database.
4. Use the CREATE DBSPACE command to add space.
5. Rerun the operation that originally failed due to insufficient space.

Managing dbspace size

Growth of catalog files is normal and varies depending on application and catalog content. The size of the *.db* file does not affect performance, and free pages within the *.db* file are reused as necessary. To minimize catalog file growth:

- Avoid using IN SYSTEM on CREATE TABLE statements.
- Issue COMMIT statements after running system stored procedures.
- Issue COMMIT statements after long-running transactions

If the Catalog Store cannot extend one of its files (*.tmp*, *.db,* or *.iqmsg*), Sybase IQ returns the error `A dbspace has reached its maximum file size`. To prevent this problem:

- Monitor space usage periodically.
- Verify that there are no operating system file size limits (such as Sun Solaris ulimit) where the *.tmp, .db,* or *.iqmsg* files are located. The *.db* and *.tmp* files are typically in the main Sybase IQ database directory. The *.tmp* file is located under *$ASTMP/<servername>/tmp*, or if $ASTMP is not set, under */tmp/.SQLAnywhere/<servername>/tmp*.

Adding the wrong type of space

If the temporary dbspace runs out of space and you accidentally omit the temporary keyword in the create dbspace command, the create dbspace command waits for you to add space to the temporary dbspace.

To provide the temporary dbspace needed by the create dbspace command, connect to the database from a new connection and create the temporary dbspace. Once the temporary dbspace is created, the create dbspace for main completes and all waiting connections resume running.

The same is true, if the main dbspace runs out of space and you attempt to add temporary dbspace. Sybase IQ waits for you to add space to main first, before adding to the temporary dbspace. You must add space to the dbspace that runs out of space first, and Sybase IQ ensures this sequence.

Fragmentation

Sybase IQ provides control over fragmentation by taking advantage of even the smallest unused spaces. However, fragmentation can still occur. If your database runs out of space, even though Mem Usage listed by sp_iqstatus or the *.iqmsg* file shows Main IQ Blocks Used is less than 100%, it usually indicates that your database is fragmented,

Freeing space

Note that when a connection is out of space, freeing space by dropping tables or indexes in another connection is not possible, because the out of space transaction will see those objects in its snapshot version.

Monitoring disk space usage

Recognizing when the server is low on disk space and adding a new dbspace *before* the server runs out of space is important. See the section “Monitoring disk space usage” on page 23 for an example of using an event handler to monitor disk space usage and to notify you when available space is low.

Reserving space for the future

To ensure that you have enough room to add new dbspaces if you run out of space in the future, set the database options MAIN_RESERVED_DBSPACE_MB and TEMP_RESERVED_DBSPACE_MB. Set these options large enough to handle running out of space during a COMMIT or CHECKPOINT. See Chapter 2, “Database Options” in the *Sybase IQ Reference Manual* for details.

Monitoring disk space usage

You can use an event handler to monitor disk space usage and notify you when available space is running low. The example in this section is especially useful for monitoring space during loads. You can enable the event handler before you start the load and disable the event handler after the load completes.

The following is sample event handler code. You can modify this code to perform other types of monitoring.

```
-- This event handler sends email to the database
-- administrator whenever the IQ Main DBSpace is more than
-- 95 percent full.

-- This event handler runs every minute. The event handler uses
-- sp_iqspaceused to sample the space usage. If the space is
-- more than 95 percent full, a file that contains the date and
-- time is created in the directory where asiqsrv12 is
-- running. The file contents are then mailed to the database
-- administrator and the file is removed.
-- This event can be enabled before a load and be used
-- to monitor disk space usage during loading. The event can
-- then be disabled after the load.

create event out_of_space
schedule
start time '1:00AM' every 1 minutes
handler

begin
declare mt unsigned bigint;
declare mu unsigned bigint;
declare tt unsigned bigint;
declare tu unsigned bigint;
```

```
call sp_iqspaceused(mt, mu, tt, tu);

if mu*100/mt  > 95  then
  call xp_cmdshell('date > ./temp_m_file');
  call xp_cmdshell('mailx -s add_main_dbspace iqdba@iqdemo.com
    < ./temp_m_file');
  call xp_cmdshell('/bin/rm -rf ./temp_m_file');
end if

if tu*100/tt  > 95  then
  call xp_cmdshell('date > ./temp_file');
  call xp_cmdshell('mailx -s add_temp_dbspace iqdba@iqdemo.com
    < ./temp_file');
  call xp_cmdshell('/bin/rm -rf ./temp_file');
end if

end
```

For more information on using events, see Chapter 18, "Automating Tasks Using Schedules and Events" in the *Sybase IQ System Administration Guide*. For details on the SQL statements that create, modify, and control events, see "CREATE EVENT statement", "ALTER EVENT statement", and "TRIGGER EVENT statement" in Chapter 6, "SQL Statements" of the *Sybase IQ Reference Manual*.

Insufficient threads

Possible cause

The query you have issued requires more kernel threads for the IQ Store.

Actions

- Wait for another query to finish and release the threads it is using. Then resubmit your query.
- Run the system stored procedure sp_iqconnection. The column IQThreads contains the number of IQ threads currently assigned to the connection. This column can help you determine which connections are using the most resources. Remember that some threads may be assigned but idle.
- If the condition persists, you may need to restart the server and specify more IQ threads. Use the -iqmt server startup switch to increase the number of processing threads that Sybase IQ can use. The default value of -iqmt is `60*numCPU+2*num_conn+1`. The total number of threads (-iqmt plus -gn) must not exceed 4096 on 64-bit platforms, 1000 on IBM AIX 32-bit servers, or 2048 on all other 32-bit platforms. This option is set automatically to 450 by the start_asiq startup utility on the IBM AIX platform.

Stack overflow

Possible cause

If you see the error `AbortIfEndofStack` in the stack trace file (*stktrc-YYYYMMDD-HHNNSS_#.iq*), the thread stack has overflowed.

Actions

- To avoid this problem, restart Sybase IQ with the server parameter -iqtss set to 300 on 32-bit operating systems or 500 on 64-bit operating systems. On 32-bit systems, you may need to decrease LOAD_MEMORY_MB in order to increase -iqtss. The server startup switch -iqtss specifies thread stack size in KB. If this is not adequate, raise the value of -iqtss by 72 until the problem is solved.
- If possible, identify the command that caused the error and forward it to Sybase Technical Support.

Unused semaphores and shared memory left after abnormal exit

Possible causes

Killing processes on UNIX systems may result in semaphores or shared memory being left behind instead of being cleaned up automatically. To eliminate unneeded semaphores, you should periodically run the UNIX ipcs command to check the status of semaphores and shared memory.

The ipcs -a command lists the ID numbers, owners, and create times of semaphores and shared memory segments. When all Sybase IQ instances are started by the same user (as Sybase recommends), you can search the OWNER column for that user name. Identify shared memory segments and semaphores that are not being used.

Action

After verifying with the owner that these shared memory segments and semaphores are not in use, run the UNIX ipcrm command to remove them. Use the -m parameter to specify the memory segment ID and the -s command to specify the semaphore ID number, in the following format:

```
ipcrm -m mid1 -m mid2 ... -s sid1 -s sid2 ...
```

For example:

```
% ipcrm -m 40965 -s 5130 -s36682
```

Insufficient procedure identifiers

Sybase IQ assigns internal catalog proc_ids for procedures sequentially and and unused proc_ids are not reused. As procedures are dropped and created, databases created prior to Sybase IQ 12.6 may eventually reach the maximum proc_id limit of 32767, causing CREATE PROCEDURE to return an “Item already exists” error in Sybase IQ 12.6.

For databases created with a version prior to Sybase IQ 12.6 GA, the maximum proc_id for procedures is 32767, even if the database has been upgraded to Sybase IQ 12.6 or higher. This limit does not apply to databases created with Sybase IQ 12.6 and higher.

To determine if your database has a 32767 maximum proc_id limit, run `sp_columns sysprocedure`. If the data type for the proc_id column is smallint than the maximum proc_id of 32767 applies.

To determine the current maximum proc_id value for your database, run the following query:

```
SELECT MAX (proc_id) FROM sys.sysprocedure
```

Sybase IQ 12.6 ESD7 and higher ensures that, for databases created prior to Sybase IQ 12.6, the maximum proc_id is at a level that allows ALTER DATBASE UPGRADE to complete. If the maximum proc_id is higher, ALTER DATABASE UPGRADE fails and returns the message “Database upgrade not possible”.

To resolve this issue for databases created prior to Sybase IQ 12.6, ALTER DATABASE UPGRADE supports a PROCEDURE ON clause in 12.6 ESD7 and higher that compacts the proc_ids by recreating all stored procedures. The syntax is ALTER DATABASE UPGRADE PROCEDURE ON. The PROCEDURE ON clause is ignored for databases created in 12.6 and higher.

ALTER DATABASE UPGRADE PROCEDURE ON recreates all procedures without comments. If you want the comments back in the procedures after running the command, run ALTER PROCEDURE <procedure_name> with your source code for the procedures that contain comments. The sp_helptext <owner>.<procname> command can be used to save the text of procedures with comments before running ALTER DATABASE UPGRADE PROCEDURE ON.

As a backup, copy the *.db* and *.log* files for the database immediately before running ALTER DATABASE UPGRADE PROCEDURE ON. Since only the catalog is modified during an ALTER DATABASE UPGRADE command, a full backup is unnecessary.

Processing issues

This section contains information on troubleshooting processing issues related to loads, queries, indexes, and table access.

For information on monitoring disk space usage during loads, see “Monitoring disk space usage” on page 23.

Too many indexes on table

Possible cause

A Microsoft Access user is trying to link to a table that has more than 32 indexes.

Action

Create a view that selects all the columns in the table, and link to the view instead of the base table.

See also

Chapter 8, “Using Views” in *Introduction to Sybase IQ*.

Unexpectedly long loads or queries

Possible causes

- IQ buffer cache is too large, so the operating system is thrashing.
- IQ buffer cache is too small, so Sybase IQ is thrashing because it cannot fit enough of the query data into the cache.
- You attempted to set IQ buffer cache sizes so that total memory requirements on your system exceed total system memory. The buffer caches were therefore automatically reduced to their default sizes.
- User defined functions or cross database joins requiring CIS intervention.
- Missing HG or LF index on columns used in the WHERE clause and GROUP BY clause.

Action

Monitor paging to determine if thrashing is a problem.

- To monitor IQ paging, run the IQ buffer cache monitor, as described in the *Sybase IQ Performance and Tuning Guide* section “Monitoring the buffer caches” in Chapter 6, “Monitoring and Tuning Performance.”
- To monitor operating system paging, use the UNIX vmstat utility or other platform specific tools, or the Windows Performance Monitor.

Reset your buffer sizes as needed. See the section “Monitoring the buffer caches” in Chapter 6, “Monitoring and Tuning Performance” of the *Sybase IQ Performance and Tuning Guide*.

If you monitor paging and determine that thrashing is a problem, you can also limit the amount of thrashing during the execution of a statement which includes a query that involves hash algorithms. Adjusting the HASH_THRASHING_PERCENT database option controls the percentage of hard disk I/Os allowed before the statement is rolled back and an error is returned.

The default value of HASH_THRASHING_PERCENT is 10%. Increasing HASH_THRASHING_PERCENT permits more paging to disk before a rollback and decreasing HASH_THRASHING_PERCENT permits less paging before a rollback.

Queries involving hash algorithms that executed in earlier versions of Sybase IQ may now be rolled back when the default HASH_THRASHING_PERCENT limit is reached. The error `"Hash insert thrashing detected."` or `"Hash find thrashing detected."` (SQLState QFA43, SQLCode -1001047) is reported. Take one or more of the following actions to provide the query with the resources required for execution:

- Relax the paging restriction by increasing the value of HASH_THRASHING_PERCENT.
- Increase the size of the temporary cache (DBA only). Keep in mind that increasing the size of the temporary cache reduces the size of the main cache.
- Attempt to identify and alleviate why Sybase IQ is misestimating one or more hash sizes for this statement.
- Decrease the value of the database option HASH_PINNABLE_CACHE_PERCENT.

To identify possible problems with a query, generate a query plan by running the query with the temporary database options QUERY_PLAN = 'ON' and QUERY _DETAIL = 'ON', then examine the estimates in the query plan. The option QUERY_PLAN_AFTER_RUN = 'ON' provides additional information, as the query plan is printed after the query has finished running. The generated query plan is in the message log file.

See also

- Chapter 5, “Managing System Resources” in the *Sybase IQ Performance and Tuning Guide*
- Chapter 6, “Monitoring and Tuning Performance” in the *Sybase IQ Performance and Tuning Guide*
- “HASH_THRASHING_PERCENT option” and “HASH_PINNABLE_CACHE_PERCENT option” in Chapter 2, “Database Options” of the *Sybase IQ Reference Manual*

Load fails on number of unique values

Possible cause

The following message in the log file indicates that you have more than 10000 unique values in a column with an LF index:

```
1009103: Number of unique values exceeded for index.
index_name_LF 10000
```

The Low_Fast index is optimized for 1000 unique values, but has an upper limit of 10000.

Action

Replace the LF index with an HG index.

