SYBASE®

Utility Guide

Sybase® IQ

12.6

DOCUMENT ID: DC00168-01-1260-01

LAST REVISED: December 2004

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

Contents

About This Book

Sybase® IQ is a high-performance decision support server designed specifically for data warehouses and data marts. This book, *Sybase IQ Utility Guide*, provides reference material for the utility programs used with Sybase IQ. Utility programs are commands that you invoke directly from the operating system. Other manuals provide more context on how to carry out particular tasks. This reference manual is the place to look for information such as available syntax, parameters, and options.

Audience

This manual is a reference for all users of Sybase IQ.

How to use this book

This book provides comprehensive descriptions of the Sybase IQ utility programs. However, it does not describe when and why you may want to use each utility. This manual is designed to be used as a reference together with the other books in the Sybase IQ documentation set.

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

Table 1: Guide to using this book

To learn how to...	Refer to...
Stop and start the database server	Chapter 1, “Running the Database Server”
Use Interactive SQL (dbisql)	Chapter 2, “Using Interactive SQL (dbisql)”
Run the database administration utilities	Chapter 3, “Database Administration Utilities”
Run the SQL preprocessor (sqlpp)	Chapter 4, “The SQL Preprocessor”

Note 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, with the Sybase Central™ database management tool.

- *New Features in Sybase IQ 12.6*

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 manage the IQ Store.

- *Sybase IQ Troubleshooting and Error Messages Guide*

  Read to solve problems, perform system recovery and database repair, and understand both IQ error messages which are referenced by SQLCODE, SQLSTATE, Sybase error code, and message text, and SQL preprocessor errors and warnings.

- *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.

- *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.

Note 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 Error Messages*

  This book lists all Adaptive Server Anywhere error messages with diagnostic information.

- *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 9.0.1 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/.

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. Select Products from the navigation bar on the left.
3. Select a product name from the product list and click Go.
4. Select the Certification Report filter, specify a time frame, and click Go.
5. Click a Certification Report title to display the report.

❖ **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, which many of the examples in the IQ documentation use.

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 12.6 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.

Note 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

Running the Database Server

About this chapter

This chapter describes the utility programs used to start and stop the Sybase IQ database server and the available startup options. There are other ways to start and stop the database server in addition to these utility programs. For a complete discussion of methods to start and stop the database server, and when each method is appropriate to use, see Chapter 2, "Running Sybase IQ" in the *Sybase IQ System Administration Guide*.

Starting the database server

The database startup utility start_asiq starts a Sybase IQ network database server.

The UNIX versions of Sybase IQ provide the script start_asiq and the Windows version of IQ provides start_asiq.exe. These scripts verify that your environment is set correctly and start the server with all required switches preset to recommended defaults (along with any switches you add). start_asiq also includes some parameters and calculates others. For more information about switches specific to your operating system, see the *Sybase IQ Installation and Configuration Guide*.

Note The Start Database Server utility in Sybase Central provides an easy graphical interface for starting servers and is the recommended method for starting IQ multiplex servers. For details, see Chapter 2, "Running Sybase IQ" in the *Sybase IQ System Administration Guide*.

Syntax

start_asiq -n *server-name*

[*server-options*] [*database-file* [*database-options*]]

Parameters

The following tables list the available switches for the start_asiq *server-options* and *database-options* parameters.

You can list the start_asiq options using the command

```
start_asiq -?
```

Server switches

Table 1-1: start_asiq server-options

Option	Description
@*filename*	Read in options from configuration file
@envvar	Read in options from environment variable
-c *cache-size*	Set initial Catalog Store cache size.
-ca 0	Disable dynamic Catalog Store cache sizing
-ch *size*	Set Catalog Store cache size upper limit
-cl *size*	Set the cache size lower limit
-ct { + | - }	Enable/disable character set translation (enabled by default)
-cw	Enable use of Address Windowing Extensions on Windows 2000, Windows XP, and Windows Server 2003 for setting the size of the database server cache.
-ec *encryption-options*	Enable packet encryption [network server].
-ek *key-spec*	Starts an encrypted database, when key value is provided as an argument.
-ep	(Windows) Displays a dialog box that prompts you for an encryption key to start an encrypted database. Provides extra security by never allowing the encryption key to be seen in clear text.
-ga	Automatically unload database after last connection closed
-gb *level*	Set database process priority class to *level* [Windows]
-gc *num*	Set checkpoint timeout period to *num* minutes
-gd *level*	Set the permission required to start and stop the database
-ge *size*	Sets the stack size for threads that run external functions [Windows]
-gk *level*	Set the permission required to stop the server
-gl *level*	Set the permission required to load data
-gm *level*	Limit the number of connections to the server that can be active at one time. If this number is greater than the number that is allowed under licensing constraints, this option has no effect. The value should approximate the number of users expected to connect to the server. The default is 10 connections.
-gn *integer*	Set the number of execution threads that will be used for the Catalog Store and connectivity while running with multiple users.
-gp *size*	Set maximum page size to *size* bytes
-gr *num*	Set maximum recovery time to *num* minutes
-gu *level*	Utility commands permission level: utility_db, all, none, dba