To do this, issue a DROP INDEX statement to drop the LF index identified in the error message. For example:

```
DROP INDEX DBA.employee.emp_lname_LF
```

Then issue a CREATE INDEX statement to create the new HG index. For example:

```
CREATE HG INDEX ON DBA.employee (emp_lname)
```

Cannot write to a locked table

Possible causes

The following error message is reported, when an attempt is made to write to an object to which another user already has write access.

```
Cannot open the requested object for write in the
current transaction (TxnID1). Another user has write
access in transaction TxnID2.
```

Action

Use the sp_iqlocks stored procedure to identify users who are blocking other users from writing to a table. This procedure displays information about locks currently held in the database, including the connection and user ID that holds the lock, the table on which the lock is held, the type of lock, and a name to identify the lock.

The error message also includes the transaction ID of the user who is attempting to write (TxnID1)and the transaction ID of the user who is currently writing (TxnID2). If you need more detailed information about the transaction that has locked the table, run the sp_iqtransaction stored procedure.

See also

Chapter 10, “System Procedures” in the *Sybase IQ Reference Manual* and “Managing write lock contention on a table” on page 30.

Managing write lock contention on a table

High contention for write locks on a table used by multiple users can impact processing, if most of the transactions are able to obtain the lock. The sample stored procedure in this section is an example of a method to manage the contention for a write lock on a table. This procedure does not eliminate the write lock contention on the table, but does manage the contention, so that transactions are able to get the write lock.

The following stored procedure code manages the lock contention on a table named dbo.event that is used to record events. The procedure returns the event_id to the caller. This table is in high contention for write locks. The stored procedure dbo.log_event records information in the table dbo.event. If an access error occurs, the error is captured, the hopeful writer sleeps for a five second interval, and then attempts to write to the table again. The five second re-try interval is usually long enough for the contention to be resolved, so the write lock on the dbo.event table is available.

You can modify this code to perform other similar tasks.

```
if exists (select 1
           from   sys.sysprocedure a
           join   sys.sysuserperm b on a.creator = b.user_id
           where  a.proc_name = 'log_event' and b.user_name = 'dbo') then
   drop procedure dbo.log_event;
end if;

create procedure dbo.log_event(in @event varchar(255))
on exception resume
begin
    declare @event_id   bigint;
    declare @res        char(5);
    set @event_id=0;
    loop1: loop
        commit work;
        select  max(event_id)+1
            into    @event_id
            from    dbo.event;
        insert  dbo.event
            values (@event_id,@event,current timestamp,null,null);
        set @res=sqlstate;
        if @res = ' ' or(@res <> 'QDA29' and @res <> 'QDA11') then
            leave loop1
        end if;
        call dbo.sleep(5);
    end loop loop1;
    commit work;
    return @event_id
end
```

See also

For more information on using stored procedures, see Chapter 8, "Using Procedures and Batches" in the *Sybase IQ System Administration Guide*.

For more information on locking and managing locks, see Chapter 10, "Transactions and Versioning" in the *Sybase IQ System Administration Guide*.

Checkpoint hints

The time between checkpoints defaults to 60 minutes. The time between checkpoints can be adjusted when you start your server by changing the -gc and -gr options in the start_asiq command or in the *dbname.cfg* configuration file. The -gc switch specifies the number of minutes for the checkpoint timeout period. The -gr switch specifies the number of minutes for the maximum recovery time. The database engine uses both switches to calculate the checkpoint time.

The default values for checkpoint time and recovery time are sufficient and do not need to be changed. If you are advised to change the values of -gc and -gc, see Chapter 1, "Running the Database Server" in the *Sybase IQ Utility Guide* for details on setting these server switches.

Performance issues

This section notes a few settings that can impact performance. For complete information on diagnosing and resolving performance issues, see these chapters in the *Sybase IQ Performance and Tuning Guide*:

- To understand Sybase IQ memory, disk, and other resource use and their performance implications, see Chapter 5, "Managing System Resources"
- To use the IQ buffer cache monitor, see Chapter 6, "Monitoring and Tuning Performance"

Slow performance on a multi-CPU or hyperthreaded machine

Possible cause

Sybase IQ runs most efficiently when it knows how many physical CPUs are available to it. On a machine with hyperthreads turned on, or where Sybase IQ is unable to access all of the available CPUs, Sybase IQ will create too many threads and run less efficiently than it should.

Action

Start the server with -iqnumbercpus set to the number of CPUs available to Sybase IQ, overriding the physical number of CPUs. For details, see the -iqnumbercpus server option in Chapter 1, “Running the Database Server” in the *Sybase IQ Utility Guide*.

Sybase Central issues

This section contains information on troubleshooting issues related to the operation of Sybase Central.

Some Sybase Central fields do not display

Possible cause

System is using a dark background with white text.

Action

Use the facilities your windowing system provides to change the Sybase Central display to use dark text on a white or light background.

Troubleshooting network communications

The following sections are primarily for troubleshooting communications problems on Windows and with Windows-based clients.

Network software involves several different components, increasing the likelihood of problems. Although we provide some tips concerning network troubleshooting here, the primary source of assistance in network troubleshooting should be the documentation and technical support for your network communications software, as provided by your network communications software vendor.

Also see the section “Diagnostic tools” on page 38 for instructions on how to obtain information you can use in diagnosing various conditions, including those described in the following sections.

Ensuring that you are using compatible protocols

If you have more than one protocol stack installed on the client or server computer, you should ensure that the client and the database server are using the same protocol. The `-x` command line switch for the server selects a list of protocols for the server to use, and the CommLinks connection parameter does the same for the client application.

You can use these options to ensure that each application is using the same protocol.

By default, both the database server and client library use all available protocol stacks. The server supports client requests on any active protocol, and the client searches for a server on all active protocols.

More information about the `-x` switch is in Chapter 1, “Running the Database Server” in the *Sybase IQ Utility Guide*.

Ensuring that you have current drivers

Old network adapter drivers are a common source of communication problems. You should ensure that you have the latest version of the NDIS or ODI driver for your network adapter, as appropriate. You should be able to obtain current network adapter drivers from the manufacturer or supplier of the adapter card.

Network adapter manufacturers and suppliers make the latest versions of drivers for their cards available. Most card manufacturers have a Web site from which you can download the latest versions of NDIS and ODI drivers.

You may also be able to obtain a current network adapter driver from the provider of your networking software.

When you download Novell client software, ODI drivers for some network adapters are included in addition to the Novell software that is used for all network adapters.

Switching off your computer between reboots

Some network adapter boards do not reset cleanly when you reboot the computer. When you are troubleshooting, turn the computer off, wait a few seconds, and then turn it back on between reboots.

Diagnosing your protocol stack layer by layer

If you are having problems getting your client application to communicate with a database server, you need to ensure that the client and the database server are using compatible protocol stacks.

A helpful method of isolating network communication problems is to work up the protocol stack, testing whether each level of communication is working properly.

If you can connect to the server computer in any way, then the data link layer is working, regardless of whether the connection is made using the same higher-layer protocols you will be using for Sybase IQ.

For example, you may want to try to connect to a disk drive on the computer running the database server from the computer running the client application.

Having verified that the data link layer is working, the next step is to verify that other applications using the same network and transport layers as Sybase IQ are working properly.

Testing a NetBIOS protocol stack

If you are using Windows 98 or Windows NT/2000/ME/XP, and you are using the native protocol, try using the chat or WinPopup application. This tests whether applications on the client and server computers can communicate with each other.

You should ensure that the applications that come with your networking software are running properly before testing Sybase IQ.

Testing a TCP/IP protocol stack

If you are running under TCP/IP, there are several applications that you can use to test the compatibility of the client computer and server computer TCP/IP protocol stack. The ping utility provided with many TCP/IP packages is useful for testing the IP network layer.

Using ping to test the IP layer

Each IP layer has an associated address—a four-integer period-separated number (such as 191.72.109.12). Ping takes as an argument an IP address and attempts to send a single packet to the named IP protocol stack.

First, determine if your own protocol stack is operating correctly by "pinging" your own computer. For example, if your IP address is 191.72.109.12, enter:

```
ping 191.72.109.12
```

at the command line prompt and wait to see if the packets are routed at all. If they are, the output will appear similar to the following:

```
c:> ping 191.72.109.12
Pinging 191.72.109.12 with 32 bytes of data:
Reply from 191.72.109.12: bytes=32 time<.10ms TTL=32
Reply from 191.72.109.12: bytes=32 time<.10ms TTL=32
Reply from 191.72.109.12: bytes=32 time<.10ms TTL=32
...
```

If the ping works, then the computer is able to route packets to itself. This is reasonable assurance that the IP layer is set up correctly. Ask someone else running TCP/IP for their IP address and try pinging their computer.

Ensure that you can ping the computer running the database server from the client computer before proceeding.

Using Telnet to test the TCP/IP stack

To further test the TCP/IP stack, start a server application on one computer, and a client program on the other computer, and test whether they can communicate properly.

There are several applications commonly provided with TCP/IP implementations that can be used for this purpose. The following procedure shows how to use the telnet command to test the TCP/IP stack.

1 Start a Telnet server process (or **daemon**) on one machine. Check your TCP/IP software documentation to see how to do this. For a typical command line Telnet program, type the following instruction at the command prompt:

```
telnetd
```

2 Start the Telnet client process on the other machine, and see if you get a connection. Again, check your TCP/IP software documentation to see how to do this. For command line programs, you typically type the following instruction:

```
telnet server_name
```

where *server_name* is the name or IP address of the computer running the Telnet server process.

If a Telnet connection is established between these two machines, the protocol stack is stable and the client and server should be able to communicate using the TCP/IP link between the two computers. If a Telnet connection cannot be established, there is a problem. You should ensure that your TCP/IP protocol stack is working correctly before proceeding.

Diagnosing wiring problems

Faulty network wiring or connectors can cause problems that are difficult to isolate. Try recreating problems on a similar machine with the same configuration. If a problem occurs on only one machine, the issue may be a wiring problem or a hardware problem.

For information on detecting wiring problems under NetWare, see your Novell NetWare manuals. The Novell LANalyzer program is useful for diagnosing wiring problems with Ethernet or TokenRing networks. Your NetWare authorized reseller can also supply you with the name of a Certified NetWare Engineer who can help diagnose and solve wiring problems.

Checking common network communications problems

For a description of network communications parameters, see the section "Network communications parameters" in Chapter 4, "Connection and Communication Parameters" of the *Sybase IQ System Administration Guide*.

The following list presents some common network communications problems and their solutions.

"Unable to start — server not found" message

If you receive the message

```
Unable to start — server not found
```

when trying to start the client, the client cannot find the database server on the network. Check for the following problems:

- The network configuration parameters of your network driver on the client machine are different from those on the server machine. For example, two Ethernet adapter cards should be using a common frame type. For Novell NetWare, the frame type is set in the *net.cfg* file. Under Windows 98, Windows NT, and Windows 2000, the settings are accessed through the Control Panel Network Settings.
- Under the TCP/IP protocol, clients search for database servers by broadcasting a request. Such broadcasts typically do not pass through gateways, so any database server on a machine in another (sub)network, is not found. If this is the case, you must supply the host name of the machine on which the server is running using the `-x` server startup command-line option. This is required to connect to NetWare servers over TCP.
- Your network drivers are not installed properly or the network wiring is not installed properly.
- The network configuration parameters of your network driver are not compatible with Sybase IQ multi-user support.
- If your network communications are being carried out using TCP/IP and you are operating under Windows for Workgroups or Windows NT/2000/XP, check that your TCP/IP software conforms to the Winsock 1.1 standard.

"Unable to initialize any communication links" message

If you receive the message

```
Unable to initialize any communication links
```

no link can be established. The probable cause is that your network drivers have not been installed. The server and the client try to start communication links using all available protocols, unless you have specified otherwise using the `-x` server startup option. Check your network documentation to find out how to install the driver you need to use.

Diagnostic tools

This section tells how to obtain information you can use in diagnosing various conditions, including those described in the previous sections.

The sp_iqstatus stored procedure

The sp_iqstatus stored procedure provides a variety of IQ status information.