Option	Description
-h or -?	Display usage information; show a list of all options
-iqgovern *num*	Specify the number of concurrent queries
-iqlocalreplay { all | none | [grants | procedures | views] ...}	Specify what gets recreated during TLV log replay from the local store after IQ server synchronization. The default is -iqlocalreplay all.
-iqmc *size*	Specify the main cache size in MB. (Overrides default.)
-iqmt *num*	Specify the number of threads that IQ can use on a multi-threaded system. The default is `60*numCPU+2*num_conn+1`. The minimum value is `2*num_conn+1`.
-iqmpx_ov 1	*For use starting multiplex databases only.* Starts the server with override to acknowledge that the write server is starting (1) on a different host, (2) with a different server name, or (3) using a different path to its catalog (.db) file. Do not start two write servers against the same database.
-iqmpx_sn 1	*For use starting multiplex databases only.* Starts the write server in single-node operating mode, for certain types of recovery.
-iqnumbercpus	Override the number of physical CPUs with the number available to IQ
-iqpartition	Specify number of partitions in main and temp buffer caches
-iqtc *size*	Specify temporary cache size in MB. (Overrides default.)
-iqtss *size*	Specify the thread stack size in KB.
-iqwmem *size*	Specify the size in MB of a special memory pool that cannot be paged for HP and Sun UNIX platforms
-m	Truncate transaction log after checkpoint
-n *name*	Use *name* as the name of the database server **Note** *There are two -n options.*The -n option is positional. If this option appears after a database file name, it has a different meaning. See"Database options" on page 4.
-o *filename*	Output server messages to the specified file
-os	Specify maximum size of file for server messages
-p *packet-size*	Set maximum network packet size
-qi	Control whether database server tray icon and window appear [Windows]
-qp	Do not display messages about performance in the database server window

Option	Description
-qs	Suppress startup error dialogs [Windows]
-qw	Do not display database server screen
-s	Set the syslog facility ID (none, user, daemon, local0,..., local7) [UNIX]
-sb {0 | 1}	Specify how the server reacts to broadcasts on TCP/IP
-ti *min*	Client idle time before shutdown:—default 4400 minutes
-tl *sec*	Default liveness timeout for clients in seconds—default is 120 seconds
-tq *time*	Set quitting time
-ud	Run as a daemon [UNIX] (not recommended; not used with start_asiq)
-ut *min*	Touch temporary files every *min* minutes [UNIX]
-v or -v2	Display database server version
-x *list*	Comma separated list of communication links to try
-xs	Specify server side web services communications protocols.
-z	Provide diagnostic information on communication links
-zl	Capture most recently-prepared SQL statement for each connection to a database on the server
-zo	Specify file for logging server requests
-zr *level*	Enable server request-level logging
-zs *integer* **|** *integerG* **|** *integerK* **|** *integerM* **}**	Specify maximum size of file for server request logging

Database options

Table 1-2: start_asiq database-options

Option	Description
-m	Truncate transaction log after checkpoint
-n *name*	Name the database **Note** *There are two -n options.*The -n option is positional. If this option appears after a database file name, it is a database option. Otherwise, it is a server option.

Multiplex options

Sybase recommends that you start and stop multiplex databases using the Sybase Central interface. If you cannot run Sybase Central, or if you need to start multiplex databases from a script, use the startup commands described in this section instead.

Startup or shutdown operations require multiple commands:

- A start_asiq command to start the appropriate server(s).
- One or more dbremote command(s) to control communication between the write and query servers. dbremote is always run on the write server host.
- A dbstop or stop_asiq command to stop the server(s).

You can specify the following multiplex options on the start_asiq command line for managing multiplex databases.

Note If used improperly, these options can cause problems. Make sure you read the following descriptions carefully before using the multiplex startup options.

Table 1-3: start_asiq multiplex options

Switch	Description
-iqmpx_ov 1	*For use starting multiplex databases only.* Starts the server with override to acknowledge that the write server is starting (1) on a different host, (2) with a different server name, or (3) using a different path to its catalog (.db) file. Do not start two write servers against the same database.
-iqmpx_sn 1	*For use starting multiplex databases only.* Starts the write server in single-node operating mode, for certain types of recovery. Query server(s) cannot run while the write server is in single-node mode.

For help on dbremote options, type:

```
dbremote -h
```

Table 1-4 lists the dbremote options that are recommended when running IQ multiplex servers.

Table 1-4: Recommended dbremote options for multiplex servers

Switch	Description
-k	Closes window when finished
-o *output_filename*	Specifies an output file
-q	Runs minimized
-v	Specifies verbose output

Sybase provides shell scripts in the write server database directory to start dbremote for each server in the multiplex. There is currently no service mechanism to start these processes automatically; this is up to the user. For example, on a Windows system, a DBA may add an entry in the Windows registry so that these processes start as soon as the DBA logs into the system. Note that dbremote is only required for data replication and servers run without it are still viable. Old data versions will not be recovered at the write server when dbremote is running.

Multiplex startup example

The following example starts the multiplex server *Server02*. If it was configured to be a query server, it will come up as a query server automatically.

```
start_asiq @params.cfg -n Server02
-x "tcpip(port=1234)" mpxdb.db
```

Then, on the write server's host:

```
dbremote -c
"uid=DBA;pwd=SQL;eng=Server02;links=tcpip{host=ciaran;
port=1234};dbf=mpxdb"
-v -o outfile.out
```

For more examples of startup and dbremote commands, generate the administrative scripts as described in *Sybase IQ System Administration Guide*. Use a text editor to view the commands in the scripts.

Always run dbremote on the write server's host, whether you are starting a query server or the write server. You can customize the administrative scripts to run dbremote. For more information, see "Using administrative shell scripts" in Chapter 5, "Working with Database Objects" of the *Sybase IQ System Administration Guide*.

On Windows systems, a SQL Remote screen will appear. You can minimize this screen.

Recovery options

The start_asiq *recovery-options* are a subset of the database-options that are used only for database recovery operations.

Table 1-5: start_asiq recovery-options

Description	Option
-iqdroplks *dbname*	Allow the sp_iqcheckdb stored procedure to recover leaked storage within the specified database.
-iqfrec *dbname*	Mark the specified database as in use and restore database to its last known consistent state.

Usage

-n server-name Specifies the name of the database server.

database-file Specifies the database filename. If *database-file* is specified without a file extension, Sybase IQ looks for *database-file* with extension .db.

If you use a relative path, the path is read relative to the current working directory of the server. You can supply a full path. On Windows you can supply a path that conforms to the Universal Naming Convention (UNC) format:

```
\\server\volume\path\file.ext
```

Warning! The database file must be on the same machine as the database server. Managing a database file that is located on a network drive can lead to file corruption.

Server command-line options

@filename Reads in command-line options from the supplied file.

The file may contain line breaks, and may contain any set of command line options. For example, the following Windows command file holds a set of command line options for a server named myserver that allows 15 connections, sets the maximum Catalog page size to 4 MB, and loads the sample database:

```
-gm 15
-gp 4096
-n myserver
c:\sybase\ASIQ-12_5\demo\asiqdemo.db
```

If this configuration file is saved as *c:\config.txt*, the file can be used in an command line as follows:

```
start_asiq @c:\config.txt
```

@environment-variable Reads in command-line options from the supplied environment variable. The environment variable may contain any set of command line options. For example, the first of the following pair of Windows statements sets an environment variable holding a set of command line options and loads the sample database. The second statement starts the database server:

```
set envvar= “-gp 4096 -gm 15”
```

```
c:\sybase\ASIQ-12_5\demo\asiqdemo.db start_asiq
@envvar
```

Note If you have both a file and an environment variable with the value of your @ command-line option, the result is unpredictable. Use only one of these methods to set a given @ command line option.

-c *cache-size* Sets initial memory reserved for caching Catalog Store pages and other server information. The database server uses extra memory for caching database pages if the memory is set aside in the cache. Any cache size less than 10000 is assumed to be KB (1K = 1024 bytes). Any cache size 10000 or greater is assumed to be in bytes. The cache size may also be specified as `nK` or `nM` or `nP` (1M = 1024 KB).

The default value of `-c` in the default.cfg file and start_asiq is 32MB (`-c 32M`) for Windows platforms, and 48MB (`-c 48M`) for UNIX platforms. For IQ databases, Sybase recommends that you use this default or set `-c` to a higher value.