The following output is from the sp_iqstatus stored procedure:

```
Adaptive Server IQ (TM)              Copyright (c) 1992-2006 by Sybase, Inc.
                                     All rights reserved.
Version:                             12.7.0/040810/P/GA/MS/
                                     Windows 2000/32bit/2006-06-10 09:54:19
Time Now:                            2006-06-11 18:53:34.274
Build Time:                          2006-06-10 09:54:19
File Format:                         23 on 03/18/1999
Server mode:                         IQ Server
Catalog Format:                      2
Stored Procedure Revision:           1
Page Size:                           131072/8192blksz/16bpp
Number of DB Spaces:                 1
Number of Temp Spaces:               1
DB Blocks: 1-5632                    IQ_SYSTEM_MAIN
Temp Blocks: 1-2816                  IQ_SYSTEM_TEMP
Create Time:                         2006-06-03 14:14:06.124
Update Time:                         2006-06-03 14:14:26.687
Main IQ Buffers:                     127, 16Mb
Temporary IQ Buffers:                95, 12Mb
Main IQ Blocks Used:                 4541 of 5632, 80%=35Mb, Max Block#: 5120
Temporary IQ Blocks Used:            65 of 2816, 2%=0Mb, Max Block#: 0
Main Reserved Blocks Available:      512 of 512, 100%=4Mb
Temporary Reserved Blocks Available: 256 of 256, 100%=2Mb
IQ Dynamic Memory:                   Current: 41mb, Max: 41mb
Main IQ Buffers:                     Used: 4, Locked: 0
Temporary IQ Buffers:                Used: 4, Locked: 0
Main IQ I/O:                         I: L168/P2 O: C2/D16/P15 D:0 C:100.0
Temporary IQ I/O:                    I: L862/P0 O: C136/D150/P17 D:132 C:100.0
Other Versions:                      0 = 0Mb
Active Txn Versions:                 0 = C:0Mb/D:0Mb
```

The following is a key to understanding the `Main IQ I/O` and `Temporary IQ I/O` output codes:

- `I:` Input
- `L:` Logical pages read (“Finds”)
- `P:` Physical pages read
- `O:` Output
- `C` Pages Created
- `D` Pages Dirtied
- `P:` Physically Written
- `D:` Pages Destroyed
- `C:` Compression Ratio

Check the following information:

- The lines `Main IQ Blocks Used` and `Temporary IQ Blocks used` tell you what portion of your dbspaces is in use. If the percentage of blocks in use (the middle statistic on these lines) is in the high nineties, you need to add a dbspace.
- The lines `Main IQ Buffers` and `Temporary IQ Buffers` tell you the current sizes of your main and temp buffer caches.
- `Other Versions` shows other db versions and the total space consumed. These versions will eventually be dropped when they are no longer referenced or referencable by active transactions.
- `Active Txn Versions` shows the number of active write transactions and the amount of data they have created and destroyed. If these transactions commit, the “destroyed” data will become an old version and eventually be dropped. If they rollback, the “created” data will be freed.
- `Main Reserved Blocks Available` and `Temporary Reserved Blocks Available` show the amount of reserved space that is available.
- The lines `Main IQ I/O` and `Temporary IQ I/O` display I/O status in the same format as in the IQ message log. For an explanation of these statistics, see the section “Main buffer cache activity message” in Chapter 7, “Moving Data In and Out of Databases” of the *Sybase IQ System Administration Guide*.

The sp_iqcheckdb stored procedure

If you suspect problems in your database, try running the stored procedure sp_iqcheckdb. This procedure reads every database page from disk into memory and does various consistency checks. However, depending on the size of your database, the check can take a long time to run.

The sp_iqdbstatistics stored procedure displays the database statistics collected by the most recent execution of the sp_iqcheckdb procedure.

For more information on running and using the sp_iqcheckdb and sp_iqdbstatistics stored procedures, see “Database verification” in Chapter 2, “System Recovery and Database Repair.”

Checking database and server startup option values

When diagnosing server startup, resource, or processing issues, you may need to check the current values of database options and server startup options. For the connected user, the sp_iqcheckoptions stored procedure displays a list of the current value and the default value of database options that have been changed from the default. sp_iqcheckoptions also lists server startup options that have been changed from the default values.

When sp_iqcheckoptions is run, the DBA sees all options set on a permanent basis for all groups and users and sees temporary options set for DBA. Non-DBA users see their own temporary options. All users see non-default server startup options.

The sp_iqcheckoptions stored procedure requires no parameters. In Interactive SQL, run the following command:

```
sp_iqcheckoptions
```

The system table DBA.SYSOPTIONDEFAULTS contains all of the names and default values of the Sybase IQ and ASA options. You can query this table, if you need to see all option default values.

For more information, see “sp_iqcheckoptions procedure” in Chapter 10, “System Procedures” and “Finding option settings” in Chapter 2, “Database Options” of the *Sybase IQ Reference Manual*.

Finding the currently executing statement

When diagnosing a problem, you may want to know what statement was executing when the problem occurred. The sp_iqcontext stored procedure tells you what statements are running on the system when you run the procedure, and identifies the user and connection that issued the statement. You can use this utility together with information provided by sp_iqconnection, the *.iqmsg* log, and the -zr SQL log, as well as stack traces, to determine what was happening when a problem occurred.

For details and sample output, see “sp_iqcontext procedure” in Chapter 10, “System Procedures” of the *Sybase IQ Reference Manual*. To match *.iqmsg* log and the -zr SQL log entries using connection information, see “Correlating connection information” on page 47.

Logging server requests

For isolating some types of problems, especially problems with queries, logging server requests is helpful. You can enable request-level logging in two ways:

- By setting the -zr command-line option when you start the server.
- By calling the sa_server_option stored procedure, which overrides the current setting of the -zr command-line option.

Server requests are logged in the server log file **.srvlog*. The -zr server startup option enables request-level logging of operations and sets the type of requests to log (ALL, NONE, or SQL). The -zo option redirects request-level logging information to a file separate from the regular log file and -zs limits the size of this file.

You can enable and disable request-level logging without restarting the Sybase IQ server using the sa_server_option stored procedure. The following commands enable request-level logging of a limited set of requests and redirect the output to the file *sqllog.txt*:

```
call sa_server_option('request_level_logging','SQL');
call sa_server_option('request_level_log_file',
                       'sqllog.txt');
```

The following command disables request-level logging:

```
call sa_server_option('request_level_log_file','');
```

To view the current settings for the SQL log file and logging level, execute the following statement:

```
select property('RequestLogFile'),
       property('RequestLogging');
```

To match *.iqmsg* log and the -zr SQL log entries using connection information, see "Correlating connection information" on page 47.

The following output shows the limited requests that are logged, when the server is started with the -zr SQL option. In this example, the user connects to the asiqdemo database, executes the command

```
SELECT * FROM customer
```

then disconnects. In the server log, each line is preceded by a datetime stamp, which has been removed from this sample output.

```
REQUEST conn: 1182240 CONNECT
DONE    conn: 1182240 CONNECT          Conn=1330880449
REQUEST conn: 1182240 STMT_EXECUTE_IMM  "SET TEMPORARY OPTION
CONNECTION_AUTHENTICATION='Company=MyComp;Application=DBTools;Signature=4f13d
14f3'"
DONE    conn: 1182240 STMT_EXECUTE_IMM
REQUEST conn: 1182240 STMT_PREPARE      "SELECT user, db_property('IQStore')
FROM SYS.DUMMY"
DONE    conn: 1182240 STMT_PREPARE     Stmt=65536
REQUEST conn: 1182240 STMT_EXECUTE     Stmt=65536
DONE    conn: 1182240 STMT_EXECUTE
REQUEST conn: 1182240 STMT_DROP        Stmt=65536
DONE    conn: 1182240 STMT_DROP
REQUEST conn: 1182240 STMT_PREPARE      "SELECT default_collation FROM
SYS.SYSINFO"
DONE    conn: 1182240 STMT_PREPARE     Stmt=65537
REQUEST conn: 1182240 STMT_EXECUTE     Stmt=65537
DONE    conn: 1182240 STMT_EXECUTE
REQUEST conn: 1182240 STMT_DROP        Stmt=65537
DONE    conn: 1182240 STMT_DROP
REQUEST conn: 1182240 STMT_PREPARE      "SELECT connection_property('charset')
FROM SYS.DUMMY"
DONE    conn: 1182240 STMT_PREPARE     Stmt=65538
REQUEST conn: 1182240 STMT_EXECUTE     Stmt=65538
DONE    conn: 1182240 STMT_EXECUTE
REQUEST conn: 1182240 STMT_DROP        Stmt=65538
DONE    conn: 1182240 STMT_DROP
REQUEST conn: 1182240 STMT_PREPARE     "select @@servername"
DONE    conn: 1182240 STMT_PREPARE     Stmt=65539
REQUEST conn: 1182240 STMT_EXECUTE     Stmt=65539
```

```
DONE    conn: 1182240 STMT_EXECUTE
REQUEST conn: 1182240 STMT_DROP        Stmt=65539
DONE    conn: 1182240 STMT_DROP
REQUEST conn: 1182240 STMT_PREPARE     "SELECT CURRENT DATABASE FROM DUMMY"
DONE    conn: 1182240 STMT_PREPARE     Stmt=65540
REQUEST conn: 1182240 CURSOR_OPEN      Stmt=65540
DONE    conn: 1182240 CURSOR_OPEN      Crsr=65541
REQUEST conn: 1182240 CURSOR_CLOSE     Crsr=65541
DONE    conn: 1182240 CURSOR_CLOSE
REQUEST conn: 1182240 STMT_DROP        Stmt=65540
DONE    conn: 1182240 STMT_DROP
REQUEST conn: 1182240 STMT_PREPARE     "select * from customer"
DONE    conn: 1182240 STMT_PREPARE     Stmt=65542
REQUEST conn: 1182240 CURSOR_OPEN      Stmt=65542
DONE    conn: 1182240 CURSOR_OPEN      Crsr=65543
REQUEST conn: 1182240 CURSOR_EXPLAIN   Crsr=65543
DONE    conn: 1182240 CURSOR_EXPLAIN
REQUEST conn: 1182240 CURSOR_CLOSE     Crsr=65543
DONE    conn: 1182240 CURSOR_CLOSE
REQUEST conn: 1182240 STMT_DROP        Stmt=65542
DONE    conn: 1182240 STMT_DROP
REQUEST conn: 1182240 DISCONNECT
DONE    conn: 1182240 DISCONNECT
```

For more information on request-level logging, see the section “Server command-line switches” in Chapter 1, “Running the Database Server” of the *Sybase IQ Utility Guide* and “sa_server_option system procedure” in Chapter 10, “System Procedures”of the *Sybase IQ Reference Manual*.

Connection for collecting diagnostic information

The database option DEDICATED_TASK lets the DBA dedicate a request handling task to handling requests from a single connection. This pre-established connection allows you to gather information about the state of the database server if it becomes otherwise unresponsive. For more information, see “DEDICATED_TASK option” in *Sybase IQ Reference Manual*.

Diagnosing communications issues

If your server is having communications problems, you may want to set the -z command-line option when you start the server. This switch provides diagnostic information on communications links at server startup.

Reporting problems to Technical Support

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, the designated person should contact Sybase Technical Support or the Sybase subsidiary in your area.

Technical Support needs information about your Sybase IQ environment in order to resolve your problem. This section describes this information, tells you how to collect it using the automated getiqinfo tool, and explains how to correlate information in various Sybase IQ utilities and log files.

Collecting diagnostic information using getiqinfo

Sybase IQ includes a script for collecting information that Sybase Technical Support needs to diagnose problems. The getiqinfo script collects information about the operating system environment, the Sybase IQ environment, and log files.

Run this script before reporting a problem to Sybase Technical Support. By doing so, you can help Sybase staff resolve your issue more quickly, with less effort on your part.

The getiqinfo script automatically collects all of the information discussed in "Reporting problems to Technical Support" on page 44, as well as other information that may be needed to resolve your issue.

The getiqinfo script is not designed for troubleshooting Sybase IQ installations and does not provide on-site troubleshooting facilities. This script executes successfully only when the Sybase IQ environment is properly set up and the server is running.

Before you run getiqinfo

Have the following information ready before running the script:

- Location of the database file

- Full path of the configuration file used to start the server, if one is used
- Full path of the *.iqmsg* file, if the Sybase IQ message file has been renamed

If possible, leave the Sybase IQ server running, or start the server before running getiqinfo. This allows the script to collect internal database data that is only available when Sybase IQ is running. The script does not automatically start the server.

The script runs with the same environment settings that are used to start the Sybase IQ server. getiqinfo uses some IQ-specific environment variables to search for files.

The script puts collected data in the current directory (where you start the program). Be sure you have enough space under that directory. The script does not prompt for an alternative, but you can modify the script to change the output location by resetting the variable DEST_DIR.

Running the getiqinfo script

On UNIX platforms, getiqinfo is a shell script. On Windows platforms, *getiqinfo.bat* is a batch script in the *ASIQ-12_7\win32* directory.

❖ **To run getiqinfo:**

The steps vary for UNIX and Windows platforms.

1 Start the script according to your platform:

- At the UNIX command prompt, in the *ASIQ-12_7/bin* directory, type:

```
getiqinfo.sh
```

- In the Windows menu, enter Start → Run → *<install_path>\ASIQ-12_7\win32\getiqinfo.bat*.

2 As the program prompts you, enter:

- The directory of the database file. This is also the default location of the *.iqmsg* file, and the *stktrc*.iq* file on UNIX.
- The base name of the database file (the file name without the *.db* suffix). This is also the default base name of the *.iqmsg* file.
- Other directories to search for these files
- Sybase IQ engine name (server name) and port number for this database server
- User ID and password with DBA privileges for this database
- The full path to the configuration file used to start the Sybase IQ server, if one was used

  - The full path to the output file in the -zo server option, if one was specified

The program also directs you to send the listed files to Sybase Technical Support.