The unit `P` is a percentage of the physical system memory. You can use % as an alternative to P, but as most non-UNIX operating systems use % as an environment variable escape character, you must escape the % character. For example, to use 20 percent of the physical system memory, you would specify:

```
start_asiq -c 20%% ...
```

On UNIX operating systems, if the cache size specified with `-c` is greater than the amount of available memory, the database server uses a maximum cache size that is calculated as follows:

If no `-c` option is provided (either on the command line or using the start_asiq default), the database server computes the initial Catalog Store cache allocation as follows:

1 The database server uses 32MB as the minimum default cache size.

2 The database server computes a runtime-specific minimum default cache size, which is the lesser of the following items:

- 25% of the physical memory of the machine
- The sum of the sizes of the main database files specified on the command line. Additional dbspaces apart from the main database files are not included in the calculation. If no files are specified, this value is zero.

3 The database server allocates the greater of the two values computed.

Do not use -c in the same configuration file or command line with -ch or -cl. For related information, see the -ch cache-size option.

See also the -ca 0 option.

-ca 0 Enforces a static Catalog cache size. The zero argument is required.

Ordinarily, the database server automatically takes additional cache as needed. You can disable automatic cache increase due to high server load by using -ca 0 on the command line. The cache size still increases if the database server would otherwise run into the error

```
Fatal Error: dynamic memory exhausted
```

or if the Java VM requires memory that would lead to a fatal error.

-ch *cache-size* Limits the Catalog Store cache that the database server can take during automatic cache growth. By default the upper limit is approximately the lower of 256 MB and 90% of the physical memory of the machine.

You specify the cache-size using the K, M, and P characters as in the -c option. For the meaning and usage of the cache size argument and the K, M, and P characters, see -c cache-size.

In some cases the standard Catalog cache size may be too small, for example, to accommodate certain queries that need a lot of parsing. In these cases, you may find it helpful to set -cl and -ch. For example, on 32-bit platforms, try these settings

```
-cl 128M
-ch 256M
```

Warning! To control Catalog Store cache size explicitly, you must do *either* of the following, but not both, in your configuration file (*.cfg*) or on the UNIX command line for server startup:

- Set the -c parameter
- Set specific upper and lower limits for the Catalog Store cache size using the -ch and -cl parameters

Specifying different combinations of the parameters above can produce unexpected results.

-cl *cache-size* Sets a lower limit to the Catalog Store cache during automatic cache growth. By default the lower limit is the initial cache size.

The amount of cache memory available to the Catalog Store can affect performance, especially when many users are connected. You specify the cache-size using the K, M, and P characters as in the -c option. For the meaning and usage of the cache size argument and the K, M, and P characters, see -c cache-size.

-ct { **+** | **-** } By default, character set translation is turned on. Character set translation converts strings between character sets that represent the same characters, but at different values. This is useful when the client machine and the database use different character sets. To disable character set translation, specify -ct-. To enable character set translation, specify -ct+.

Prior to version 12.6, the + or - values were not accepted: specifying the -ct option enabled character set translation.

For more information, see Chapter 11, "International Languages and Character Sets" of the *Sybase IQ System Administration Guide*.

-cw Enables use of Address Windowing Extensions (AWE) on Windows 2000, Windows XP, and Windows Server 2003 for setting the size of the Catalog Store cache.

Because Windows 2000, Windows XP, and Windows Server 2003 support Address Windowing Extensions, you can use the -cw option to take advantage of large cache sizes based on the maximum amount of physical memory in the system. Remember, though, that the size of the Catalog Store cache has much less impact on performance for IQ databases than the IQ main and temporary buffer caches.

Operating system	Maximum non-AWE cache size	Maximum amount of physical memory supported by Windows
Windows 2000 Professional	1.8 Gb	4 Gb
Windows 2000 Server	1.8 Gb*	4 Gb
Windows 2000 Advanced Server	2.7 Gb*	8 Gb
Windows 2000 Datacenter Server	2.7 Gb*	64 Gb
Windows XP Home Edition	1.8 Gb	2 Gb
Windows XP Professional	1.8 Gb	4 Gb
Windows Server 2003, Web Edition	1.8 Gb	2 Gb

Operating system	Maximum non-AWE cache size	Maximum amount of physical memory supported by Windows
Windows Server 2003, Standard Edition	1.8 Gb	4 Gb
Windows Server 2003, Enterprise Edition	2.7 Gb*	32 Gb
Windows Server 2003, Datacenter Edition	2.7 Gb*	64 Gb

*You must boot the operating system using the `/3GB` option to use a cache of this size.

When using an AWE cache, almost all of the available physical memory in the system can be allocated for the cache.

If you can set a Catalog Store cache of the desired size using a non-AWE cache, this is recommended because AWE caches allocate memory that can only be used for the Catalog Store. This means that while the database server is running, the operating system and the IQ Store caches cannot use the memory allocated for the Catalog Store cache.