Information collected by getiqinfo

The getiqinfo script collects all of the following information:

- Type of hardware, amount of memory, CPU type, speed, number of CPUs
- Operating system (for example, Sun Solaris 2.9)
- Swap space size
- Sybase IQ version and EBF level, and Anywhere version
- Stack trace file for the date and time this problem occurred, named *stktrc-YYYYMMDD-HHMMSS_#.iq*, in the directory where you started the database server. (UNIX and Linux platforms only)
- Command or query that produced the error
- Message log file, named *dbname.iqmsg*, located by default in the directory where you started the database server.
- Query plan (recorded in *.iqmsg* file; see the Note below)
- Server logs
  - For UNIX, *ASIQ-12_7/logfiles/<servername>.00n.stderr* and *ASIQ-12_7/logfiles/<servername>.00n.srvlog*
  - On Windows platforms, if needed, you must restart the server and manually collect a copy of the console window.
- Startup and connection option settings, from the configuration file (by default, *dbname.cfg*)
- Database option settings and output from sa_conn_properties (if the server is still running)

On multiplex databases, you must execute getiqinfo on the write server and/or query servers, and the following information is also collected:

- *servername.out* on query servers
- *write_server_name/repDirs/logfiles/servername.dbrlog* on the write server

The following information is not collected by getiqinfo, but may also be requested by Technical Support:

- Connectivity protocol used (for example, ODBC, JDBC, TDS)
- Open Client version
- Configuration type (single user or multi-user)
- Front end tool used (for example, Brio Query)
- Schema and indexes for the database
- Output from sp_iqcheckdb procedure

A checklist for recording information that Technical Support may need is provided at the end of this chapter, in the unlikely event that you need to collect this information manually.

Note Query plan detail is collected automatically by getiqinfo if the options below are set. You can also collect this information manually, by setting the options and then rerunning the command that produced the error.

```
SET TEMPORARY OPTION QUERY_PLAN = 'ON'
SET TEMPORARY OPTION QUERY_DETAIL = 'ON'
```

The plan will be in the message log file. The default for databases created as of Version 12.5 is Query_Plan = ON, and Query_Detail = OFF.

If you have performance problems, set the following option:

```
SET TEMPORARY OPTION QUERY_PLAN_AFTER_RUN = 'ON'
```

This will enable technical support to see which steps in the query processing used the time.

Correlating connection information

Technical Support may ask you to set the -zr option on the start_asiq command in your configuration file. This server startup option sets the request logging level to track statements sent to the server. Parameters are ALL, NONE, or SQL. The option produces a log file named for the database, with the suffix *.zr*. In the log file, each connection to the server is identified by a connection handle.

Because the connection handle is not unique, Sybase IQ assigns its own Sybase IQ connection ID, which is displayed in the Sybase IQ message file. The Sybase IQ message file records the errors, warnings, and tracing information for each connection. Because the two files use different identifiers for the connections, you cannot compare the *.zr* output with the *.iqmsg* file and easily locate information for a particular connection.

The following procedure tells how to correlate the identifiers in the two files to find relevant information. For example, assume that the *.zr* output file is *example.zr* and the Sybase IQ message file is *example.iqmsg*.

❖ **To correlate connection information between the .zr and .iqmsg files:**

1 In the *.zr* file, locate a connection of interest, for example:

```
.conn: 240215640
```

For example, on a UNIX system:

```
grep 240215640 example.zr | grep CONNECT
```

```
04/19 06:42:06.690 ** REQUEST conn: 240215640 CONNECT
04/19 06:42:07.204 ** DONE    conn: 240215640 CONNECT    Conn=569851433
04/19 06:46:17.646 ** REQUEST conn: 240215640 DISCONNECT
04/19 06:46:17.670 ** DONE    conn: 240215640 DISCONNECT
```

2 In the same line, find the number that follows `Conn=`. In this example:

```
Conn=569851433
```

3 Search the *.iqmsg* file for “Connection handle is” followed by that number. For example:

```
grep 569851433 example.iqmsg
```

```
2006-06-19 07:46:57 0000000002 Connection handle is : 569851433. SA
connID: 2.User Name is : DBA.
```

The Sybase IQ connection handle in this example is 000000002.

4 Isolate all the lines from the *.iqmsg* file for that connection:

```
grep ' 0000000002 ' example.iqmsg
```

Another source of helpful information

If you are unable to resolve a problem, you may find additional help on the Sybase online support Web site, MySybase. MySybase lets you search through closed support cases, latest software bulletins, and resolved and known problems, using a view customized for your needs. You can even open a Technical Support case online.

MySybase can be used from most Internet browsers. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/ and click MySybase for information on how to sign up for and use this free service.

MySybase can be used from most Internet browsers. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/ and click MySybase for information on how to sign up for and use this free service.

Checklist: information for Technical Support

Information requested	Value
type of hardware	
amount of memory	
number of CPUs	
operating system name and version (e.g., Sun Solaris 2.10 (SPARC))	
operating system patch level	
front end tool used (e.g., Brio Query)	
connectivity protocol used (e.g., ODBC, JDBC, TDS)	
Open Client version	
configuration type (single or multiuser)	
message log file (dbname.iqmsg)	
stack trace file stktrc-YYYYMMDD-HHNNSS_#.iq UNIX platforms only	
command or query that produced the error	
startup option settings	
connect option settings	
database option settings	
schema and indexes for the database	
sp_iqstatus output	
query plan: set options (Query_Plan, Query_Detail, Query_Plan_After_Run), rerun command or query	

CHAPTER 2

System Recovery and Database Repair

About this chapter

When you restart the database server, Sybase IQ attempts to recover automatically. If the server is unable to recover and restart, especially after a system failure or power outage, the database may be inconsistent. This chapter describes what happens during normal recovery, how to verify database consistency, how to repair database inconsistencies, and special recovery modes.

Contents

Topic	**Page**
Recovery and repair overview	51
Normal recovery	52
Database verification	52
Database repair	62
Forced recovery mode	73
Emergency recovery without a transaction log	79
Handling unrepairable problems	80
DBCC error messages	83

Recovery and repair overview

If your Sybase IQ server or database has problems restarting, use the information in this chapter to diagnose database startup problems, verify the consistency of databases, and repair databases. If you are able to restart the server after a failure, Sybase recommends that you verify your database, preferably before allowing users to connect. You verify databases using the sp_iqcheckdb stored procedure, as described in this chapter.

If you have trouble starting a server or database, if the database starts but users are unable to connect to it, or if problems are found during database verification, you may need to perform a forced recovery or restore the database.

This chapter explains how to determine when you need to perform each of these functions. It describes database verification, forced recovery, leaked space recovery, and index repair. For details on restoring databases, see Chapter 14, “Data Backup, Recovery, and Archiving” in the *Sybase IQ System Administration Guide*.

Examining the server log and IQ message log

To determine what type of recovery or repair is needed, you need information from your server log (*servername.nnn.srvlog*) and IQ message log (*dbname.iqmsg*). Be sure to retain this information so you can provide it to Sybase Technical Support if necessary.

For example, if data inconsistency is detected, the *dbname.iqmsg* file may include detailed diagnostic information.

Normal recovery

During system recovery, any uncommitted transactions are rolled back and any disk space used for old versions (snapshots of database pages that were being used by transactions that did not commit) returns to the pool of available space. The database then contains only the most recently committed version of each permanent table, unless it is a multiplex database. A multiplex database contains all versions accessible to query servers. For more information on versioning, see Chapter 10, “Transactions and Versioning” in the *Sybase IQ System Administration Guide*.

During recovery from a system failure or normal system shutdown, Sybase IQ reopens all connections that were active. If the `-gm` option, which sets the number of user connections, was in effect at the time of the failure, you need to restart the IQ server with at least as many connections as were actually in use when the server stopped.

Database verification

Check the consistency of your database as soon as possible after the server restarts following an abnormal termination, such as a power failure. Database consistency should also be checked before performing a backup of the database. In both of these cases, you can use the `sp_iqcheckdb` stored procedure to detect and repair database consistency problems.

This section describes using sp_iqcheckdb for database verification. The section "Database repair" contains details on using sp_iqcheckdb to repair the consistency problems detected.

The sp_iqcheckdb stored procedure

The IQ Database Consistency Checker (DBCC) performs database verification and repair functions. The sp_iqcheckdb stored procedure, in conjunction with server startup options, is the interface to DBCC. You select the different modes of check and repair by specifying an sp_iqcheckdb command string. sp_iqcheckdb reads every database page and checks the consistency of the database, unless you specify otherwise in the command string.

Note On a query server sp_iqcheckdb does not check the freelist. It performs all other checks.

DBCC has three different modes that perform increasing amounts of consistency checking and a repair mode for repairing inconsistent indexes. Each mode checks all database objects, unless individual tables, indexes, or index types are specified in the sp_iqcheckdb command string. If you specify individual table names, all indexes within those tables are also checked.

Note The sp_iqcheckdb stored procedure does not check referential integrity or repair referential integrity violations.

sp_iqcheckdb syntax

The following command line shows the basic syntax of sp_iqcheckdb.

sp_iqcheckdb '*mode target* [...] [resources *resource-percent*]'

mode:
{ allocation | check | verify } | repair

target:
[main | local | indextype *index-type* […]] database{ dumpdups | dumpleaks | dumpunallocs } | database resetclocks |
{ [indextype *index-type*] […] table *table-name* | index *index-name* […] }

If both mode and target are not specified in the parameter string, IQ returns the error message "At least one mode and target must be specified to DBCC."

The *index-name* parameter may contain owner and table qualifiers: `[[owner.]table-name.]index-name`. If *owner* is not specified, current user and database owner (dbo) are substituted in that order. If *table* is not specified, then *index-name must* be unique.

The *table-name* parameter may contain an owner qualifier: `[owner.]table-name`. If *owner* is not specified, current user and database owner (dbo) are substituted in that order. *table-name* cannot be a temporary or pre-join table.

Note The sp_iqcheckdb parameter string must be enclosed in single quotes and must not be greater than 255 bytes in length.

If either the table name or the index name contains spaces, enclose the *table-name* or *index-name* parameter in double quotes, as shown in this example:

```
sp_iqcheckdb 'check index "dbo.ss tab.i2" resources 75'
```

Refer to the section "sp_iqcheckdb procedure" in Chapter 10, "System Procedures" of the *Sybase IQ Reference Manual* for the complete syntax of sp_iqcheckdb.

The following table summarizes the actions and output of the four sp_iqcheckdb modes.

Table 2-1: sp_iqcheckdb modes: actions and output

Mode	Errors detected	Output	Speed
allocation	allocation errors	allocation statistics only	4TB per hour
check	allocation errors most index errors	all available statistics	60GB per hour
verify	allocation errors all index errors	all available statistics	15GB per hour
repair	all index errors	repair statistics	15+GB per hour*

* The processing time of sp_iqcheckdb repair mode depends on the number of errors repaired.

DBCC performance

The execution time of DBCC varies according to the size of the database for an entire database check, the number of tables or indexes specified, and the size of the machine. Checking only a subset of the database, i.e., only specified tables, indexes, or index types, requires less time than checking an entire database. Refer to the table in the previous section for processing times of the sp_iqcheckdb modes.

For the best DBCC performance, you should be as specific as possible in the sp_iqcheckdb command string. Use the 'allocation' or 'check' verification mode when possible and specify the names of tables or indexes, if you know exactly which database objects require checking.

sp_iqcheckdb check mode

In check mode, sp_iqcheckdb performs an internal consistency check on all IQ indexes and checks that each database block has been allocated correctly. All available database statistics are reported. This mode reads all data pages and can detect all types of allocation problems and most types of index inconsistencies. Check mode should run considerably faster than verify mode for most databases.

When to run in check mode:

- If metadata, null count, or distinct count errors are returned when running a query

Examples of check mode:

Table 2-2: sp_iqcheckdb check mode examples

Command	Description
`sp_iqcheckdb 'check database'`	Internal checking of all tables and indexes in the database
`sp_iqcheckdb 'check table t1'`	Default checking of all indexes in table t1
`sp_iqcheckdb 'check index t1c1hg'`	Internal checking of index t1c1hg
`sp_iqcheckdb 'check indextype FP database'`	Checking of all indexes of type FP in the database

sp_iqcheckdb verify mode

In verify mode, sp_iqcheckdb performs an intra-index consistency check, in addition to internal index consistency and allocation checking. All available database statistics are reported. The contents of each non-FP index is verified against its corresponding FP index(es). Verify mode reads all data pages and can detect all types of allocation problems and all types of index inconsistencies.

When to run in verify mode:

- If metadata, null count, or distinct count errors are returned when running a query

Examples of verify mode:

Table 2-3: sp_iqcheckdb verify mode examples

Command	Description
`sp_iqcheckdb 'verify database'`	Verify contents of all indexes in the database
`sp_iqcheckdb 'verify table t1'`	Verify contents of all indexes in table t1
`sp_iqcheckdb 'verify index t1c1hg'`	Verify contents of index t1c1hg
`sp_iqcheckdb 'verify indextype HG table t1'`	Verify contents of all HG indexes in table t1

Note If you check individual non-FP indexes in check mode, the corresponding FP index(es) are automatically verified with internal consistency checks and appear in the DBCC results.

sp_iqcheckdb allocation mode

In allocation mode, sp_iqcheckdb checks that each database block is allocated correctly according to the internal physical page mapping structures (blockmaps). Database statistics pertaining to allocation are also reported. This mode executes very quickly. Allocation mode, however, does not check index consistency and cannot detect all types of allocation problems.

When to run in allocation mode:

- To check for leaked blocks or inconsistent indexes due to multiply owned blocks
- After forced recovery, run sp_iqcheckdb with the -iqdroplks server switch to reset the allocation map (must use database as the target)
- To check for duplicate or unowned blocks (use database or specific tables or indexes as the target)
- If you encounter page header errors

Examples of allocation mode:

Table 2-4: sp_iqcheckdb allocation mode examples

Command	Description
`sp_iqcheckdb 'allocation database'`	Allocation checking of entire database
`sp_iqcheckdb 'allocation database dumpleaks'`	Allocation checking of entire database and print block numbers for leaked blocks to IQ message file
`sp_iqcheckdb 'allocation table t1'`	Allocation checking of table t1
`sp_iqcheckdb 'allocation index t1c1hg'`	Allocation checking of index t1c1hg
`sp_iqcheckdb 'allocation indextype LF table t2'`	Allocation checking of all LF indexes in table t2

Allocation mode options are only allowed with the DBCC command 'allocation database'. They may be used with the -droplks server startup switch, but do not require it. For example, you can start the database with -iqfrec but without -droplks, and use allocation mode to find inconsistent indexes. You may then choose to restart the server with both -iqfrec and -droplks in order to repair problems found.

The following allocation mode options print block numbers for affected database blocks to the IQ message file:

- dumpleaks — leaked blocks
- dumpdups — duplicate blocks
- dumpunallocs — unallocated blocks

The DBCC option resetclocks is used in conjunction with forced recovery to convert a multiplex query server to a write server. The resetclocks option corrects the values of internal database versioning clocks, in the event that these clocks are slow. Do not use the resetclocks option for any other purpose unless you contact Sybase IQ Technical Support.

The resetclocks option must be run in single user mode and is only allowed with the DBCC command 'allocation database'. resetclocks does not require the -iqdroplks server startup switch. The syntax of the resetclocks command is:

```
sp_iqcheckdb 'allocation database resetclocks'
```

See the section “Replacing write servers” on page 78 for more information on converting a multiplex query server to a write server.

sp_iqcheckdb repair mode

The repair mode of sp_iqcheckdb performs detailed index checking and can repair many types of index inconsistencies. Allocation checks are not performed during repair mode. The DBCC output indicates which indexes, if any, were repaired. You must issue a COMMIT command to commit changes from repair mode to make them permanent. After DBCC repairs an index, you should run sp_iqcheckdb again in check mode before committing any changes. If an index is still inconsistent, drop and recreate the index.

When to run in repair mode:

- If index errors are reported in sp_iqcheckdb check or verify mode

Examples of repair mode:

Table 2-5: sp_iqcheckdb repair mode examples

Command	Description
`sp_iqcheckdb 'repair database'`	Detailed check and repair of entire database
`sp_iqcheckdb 'repair table t1'`	Detailed check and repair of table t1
`sp_iqcheckdb 'repair index t1c1hg'`	Detailed check and repair of index t1c1hg
`sp_iqcheckdb 'repair indextype HG database'`	Detailed check and repair of all HG indexes in the database

Note There is currently no support for repairing join indexes.

You should back up the database before executing sp_iqcheckdb in an index repair mode.

sp_iqcheckdb output

The output of sp_iqcheckdb consists of an extensive list of statistics and any errors reported by DBCC. Only non-zero values are displayed. Lines containing errors are flagged with asterisks (*****). Note that if you encounter errors, some of the statistics reported by DBCC may be inaccurate.

See the section “DBCC error messages” on page 83 for the full list of DBCC error messages.

The output of sp_iqcheckdb is always copied to the IQ message file (*.iqmsg*). To redirect the sp_iqcheckdb output to a file, enter the following command:

```
sp_iqcheckdb ># file_name
```

where *file_name* is the name of the file to receive the output.

When the DBCC_LOG_PROGRESS option is ON, sp_iqcheckdb sends progress messages to the IQ message file. These messages allow the user to follow the progress of the sp_iqcheckdb procedure as it executes.

The following is sample progress log output of the command `sp_iqcheckdb ‘check database’`

```
IQ Utility Check Database
Start CHECK STATISTICS table: tloansf
Start CHECK STATISTICS for field: aqsn_dt
Start CHECK STATISTICS processing index:
ASIQ_IDX_T444_C1_FP
Start CHECK STATISTICS processing index:
tloansf_aqsn_dt_HNG
Done CHECK STATISTICS field: aqsn_dt
```

Future Version Errors

If you see the message “DBCC Future Version Errors,” a DDL operation has been performed since the DBCC transaction began. DBCC continues to process the remaining tables, but leaked block checking is not performed and statistics do not include the tables that were skipped.

To avoid DBCC Future Version errors, execute the COMMIT command before you run sp_iqcheckdb.

The following DBCC output indicates a Future Version error:

```
=======================================|===========================|=====
DBCC Verify Mode Report                |                           |
=======================================|===========================|=====
** DBCC Future Version Errors          |1                          |*****
```

Sample output of valid database

The following is an example of running sp_iqcheckdb in verify mode. No errors are detected, there is no leaked space, the database allocation is consistent, and all indexes are consistent.

The command line for this example is `sp_iqcheckdb 'verify database'`. Note that DBCC verifies all indexes, but the index verification output shown here is abbreviated.

Each index that DBCC determines to be consistent is marked as verified in the result set.

```
         Stat                               Value                    Flags
=================================|==================================|=====
DBCC Verify Mode Report          |                                  |
=================================|==================================|=====
   DBCC Status                   |No Errors Detected                |
   DBCC Work units Dispatched    |75                                |
   DBCC Work units Completed     |75                                |
=================================|==================================|=====
Index Summary                    |                                  |
=================================|==================================|=====
   Verified Index Count          |86                                |
=================================|==================================|=====
Allocation Summary               |                                  |
=================================|==================================|=====
   Blocks Total                  |8192                              |
   Blocks in Current Version     |4855                              |
   Blocks in All Versions        |4855                              |
   Blocks in Use                 |4855                              |
   % Blocks in Use               |59                                |
=================================|==================================|=====
Allocation Statistics            |                                  |
=================================|==================================|=====
   DB Extent Count               |1                                 |
   Blocks Created in Current TXN |211                               |
   Blocks To Drop in Current TXN |212                               |
   Marked Logical Blocks         |8240                              |
   Marked Physical Blocks        |4855                              |
   Marked Pages                  |515                               |
   Blocks in Freelist            |126422                            |
   Imaginary Blocks              |121567                            |
   Highest PBN in Use            |5473                              |
   Total Free Blocks             |3337                              |
   Usable Free Blocks            |3223                              |
   % Total Space Fragmented      |1                                 |
   % Free Space Fragmented       |3                                 |
   Max Blocks Per Page           |16                                |
   1  Block Page Count           |104                               |
   3  Block Page Count           |153                               |
   ...
   16 Block Hole Count           |199                               |
=================================|==================================|=====
Index Statistics                 |                                  |
=================================|==================================|=====
   ...
   Verified Index                |fin_data.DBA.ASIQ_IDX_T209_C3_HG  |
   Verified Index                |fin_data.DBA.ASIQ_IDX_T209_C4_FP  |
   Verified Index               |product.DBA.ASIQ_IDX_T210_C1_FP   |
   ...
   Verified Index               |employee.DBA.ASIQ_IDX_T212_C20_FP |
   Verified Index               |iq_dummy.DBA.ASIQ_IDX_T213_C1_FP  |
   FP Indexes Checked           |68                                |
   HNG Indexes Checked          |1                                 |
   HG Indexes Checked           |17                                |
=================================|==================================|=====
   ...
```

The DBCC output also contains extensive statistical information grouped under headings such as Container Statistics, Buffer Manager Statistics, Catalog Statistics, Connection Statistics, and Compression Statistics. You can see an example of the available statistics by executing the command `sp_iqcheckdb 'verify database'` after connecting to the Sybase IQ demonstration database asiqdemo.

Resource issues running sp_iqcheckdb

If you experience a resource problem while running sp_iqcheckdb, you may see one the following messages in the sp_iqcheckdb output or in the *.iqmsg* file:

- `Out of memory` and `DBCC Out of Memory Errors` You do not have enough memory for this operation. You may need to prevent other IQ operations or other applications from running concurrently with the sp_iqcheckdb stored procedure.
- `No buffers available` and `DBCC Out of Buffers Errors` The DBA may need to increase the buffer cache size.

Buffer cache sizes are set permanently using the database options MAIN_CACHE_MEMORY_MB and TEMP_CACHE_MEMORY_MB. The server startup switches -iqmc and -iqtc can be used to override the buffer cache size values set using the database options. See the section “Setting buffer cache sizes” in Chapter 5, “Managing System Resources” of the *Sybase IQ Performance and Tuning Guide* for information on using both the database options and the server startup switches to set buffer cache sizes.

You should not run multiple database consistency checks at the same time, as DBCC is optimized to run one instance.

The CPU utilization of DBCC can be limited by specifying the sp_iqcheckdb parameter resources *resource-percent*, which controls the number of threads with respect to the number of CPUs. The default value of *resource-percent* is 100, which creates one thread per CPU and should match the load capacity of most machines. Set *resource-percent* to a value less than 100 to reduce the number of threads, if you are running DBCC as a background process. The minimum number of threads is 1.

If *resource-percent* > 100, then there are more threads than CPUs, which may increase performance for some machine configurations.

The database option DBCC_PINNABLE_CACHE_PERCENT can be used to tune DBCC buffer usage. The default of DBCC_PINNABLE_CACHE_PERCENT is to use 50% of cache. For more information on the DBCC_PINNABLE_CACHE_PERCENT option, see Chapter 2, "Database Options" in the *Sybase IQ Reference Manual*.

Database repair

DBCC can repair both index inconsistencies and allocation problems. The sp_iqcheckdb repair mode is used to repair indexes. The sp_iqcheckdb allocation mode is used in conjunction with the database server startup option -iqdroplks to repair allocation problems.

The repair mode of DBCC can repair many types of index inconsistencies. Repair mode is invoked using the repair keyword in the sp_iqcheckdb command string. No special server startup options are needed for sp_iqcheckdb to fix index problems. The DBCC output indicates which indexes, if any, were repaired. Allocation checks are not performed in repair mode.

Allocation problems can be repaired by starting the database server with the -iqdroplks option and running sp_iqcheckdb 'allocation database'. This is the only DBCC command allowed with the -iqdroplks switch. If DBCC detects index inconsistencies while attempting allocation repair, an error is generated and allocation problems are not fixed.

See the section "Recovering leaked space" on page 76 for specific information on recovering leaked blocks, (blocks that are allocated, but not used).

Analyzing index errors

This section describes how to analyze index inconsistencies using sp_iqcheckdb, shows the DBCC output when index problems are detected, and describes the DBCC errors related to index problems.

Sample of output with inconsistent index

The following is an example of the type of output you see when you run sp_iqcheckdb and there is index inconsistency. DBCC displays both a summary and details about the indexes checked. The Index Summary section at the top of the report indicates if any inconsistent indexes were found. The names of the inconsistent indexes and the type(s) of problems can be found in the index statistics section. The lines with asterisks (*****) contain information about inconsistent indexes.

Extra, missing, or duplicate RID errors are the most common types of errors reported. These errors are an indication that the index is misrepresentative of the data and may give incorrect results or cause other failures. These errors are generally accompanied by other errors indicating the specifics of the inconsistencies.

In this example, DBCC reports an inconsistent HNG index. DBCC repairs indexes using data from the FP indexes. Since the corresponding FP index checks are good, the FP index can be used to repair the damaged HNG index.

The command line executed for this example is `sp_iqcheckdb 'verify database'`.
Note that DBCC produces a detailed report, but some lines of the output have been removed in this example.

```
              Stat                        Value                   Flags
==============================|==================================|=====
DBCC Verify Mode Report       |                                  |
==============================|==================================|=====
** DBCC Status                |Errors Detected                   |*****
   DBCC Work units Dispatched |75                                |
   DBCC Work units Completed  |75                                |
==============================|==================================|=====
Index Summary                 |                                  |
==============================|==================================|=====
** Inconsistent Index Count   |1                                 |*****
   Verified Index Count       |85                                |
==============================|==================================|=====
Index Statistics              |                                  |
==============================|==================================|=====
** Inconsistent Index         |contact.DBA.idx01_HNG             |*****
   ...
   Verified Index             |fin_data.DBA.ASIQ_IDX_T209_C3_HG  |
   Verified Index             |fin_data.DBA.ASIQ_IDX_T209_C4_FP  |
   ...
   Verified Index             |employee.DBA.ASIQ_IDX_T212_C19_FP |
   Verified Index             |employee.DBA.ASIQ_IDX_T212_C20_FP |
   Verified Index             |iq_dummy.DBA.ASIQ_IDX_T213_C1_FP  |
** Extra Index RIDs           |5                                 |*****
   FP Indexes Checked         |68                                |
   HNG Indexes Checked        |1                                 |
   HG Indexes Checked         |17                                |
                              |                                  |
```

The inconsistent index detected by sp_iqcheckdb is contact.DBA.idx01_HNG.

The following DBCC output is generated when sp_iqcheckdb is run again to check just the inconsistent index. The command line executed for this example is `sp_iqcheckdb 'verify index DBA.contact.idx01_HNG'`.

```
             Stat                          Value                  Flags
==============================|==================================|=====
DBCC Verify Mode Report       |                                  |
==============================|==================================|=====
** DBCC Status                |Errors Detected                   |*****
   DBCC Work units Dispatched |1                                 |
   DBCC Work units Completed  |1                                 |
                              |                                  |
==============================|==================================|=====
Index Summary                 |                                  |
==============================|==================================|=====
** Inconsistent Index Count   |1                                 |*****
   Verified Index Count       |1                                 |
                              |                                  |
==============================|==================================|=====
Index Statistics              |                                  |
==============================|==================================|=====
** Inconsistent Index         |contact.DBA.idx01_HNG             |*****
   Verified Index             |contact.DBA.ASIQ_IDX_T206_C1_FP   |
** Extra Index RIDs           |5                                 |*****
   FP Indexes Checked         |1                                 |
   HNG Indexes Checked        |1                                 |
                              |                                  |
==============================|==================================|=====
```

DBCC index errors

Messages in the DBCC output related to problems with indexes are listed in the following table. See the section "DBCC error messages" on page 83 for a more extensive list of DBCC messages.

Table 2-6: DBCC index errors

DBCC message	Description/action
Inconsistent Index Count	The number of indexes that DBCC found to have inconsistencies.
Inconsistent Index	The name of an index that DBCC found to be inconsistent.
Extra Index RIDs Missing Index RIDs Duplicate Index RIDs	The total number of rows that are inconsistent for all inconsistent indexes.
Bitmap Verify Errors	The total number of inconsistent bitmaps in all database objects
FP Lookup Table Inconsistencies	An unrepairable error, where the 1-byte or 2-byte FP is internally inconsistent.
Non-Completed Index Count	The number of indexes that could not be verified, because an exception occurred while checking.
Non-Completed Index	The name of an index that was not verified because an exception occurred while checking. If the exception is a future version, out of memory, or out of buffers error, commit the DBCC connection and re-run DBCC.
VDO Incorrect First Available Fields VDO Incorrect Next Available Fields VDO Incorrect Used Count Fields VDO Incorrect In-use Bitvec VDO Incorrect In-use Bitmap VDO Incorrect Partial Bitmap VDO Incorrect Deleted Bitmaps	Unrepairable errors that can cause entire tables to be inaccessible. You must force drop the inconsistent table to resolve these errors.
HG Missing Groups HG Extra Groups HG Extra Keys HG Missing Keys B-Tree Invalid Item Count B-Tree Invalid Item Count G-Array Empty Page Errors G-Array Bad Group Type Errors G-Array Out of Order Group Errors	High Group index specific errors.

Repairing index errors

In repair mode, sp_iqcheckdb performs detailed index checks and can repair many types of index inconsistencies. The DBCC output indicates which indexes, if any, were repaired. After DBCC repairs an index, you should run sp_iqcheckdb again in check mode before committing any changes. If an index is still inconsistent, drop and recreate the index as described in the section "Dropping inconsistent indexes, tables, or columns" on page 81, and then rebuild the index.

❖ Repairing index problems using DBCC

1. Run sp_iqcheckdb in repair mode to verify and repair the specified targets. For example, to check and repair the indexes on the table t1:

   ```
   sp_iqcheckdb 'repair table t1';
   ```

2. Run the stored procedure sp_iqcheckdb again in verify mode. For example, to verify the indexes on table t1 again:

   ```
   sp_iqcheckdb 'verify table t1';
   ```

3. When sp_iqcheckdb completes, review the report for the repairs made to the database. You must issue a COMMIT command in order to commit the changes to the database. Commit the changes *only* if the second sp_iqcheckdb reports no errors. If an index is still inconsistent, drop and recreate the index.

DBCC index repair output

DBCC displays an Index Summary section at the top of the report, which lists the number of repaired and verified indexes. The Index Statistics section provides details about the indexes repaired. Lines containing information about repaired indexes are flagged with asterisks (*****). See the section "DBCC index errors" on page 65 for more information on DBCC output messages related to indexes.

In this example, DBCC repaired the inconsistent HNG index. The command line executed for this example is `sp_iqcheckdb 'repair database'`.

Note DBCC repair mode does verify all indexes, but does not check allocation or report allocation statistics. Some lines of the output have been removed in this example.

```
              Stat                           Value                  Flags
==============================|===================================|=====
DBCC Repair Mode Report       |                                   |
==============================|===================================|=====
** DBCC Status                |Errors Detected                    |
   DBCC Work units Dispatched |75                                 |
   DBCC Work units Completed  |75                                 |
==============================|===================================|=====
Index Summary                 |                                   |
==============================|===================================|=====
** Repaired Index Count       |1                                  |*****
   Verified Index Count       |85                                 |
==============================|===================================|=====
Index Statistics              |                                   |
==============================|===================================|=====
** Repaired Index             |contact.DBA.idx01_HNG              |*****
   ...
   Verified Index             |employee.DBA.ASIQ_IDX_T212_C19_FP  |
   Verified Index             |employee.DBA.ASIQ_IDX_T212_C20_FP  |
   Verified Index             |iq_dummy.DBA.ASIQ_IDX_T213_C1_FP   |
** Extra Index RIDs Repaired  |5                                  |*****
   FP Indexes Checked         |68                                 |
   HNG Indexes Checked        |1                                  |
   HG Indexes Checked         |17                                 |
                              |                                   |
==============================|===================================|=====
```

Recreating FP indexes

If an FP index is inconsistent, you may be able to rebuild the FP index from an HG or LF index on the same column. Recreate the FP index only if you are sure that the FP index is damaged (i.e., read errors).

If you specify an FP index in DBCC repair mode and a LF or HG index is present on the corresponding FP column, the FP index is recreated from the contents of that LF or HG index. Note that the FP index appears twice in the DBCC results, since the FP index verifies against the LF or HG after it is repaired.

When you recreate FP indexes, space for the original FP index is intentionally leaked. Use sp_iqcheckdb with the -iqdroplks server option to recover this space, as described in the section “Repairing allocation problems” on page 71.

```
         Stat                          Value                   Flags
===========================|===============================|=====
DBCC Repair Mode Report    |                               |
===========================|===============================|=====
...
===========================|===============================|=====
Index Statistics           |                               |
===========================|===============================|=====
** Repaired Index          |t1.DBA.ASIQ_IDX_T185_C1_FP     |*****
   Verified Index          |t1.DBA.ASIQ_IDX_T185_C1_FP     |
   Verified Index          |t1.DBA.t1c1hg                  |
   FP Indexes Checked      |1                              |
   HG Indexes Checked      |1                              |
```

DBCC index repair messages

Messages in the DBCC output related to repaired indexes are listed in the following table. See the section "DBCC error messages" on page 83 for a more extensive list of DBCC messages.

Table 2-7: DBCC index repair messages

DBCC message	Description/action
Repaired Index Count	The number of indexes that DBCC attempted to repair.
Repaired Index	The name of an index that DBCC has attempted to repair.
Extra Index RIDs Repaired Missing Index RIDs Repaired Duplicate Index RIDs Repaired	The total number of rows repaired for all inconsistent indexes.
HG Missing Groups Repaired HG Extra Groups Repaired HG Extra Keys Repaired HG Missing Keys Repaired	High Group index specific repair messages.

Time required to run DBCC repair

The execution time of DBCC repair depends on the number of errors detected. If the number of errors reported is large, Sybase recommends dropping and recreating the index.

Analyzing allocation problems

This section describes how to analyze allocation problems using sp_iqcheckdb, shows the DBCC output when allocation problems are detected, and describes the DBCC errors related to allocation problems.

The database maintains an allocation map, also known as a free list, which tracks the blocks that are in use by database objects. DBCC detects three types of allocation problems:

leaked blocks A leaked block is a block that is allocated according to the database allocation map, but is found not to be part of any database objects. DBCC can recover leaked blocks.

unallocated blocks An unallocated block is a block that is not allocated according to the database allocation map, but is found to be in use by a database object. DBCC can recover unallocated blocks.

multiply-owned blocks A multiply-owned block is a block that is in use by more than one database object. At least one of the structures involved contains inconsistent data. DBCC *cannot* repair this type of allocation problem. If you encounter this type of error, run DBCC again, specifying a list of indexes, until you identify the indexes that share the block. These indexes must then all be dropped to eliminate the multiply-owned block. See the section “Dropping inconsistent indexes, tables, or columns” on page 81 for more information on dropping inconsistent indexes.

Sample of leaked space output

The following is an example of the output you see when you run sp_iqcheckdb and there is leaked space. Lines with asterisks (*****) contain information about allocation problems. In this example, DBCC reports 16 leaked blocks.

The command line executed for this example is `sp_iqcheckdb 'allocation database'`.

```
          Stat                                Value               Flags
==============================|==============================|=====
DBCC Allocation Mode Report   |                              |
==============================|==============================|=====
** DBCC Status                |Errors Detected               |*****
   DBCC Work units Dispatched |164                           |
   DBCC Work units Completed  |164                           |
                              |                              |
==============================|==============================|=====
Allocation Summary            |                              |
==============================|==============================|=====
   Blocks Total               |8192                          |
   Blocks in Current Version  |4785                          |
   Blocks in All Versions     |4785                          |
   Blocks in Use              |4801                          |
   % Blocks in Use            |58                            |
** Blocks Leaked              |16                            |*****
                              |                              |
==============================|==============================|=====
Allocation Statistics         |                              |
==============================|==============================|=====
   ...
** 1st Unowned PBN            |1994                          |*****
   ...
==============================|==============================|=====
```

DBCC allocation errors

Allocation problems are reported in the output generated by DBCC with sp_iqcheckdb run in a allocation mode or verification mode. If the Allocation Summary section has values flagged with asterisks, such as "** Blocks Leaked" or "** Blocks with Multiple Owners," then there are allocation problems.

Messages in the DBCC output related to allocation problems are listed in the following table. See the section "DBCC error messages" on page 83 for a more extensive list of DBCC messages.

Table 2-8: DBCC allocation errors

DBCC message	Description/action
Block Count Mismatch	This count always accompanies other allocation errors.
Blocks Leaked 1st Unowned PBN	Blocks that were found not to be in use by any database object. Use -iqdroplks to repair.
Blocks with Multiple Owners 1st Multiple Owner PBN	Blocks in use by more than one database object. Drop the object that is reported as inconsistent.
Unallocated Blocks in Use 1st Unallocated PBN	Blocks in use by a database object, but not marked as in use. Use -iqdroplks to repair.

If the Allocation Summary lines indicate no problem, but the Index Summary section reports a value for "Inconsistent Index Count," then this indicates one or more inconsistent indexes. See the section "Repairing index errors" on page 66 for information on repairing indexes.

Repairing allocation problems

The following procedure uses sp_iqcheckdb, in conjunction with the -iqdroplks server startup option, to repair database allocation problems. Note that sp_iqcheckdb 'allocation database' is the only DBCC command allowed with the -iqdroplks switch.

Note The following procedure uses the -gd and -gm switches to restrict database access. For a more restrictive method, see "Restricting database access during recovery" on page 74.

❖ Repairing allocation problems using DBCC:

1 Start the server with the -iqdroplks option and database name in the start_asiq command line. For example:

```
start_asiq -n my_db_server -x 'tcpip{port=7934}'
-gd dba -gm 1 -iqdroplks my_db
/work/database/my_db.db
```

You specify the database name twice, first as a required option to -iqdroplks with no path (to specify the database in which to recover the leaked space), and again to specify the database to start.

Note You must start the database with the ".db" extension, not ".DB".

Sybase recommends using two server startup switches to restrict access:

- Use -gd DBA so that only users with DBA authority can start and stop databases. (Note that the client must already have a connection to the server to start or stop the database, so this switch does not prevent connections.)
- Use -gm 1 to allow a single connection plus one DBA connection above the limit so that a DBA can connect and drop others in an emergency.

For more information about restricting connections, see "Restricting database access during recovery" on page 74.

2 Run the stored procedure sp_iqcheckdb in allocation mode:

```
sp_iqcheckdb 'allocation database'
```

If the allocation repair is successful, sp_iqcheckdb displays the message "Freelist Updated." If errors are detected, sp_iqcheckdb returns the messages "Freelist Not Updated" and "Errors Detected."

3 Stop the server after sp_iqcheckdb finishes. To stop the server, use stop_asiq on any platform or the shutdown button in the console window on Windows.

4 Restart the server using your usual method, without the -iqdroplks option.

After allocation problems are repaired, allocation statistics appear in the DBCC output with no errors. If allocation statistics are not in the DBCC output, check to be sure that sp_iqcheckdb was run in repair mode. If errors other than multiply owned block errors are in the DBCC output, check that the -iqdroplks server startup option was specified with the correct database name in the server command line options.

DBCC allocation repair output

DBCC displays an Allocation Summary section at the top of the report, which lists information about allocation usage. The Allocation Statistics section provides more details about the blocks. The DBCC output does not contain repair messages for the leaked blocks that have been recovered.

In this example, the server is started using the -iqdroplks option. The following commands are executed:

```
sp_iqcheckdb 'allocation database';
checkpoint;
```

The sp_iqcheckdb output indicates no errors, so the checkpoint is executed.

Note that since DBCC check mode is used in this example to repair the allocation problems, all of the indexes are also checked for consistency. DBCC reports statistics that do not show in this abbreviated output.

```
          Stat                              Value                    Flags
==============================|===================================|=====
DBCC Allocation Mode Report   |                                   |
==============================|===================================|=====
   DBCC Status                |Freelist Updated                   |
   DBCC Status                |No Errors Detected                 |
   DBCC Work units Dispatched |75                                 |
   DBCC Work units Completed  |75                                 |
==============================|===================================|=====
Allocation Summary            |                                   |
==============================|===================================|=====
   Blocks Total               |8192                               |
   Blocks in Current Version  |4594                               |
   Blocks in All Versions     |4594                               |
   Blocks in Use              |4610                               |
   % Blocks in Use            |56                                 |
==============================|===================================|=====
Allocation Statistics         |                                   |
==============================|====================================|=====
   DB Extent Count            |1                                   |
   Marked Logical Blocks      |8176                                |
   Marked Physical Blocks     |4594                                |
   Marked Pages               |511                                 |
   Blocks in Freelist         |126177                              |
   Imaginary Blocks           |121567                              |
   Highest PBN in Use         |5425                                |
   Total Free Blocks          |3582                                |
   Usable Free Blocks         |3507                                |
   % Free Space Fragmented    |2                                   |
   Max Blocks Per Page        |16                                  |
   1  Block Page Count        |103                                 |
   3  Block Page Count        |153                                 |
   ...
   16 Block Hole Count        |213                                 |
==============================|====================================|=====
```

Database startup after recovery

When performing forced recovery or leaked blocks recovery, you must start the database with the ".db" extension, not ".DB". For example:

```
 start_asiq -n my_db_server -x 'tcpip{port=7934}'
-gd dba -iqdroplks my_db /work/database/my_db.db
```

Forced recovery mode

Sybase IQ helps ensure that your server can be started even with inconsistent recovery information by providing a special forced recovery mode. You can use forced recovery mode to display corruption information and then to repair the database.

These procedures are described in the following sections of this chapter:

- Starting servers in forced recovery mode
- Recovering leaked space
- Recovering multiplex databases

Forced database recovery differs from normal database recovery in these ways:

- **Forced recovery marks all storage within the database as in use.** In order to recover a potentially inconsistent allocation map, all storage within the database is marked as in use. You can use the server startup option -iqdroplks with the sp_iqcheckdb stored procedure to reset the allocation map to the correct state. For more details, see "Recovering leaked space using forced recovery" on page 77.
- **Incremental backups are disabled.** After the database is opened in forced recovery mode, incremental backups are disabled. The next backup must be a full backup. Doing a full backup reenables incremental backups.
- **The forced recovery parameter applies to all opens of the database while the server is up.** Therefore, after the database is opened, the DBA needs to bring the server back down, and then restart the server without the forced recovery flag, to be sure that subsequent opens run in regular mode. Repeated opens of the database with forced recovery on do not harm the database, but could be confusing to the DBA. Each time you open the database in forced recovery mode, all the storage within the database is marked as in use.

Starting servers in forced recovery mode

If a server fails to start with an exception or an assert when opening a database, start the server with forced recovery. Forced recovery allows the server to start if the allocation map or checkpoint information is inconsistent. In this mode, options display information about inconsistencies. You can also specify options to repair such inconsistencies.

Note *Use forced recovery only when normal database recovery fails to restore the database to a running state.*

Restricting database access during recovery

Restricting access gives the DBA greater control over inadvertent opens of the database during forced recovery. Sybase recommends using two server startup switches to restrict access:

- Use -gd DBA so that only users with DBA authority can start and stop databases on a running server. (Note that the client must already have a connection to the server to start or stop the database, so this switch does not prevent connections.)
- Use -gm 1 to allow a single connection plus one DBA connection above the limit so that a DBA can connect and drop others in an emergency.

An alternate way to restrict connections is to specify

```
sa_server_option('disable_connections', 'ON')
```

just after you start the connection where you are performing forced recovery and

```
sa_server_option('disable_connections', 'OFF')
```

on the same connection after recovery. *The disadvantage is that this method precludes emergency access from another DBA connection.*

❖ Starting a server in forced recovery mode

1 Start the server with forced recovery (to mark all pages as used), using the -iqfrec server startup option in the start_asiq command. For example:

```
start_asiq -n my_server -x 'tcpip(port=7934}'
-gd dba -gm 1 -iqfrec my_db /database/my_db.db
```

Forced recovery starts the server in single-node mode. Stop all query servers first.

Warning! You must specify the override startup switch (-iqmpx_ov 1) and start in single node mode (-iqmpx_sn 1) when starting a multiplex write server after any failure. Never use multiplex mode (the default) for recovery.

You specify the database name twice, once to specify the database undergoing forced recovery and once to specify the database to start. The -iqfrec server startup option requires the database name. Note that this is the *physical* database name, which is case sensitive. Do not use select_dbname to determine the database name, as it returns the logical name assigned by the -n startup option.

2 If desired, you can run sp_iqcheckdb to check for leaked blocks. For details, see “Analyzing allocation problems” on page 69.

3 Stop the server after it has started successfully. To stop the server, use stop_asiq on UNIX or the shutdown button in the console window on Windows.

4 Restart the server using your usual method, without the -iqfrec and -iqdroplks options.

If you are unable to start your server in forced recovery mode, contact Sybase Technical Support.

Using forced recovery without a follow on sp_iqcheckdb

Running forced recovery starts the database in a valid, but fully allocated mode. In other words, you should be able to do all operations, but no permanent main dbspace is left. Before you do anything else, you must either recover the lost dbspace by running sp_iqcheckdb with the -iqdroplks flag, or add a new dbspace. Note that queries should also run successfully, since they do not need additional permanent dbspace; however, you cannot load, insert, or delete data.

Warning! Running queries without verifying the database will not cause any inconsistency in your data. However, if there is a problem in the data that caused the server to fail, the server could fail again or produce incorrect results.

See the section “Recovering leaked space” for details on using sp_iqcheckdb to reclaim lost or leaked space.

Recovering leaked space

An allocation map is used by the server to determine if a page is in use or not in use within IQ. Either through system failure or as a result of opening a database with forced recovery, a database's allocation map may not reflect the true allocation of its usage. When this occurs, we say that the database has “leaked” storage or “leaked blocks.” In general, you need not be concerned about small numbers of leaked blocks. If you have many megabytes of leaked blocks, you probably want to recover that space.

Sybase IQ lets you recover leaked storage using the -iqdroplks server startup option in conjunction with the sp_iqcheckdb stored procedure. The -iqdroplks option allows sp_iqcheckdb to recover leaked storage space within the specified database.

When leaked storage is being recovered, other transactions that alter the allocation map are shut out. Such operations include checkpoints and commands that modify the database.

You can recover leaked storage and force recovery either at the same time or separately. To recover leaked space within a database without doing a forced recovery, follow the procedure in the section “Repairing allocation problems” on page 71. To recover leaked space within a database after doing a forced recovery, follow the procedure in the next section “Recovering leaked space using forced recovery”.

Recovering leaked space using forced recovery

If the procedure in the section “Repairing allocation problems” on page 71 fails to recover leaked storage, then use the following procedure to do so.

Note The following procedure uses the -gd and -gm switches to restrict database access. For a more restrictive method, see “Restricting database access during recovery” on page 74.

❖ Recovering leaked space with forced recovery

1 Start the server with the `-iqdroplks` and `-iqfrec` options in the start_asiq command. For example:

```
start_asiq -n my_db_server -x 'tcpip{port=7934}'
-gd dba -gm 1 -iqdroplks my_db
-iqfrec my_db /work/database/my_db.db
```

You specify the database name twice in a row, once to specify it as the database you are starting, and once to specify it as the database undergoing forced recovery. Both the -iqfrec and -iqdroplks options require the database name.

2 Connect to the database you are recovering.

3 Run the stored procedure sp_iqcheckdb in allocation mode:

```
sp_iqcheckdb 'allocation database'
```

If there are no errors and sp_iqcheckdb displays the message “Freelist Updated,” you have recovered leaked space and forced recovery. Continue to the next step.

If inconsistency is found, follow the instructions in the section “Dropping inconsistent indexes, tables, or columns” to drop inconsistent objects. Then run sp_iqcheckdb again to recover leaked space.

4 Issue a checkpoint.

5 Stop the server using your usual method.

6 Restart the server using your usual method, and proceed with normal processing.

Recovering multiplex databases

Before troubleshooting recovery problems with a multiplex database, see "Multiplex server migration and failover" in *Sybase IQ System Administration Guide*.

Single-node mode

Query servers continue to run while the write server issues DDL operations. The table version log (TLV log) stores information about DDL operations and communicates information about new table versions to the query servers. The write server preserves older table versions for as long as needed. In single-node mode, no version logging takes place. Query servers cannot run and must be synchronized to restart.

The -iqmpx_sn 1 server startup option starts the server in single-node mode.

Warning! You must specify the override startup switch (-iqmpx_ov 1) and start in single-node mode (-iqmpx_sn 1) when starting a multiplex write server after any failure. Never use multiplex mode (the default) for recovery.

If a server runs out of space during a checkpoint operation, try restarting in single-node mode. For example, if a server ran out of space and was killed, it could precipitate a rare situation where normal recovery failed. Suppose that many versions were created at the server and they were all retained because the query server used an older version or there was a dbremote delay. If the write server had too many recovered transactions to checkpoint, it could run out of space during the recovery checkpoint. Subsequently, users could not connect or add dbspaces. Starting in single-node mode deallocates retained versions and allows checkpoints to succeed.

Replacing write servers

If the write server of a multiplex fails in such a way that its SYSTEM dbspace and transaction log files cannot be recovered, you can use forced recovery and the sp_iqcheckdb option resetclocks as part of the process to convert one of the query servers to a write server. For the complete server conversion procedure, see the section "Replacing write servers" in Chapter 14, "Data Backup, Recovery, and Archiving" of the *Sybase IQ System Administration Guide*.

Emergency recovery without a transaction log

Ordinarily, you should follow the recovery procedures discussed in the previous sections of this chapter.

In rare situations, you may need to use the emergency recovery procedure, if all of the following conditions exist:

- No backup exists
- The transaction log has been lost
- There is no mirror log
- There is insufficient time to work with Sybase Engineering to develop alternative options

Under these conditions, you can restart the server with the -f option.

-f recovery option

Function

Force the database server to start after the transaction log has been lost.

Syntax

start_asiq -n *server-name* [*other-server-options*] -f

Description

If there is a transaction log in the same directory as the database, the database server carries out a checkpoint recovery of the Catalog, and a recovery using the transaction log, and then terminates—it does not continue to run. You can then restart the database server without the -f option for normal operation.

If there is no transaction log, the database server carries out a checkpoint recovery of the database and then terminates—it does not continue to run. You can then restart the database server without the -f option for normal operation.

Warning! While using the -f option can usually bring the server back online, it also very frequently results in corruption of the database, because it bypasses replay of transactions. The resulting corruption may not be encountered until a later time and usually cannot be repaired. This procedure is highly risky and is not recommended except in extreme cases. You may also need to do a forced recovery (-iqfrec) to reopen the database.

Example

```
start_asiq -n bad_server -x 'tcpip(port=7934}'
-gd dba -f
```

Handling unrepairable problems

In some cases, the sp_iqcheckdb stored procedure does not have access to sufficient valid information to repair consistency problems with indexes, tables, or columns. This section describes the actions you can take to resolve this type of problem.

The following table lists DBCC output messages that indicate problems that DBCC cannot repair. See the section “DBCC error messages” on page 83 for a more extensive list of DBCC messages.

Table 2-9: Messages for problems DBCC cannot repair

DBCC message	Description/action
FP Lookup Table Inconsistencies	An unrepairable error, where the 1-byte or 2-byte FP is internally inconsistent.
VDO Incorrect First Available Fields VDO Incorrect Next Available Fields VDO Incorrect Used Count Fields VDO Incorrect In-use Bitvec VDO Incorrect In-use Bitmap VDO Incorrect Partial Bitmap VDO Incorrect Deleted Bitmaps	Unrepairable errors that can cause entire tables to be inaccessible. You must force drop the inconsistent table to resolve these errors.
Blocks with Multiple Owners 1st Multiple Owner PBN	Blocks in use by more than one database object. Drop the object that is reported as inconsistent.
DBCC Meta-data Errors Blockmap Invalid Chunksize Error Count Blockmap Compression Bit Error Count Blockmap Invalid Block Number Error Count	An internal page mapping structure is inconsistent and the object needs to be dropped.
DBCC Inconsistent Disk Block Headers DBCC Decompress Errors	The storage for the object is inconsistent and the object needs to be dropped.

See the following sections for information on resolving these unrepairable problems.

Index problems that cannot be repaired

If DBCC detects a problem with an index, the name of the index is reported with the type of problem. Some problems can be repaired, while others can only be resolved by dropping an index, column, or table. If the DBCC output contains the errors "Inconsistent Index" or "Missing Index RIDs," DBCC may be able to fix the index. See the section "Analyzing index errors" on page 63 for the procedure to follow for indexes reported as "Inconsistent Index," when sp_iqcheckdb is run in default or check mode.

If an index is reported as "Inconsistent Index" after you run sp_iqcheckdb in repair mode to repair the index, then this index cannot be fixed. Depending on the type of problem, use DROP INDEX, ALTER TABLE DROP COLUMN, DROP TABLE, or the FORCE_DROP option to resolve the problem. See the section "Dropping inconsistent indexes, tables, or columns" on page 81.

Sybase recommends calling Sybase Technical Support for help in determining the best course of action to fix an inconsistent index or table.

Dropping inconsistent indexes, tables, or columns

If sp_iqcheckdb reports unrepairable indexes, columns, or tables, then these objects must be dropped using the DROP INDEX, ALTER TABLE DROP COLUMN, or DROP TABLE statements respectively.

Note You should not attempt to force drop objects unless Sybase Technical Support has instructed you to do so.

If you cannot drop an inconsistent object, set the temporary FORCE_DROP option. FORCE_DROP causes the IQ server to silently leak the on-disk storage of the dropped object, rather than try to reclaim it. You can recover the leaked space later using DBCC. This is desirable for an inconsistent object, because the only information about the storage of an object is within the object itself, and this information is suspect for an inconsistent object.

If FORCE_DROP is set to 'ON', you cannot drop a join index on a multiplex write server. To force drop a join index on a multiplex write server, you must first start the server in single-node mode, and after the drop, restart only the write server.

Note When force dropping objects, you must ensure that only the DBA is connected to the database. Restart the server immediately after a force drop.

The following procedure uses the -gd and -gm switches to restrict database access. The -gd switch only limits users who can start or stop databases on a running server. For a more restrictive method, see "Restricting database access during recovery" on page 74.

❖ Dropping inconsistent objects

1 Restart the server.

```
start_asiq -n bad_db_server -x 'tcpip{port=7934}'
-gm 1 -gd dba bad_db.db
```

You must not allow other users to connect when force dropping objects.

Sybase recommends using two server startup switches to restrict access:

- Use -gd DBA so that only users with DBA authority can start and stop databases. (Note that the client must already have a connection to the server to start or stop the database, so this switch does not prevent connections.)
- Use -gm 1 to allow a single connection plus one DBA connection above the limit so that a DBA can connect and drop others in an emergency.

For more information about restricting connections, see *Sybase IQ Installation and Configuration Guide*.

2 Set the temporary option FORCE_DROP to ON.

```
set temporary option FORCE_DROP = 'ON'
```

3 Drop all inconsistent objects.

Use the commands DROP INDEX, ALTER TABLE DROP COLUMN, or DROP TABLE as needed. *Do not enter any other DDL or DML commands until after restarting the server.*

4 Restart the server.

To recover the leaked space and update the allocation map to the correct state, start the server with the -iqdroplks option set. You can set this option now, as shown in this example, or whenever you restart the server.

```
start_asiq -n bad_db_server -x 'tcpip{port=7934}'
-iqdroplks bad_db bad_db.db
```

You specify the database name twice, once to specify which database you are starting (with no path), and once to identify the database for the -iqdroplks option (pathname optional).

5 Run sp_iqcheckdb.

```
sp_iqcheckdb 'allocation database';
```

This step resets the database allocation map to the calculated allocation map.

For more information, see the sections “Recovering leaked space” on page 76 and “Database verification” on page 52.

DBCC error messages

The following table lists the most important messages in the DBCC output.

Table 2-10: DBCC error messages

DBCC message	Description/action
Inconsistent Index Count	The number of indexes that DBCC found to have inconsistencies.
Inconsistent Index	The name of an index that DBCC found to be inconsistent.
Repaired Index Count	The number of indexes that DBCC attempted to repair.
Repaired Index	The name of an index that DBCC has attempted to repair.
Extra Index RIDs Repaired Missing Index RIDs Repaired Duplicate Index RIDs Repaired Extra Index RIDs Missing Index RIDs Duplicate Index RIDs	The total number of rows that are inconsistent for all inconsistent indexes.
Bitmap Verify Errors	The total number of inconsistent bitmaps in all database objects.
FP Lookup Table Inconsistencies	An unrepairable error, where the 1-byte or 2-byte FP is internally inconsistent.

DBCC message	Description/action
Non-Completed Index Count	The number of indexes that could not be verified, because an exception occurred while checking.
Non-Completed Index	The name of an index that was not verified because an exception occurred while checking. If the exception is a future version, out of memory, or out of buffers error, commit the DBCC connection and re-run DBCC.
HG Missing Groups HG Missing Groups Repaired HG Extra Groups HG Extra Groups Repaired HG Extra Keys HG Extra Keys Repaired HG Missing Keys HG Missing Keys Repaired B-Tree Invalid Item Count B-Tree Invalid Item Count G-Array Empty Page Errors G-Array Bad Group Type Errors G-Array Out of Order Group Errors	High Group index specific errors and repair messages.
VDO Incorrect First Available Fields VDO Incorrect Next Available Fields VDO Incorrect Used Count Fields VDO Incorrect In-use Bitvec VDO Incorrect In-use Bitmap VDO Incorrect Partial Bitmap VDO Incorrect Deleted Bitmaps	Unrepairable errors that can cause entire tables to be inaccessible. You must force drop the inconsistent table to resolve these errors.
Block Count Mismatch	This count accompanies other allocation errors.
Blocks Leaked 1st Unowned PBN	Blocks that were found not to be in use by any database object. Use -iqdroplks to repair.
Blocks with Multiple Owners 1st Multiple Owner PBN	Blocks in use by more than one database object. Drop the object that is reported as inconsistent.
Unallocated Blocks in Use 1st Unallocated PBN	Blocks in use by a database object, but not marked as in use. Use -iqdroplks to repair.
Freelist Updated	Indicates successful allocation repair.
Freelist Not Updated	Indicates errors detected during allocation repair and the allocation repair was not successful.
Invalid Blockmap Unique ID Generator Blockmap Unique ID Generator Updated Invalid Transaction ID Counter Transaction ID Generator Updated	Errors and repair messages specific to the DBCC resetclocks option.

DBCC message	Description/action
DBCC Future Version Errors	DBCC could not open the table, because DDL was performed on it. Commit the DBCC connection and re-run DBCC.
DBCC Out of Buffers Errors	The size of the IQ main cache is too small. Either increase the main cache size or run DBCC on individual objects.
DBCC Out of Memory Errors	There is insufficient system memory to complete the DBCC operation.
DBCC Meta-data Errors Blockmap Invalid Chunksize Error Count Blockmap Compression Bit Error Count Blockmap Invalid Block Number Error Count	An internal page mapping structure is inconsistent and the object needs to be dropped.
DBCC Page Read Errors	An I/O error occurred while trying to read an object. Perform hardware diagnostics.
DBCC Inconsistent Disk Block Headers DBCC Decompress Errors	The storage for the object is inconsistent and the object needs to be dropped.
DBCC Unknown Exceptions	An exception of a type unknown to DBCC occurred. Check the IQ message file for details.

Index

A

allocation
- DBCC repair output 72
- verifying and repairing 69

allocation map
- checking allocation 56
- fixing errors 70
- inconsistencies 74
- recovery 73

B

blocked write access
- determining blocking writers 29
- managing contention 30

blockmap 56

buffer cache
- insufficient space 9
- IQ UTILITIES command 11
- monitor 11

bugs
- reporting 44

C

catalog files
- growth 22

checklist for Technical Support 50

checkpoints
- adjusting interval 31

columns
- unrepairable errors 80

communications
- troubleshooting 32

configuration parameters
- overriding 8

connection information
- IQ message file 47

connections
- restricting 75

consistency checking
- multiplex 53

conventions
- documentation x, xi
- syntax x
- typographic xi

D

data conversion error 18

data link layer
- troubleshooting 34

data truncation error 18

database
- repair 2, 51

databases
- sample xi

DBCC
- allocation verification and repair 69
- analyzing allocation problems 69
- analyzing index problems 63
- checking allocation 56
- checking indexes and allocation 55
- database verification 52
- detecting allocation errors 70
- detecting index problems 81
- index verification and repair 63
- internal index checking 55
- output 58
- output messages 83
- performance 54
- repairing allocation 69, 71
- repairing indexes 63, 66
- sample output 58
- sp_iqcheckdb interface 53
- time to run 54

DBCC_LOG_PROGRESS option 58
DBISQL Java 18
dbspace count differs 17
deadlock
  detecting 11
  resolving 11
diagnostic tools 38
  checking database options 40
  checking server startup options 40
  communications issues 44
  logging server requests 41
  sa_server_option 41
  sp_iqcheckdb 40
  sp_iqcheckoptions 40
  sp_iqconnection 41
  sp_iqcontext 41
  sp_iqdbstatistics 40
  sp_iqstatus 38
disk
  monitoring space usage 23
  out of space 9, 19
documentation
  accessibility features xi
  Adaptive Server Anywhere vii
  conventions x, xi
  on CD viii
  online viii
  Sybase IQ v
driver, missing 15
DumpAllThreads file 10
dumpdups
  sp_iqcheckdb option 57
dumpleaks
  sp_iqcheckdb option 57
dumpunallocs
  sp_iqcheckdb option 57
dynamic result sets 18

E

errors
  unrepairable 80
Ethernet 37
events
  monitoring disk space usage 23

F

Federal Rehabilitation Act
  section 508 xi
file size
  controlling 22
FORCE_DROP option 81
forced recovery 74
  detecting duplicate blocks 57
  detecting multiply owned blocks 57
  detecting unallocated blocks 57
  procedure 77
  replacing a write server 57, 78
  server startup failure 74
frame type 37

G

getiqinfo script 44
gm switch
  effect on recovery 52

H

HASH_THRASHING_PERCENT option 27

I

iAnywhere JDBC driver 18
inconsistent state 74
indexes
  detecting logical problems 81
  dropping corrupt 81
  maximum unique values 29
  recreating FP indexes 67
  repair utility report 66
  sp_iqcheckdb errors 70, 81
  too many on table 27
  unrepairable errors 80
  verifying and repairing 63
IP address
  ping 34
IQ UTILITIES
  buffer cache monitor 11

L

LANalyzer 36
leaked space recovery 76
LF index
  exceeding maximum unique values 29
loading data
  errors 29
  monitoring space usage 23
  performance 27
locks
  managing contention 30
log files
  correlating connection information 47

M

message file
  connection information 47
Microsoft Access 27
multiplex
  consistency checks 53
MySybase
  accessing 49
  online support 49

N

NDIS
  drivers 33
net.cfg file 37
NetBIOS
  troubleshooting 34
NetWare
  network adapter settings 37
network adapters
  drivers 33
network protocols
  troubleshooting 32
Novell client software 33

O

ODI drivers 33
options
  DBCC_LOG_PROGRESS 58
out of disk space
  monitoring space usage 23
  recommended actions 9, 19

P

performance
  queries and loads 27
physical layer
  troubleshooting 36
ping
  TCP/IP 34
problems
  reporting 44
procedures
  dynamic result sets 18
product support 44
protocols
  troubleshooting 32

Q

queries
  performance issues 27
  thrashing 27
query server
  replacing a write server 57, 78
query server mismatch 17

R

recovery
  database repair 53
  database verification 53
  forced 74
  from system failure 13
  leaked space 76
  missing backup 79
  missing transaction log 79
  normal 52
  replacing a write server 57, 78

server 2, 51
special modes 73
system 51, 52
transactions in 52
versioning in 52
repair
allocation 69
database 2, 51
indexes 81
tables 81
repair utility report 66
request logging level 47
resetclocks
sp_iqcheckdb option 57

S

sample database xi
section 508
compliance xi
server
CPU usage 9
deadlock 11
out of space 9
problems with shutdown 11
recovery 2, 51
startup failure 74
startup on Windows 6
stops processing 9
transaction log 4, 5
unique name 5
unique port number 6
unresponsive 9, 10
shutdown
troubleshooting 11
sp_iqcheckdb
allocation mode 56
allocation verification and repair 69
analyzing allocation problems 69
analyzing index problems 63
check mode 55
checking allocation 56
checking indexes and allocation 55
database verification 52
DBCC functions 53
DBCC_LOG_PROGRESS option 58
dumpdups option 57
dumpleaks option 57
dumpunallocs option 57
index verification and repair 63
internal index checking 55
interpreting output 70
output 58
output messages 83
performance 54
repair mode 58
repairing allocation 69, 71
repairing indexes 58, 63, 66
resetclocks option 57
resource issues 61
sample output 58
syntax 53
time to run 54
verify mode 55
sp_iqstatus
sample output 38
use in troubleshooting 38
sp_iqtransaction
determining blocking writers 29
stack trace
generating for threads 10
location 13
standards
section 508 compliance xi
standards and compatibility
section 508 compliance xi
start_asiq
command will not run 7
parameters 7
troubleshooting 7
startup
allocation error 74
checkpoint error 74
resolving a failure 74
troubleshooting hints 4
Sybase Central
display problems 25, 32
Sybase IQ 12.5 servers 18
system failure
recovering from 13
system unresponsive 9

T

tables
- blocked access 29
- corrupt 81
- managing blocked access 30
- unrepairable errors 80

TCP/IP
- testing 35
- troubleshooting 34

Technical Support
- checklist 50
- MySybase 49
- online help 49
- reporting problems to 44

Telnet
- TCP/IP testing 35

thrashing
- HASH_THRASHING_PERCENT option 27

threads
- generating a stack trace 10
- not enough 24

trace
- generating for threads 10

transaction log
- recovering without 79

transactions
- in recovery 52

troubleshooting 1
- common problems 36
- database connection 13
- DBISQL 18
- DBISQLC 18
- multiplex 17
- processing issues 26
- protocols 32
- resource issues 19
- server operation 3
- Sybase Central 32
- wiring problems 36

V

versioning
- in recovery 52

W

wiring
- troubleshooting 36

write server
- replacing 57, 78

write server mismatch 17