AWE caches do not support dynamic cache sizing. Therefore, if an AWE cache is used and you specify the -ch or -cl options to set the upper and lower cache size, they are ignored.

To start a database server with an AWE cache, you must do the following:

- Have at least 130 Mb of memory available on your system.
- If your system has between 2 Gb and 16 Gb of memory, add the `/3GB` option to the Windows boot line in the "[operating systems]" section of the boot.ini file.

  If your system has more than 16 Gb of memory, do not add the `/3GB` option to the Windows boot line in the "[operating systems]" section of the boot.ini file because Windows will not be able to address memory beyond 16 Gb.
- If your system has more than 4 Gb of memory, add the `/PAE` option to the Windows boot line in the "[operating systems]" section of the boot.ini file.
- Grant the "Lock pages in memory" privilege to the user ID under which the server is run. The following steps explain how to do this on Windows 2000.

  a Log on to Windows as Administrator.

b From the Start menu, choose Settings > Control Panel.

c Open the Administrative Tools folder.

d Double-click Local Security Policy.

e Open Local Policies in the left pane.

f Double-click User Rights Assignment in the left pane.

g Double-click the Lock Pages In Memory policy in the right pane.

The Local Security Policy Setting dialog appears.

h In the Local Security Policy Setting dialog, click Add.

The Select Users or Groups dialog appears.

i Select the user ID from the list and click Add.

j In the Local Security Policy Setting dialog, click OK.

k Restart the computer for the setting to take effect.

If you specify the -cw option and the -c option on the command line, the database server attempts the initial cache allocation as follows:

1 The AWE cache is no larger than the cache size specified by the -c option. If the value specified by the -c option is less than 2 Mb, AWE is not used.

2 The AWE cache is no larger than all available physical memory less 128 Mb.

3 The AWE cache is no smaller than 2 Mb. If this minimum amount of physical memory is not available, an AWE cache is not used.

When you specify the -cw option and do not specify the -c option, the database server attempts the initial cache allocation as follows:

1 The AWE cache uses 100% of all available memory except for 128 Mb that is left free for the operating system.

2 The AWE cache is no larger than the sum of the sizes of the main database files specified on the command line. Additional dbspaces apart from the main database files are not included in the calculation. If no files are specified, this value is zero.

3 The AWE cache is no smaller than 2 Mb. If this minimum amount of physical memory is not available, an AWE cache is not used.

When the server uses an AWE cache, the Catalog cache page size is at least 4 KB and dynamic cache sizing is disabled.

For more information about dynamic cache sizing, see the -ch and -cl server options.

–ec Encrypts all native Sybase IQ packets (DBLib, ODBC, and OLE DB) transmitted to and from all clients. TDS packets are not encrypted.

{ **dbsrv9** | **dbeng9** } **-ec** *encryption-options* …

encryption-options:

{ **NONE** | **SIMPLE** | **ECC_TLS (CERTIFICATE=***filename*;
CERTIFICATE_PASSWORD=*password* **)**
| **RSA_TLS (CERTIFICATE=***filename*;
CERTIFICATE_PASSWORD=*password* **)**
| **ALL** } , …

The -ec option instructs the database server to accept *only* connections from ODBC, OLE DB, or embedded SQL interfaces that are encrypted using one of the specified types. Connections over the TDS protocol, which include Java applications using jConnect, are always accepted, regardless of encryption.

By default, communication packets are not encrypted, which poses a potential security risk. If you are concerned about the security of network packets, use the -ec option. Encryption affects performance only marginally. The -ec option controls the server's encryption settings and requires one or more of the following parameters in a comma-separated list:

- none — Accepts only connections that are not encrypted.
- simple — Accepts connections that are encrypted with simple encryption. This type of encryption is supported on all platforms. Simple encryption is not as strong as Certicom encryption.
- ECC_TLS — Currently not supported by Sybase IQ.
- RSA_TLS — Accepts connections that are encrypted using RSA-based encryption technology. To use this type of encryption, both the server and the client must be operating on Solaris, Linux, AIX, or any supported Windows platform, and the connection must be over the TCP/IP port. This parameter accepts the following arguments:
  - certificate — the file name of the certificate. The default value is rsaserver.crt.
  - certificate_password— the password for the certificate named above. The password for rsaserver.crt is test.

- all — Accepts connections that are not encrypted (none), or encrypted with simple encryption (simple). This is the default.

Warning! The sample certificate should be used for testing purposes only. The sample certificate provides no security in deployed situations because it and the corresponding password are widely distributed with Sybase software. To protect your system, you must create your own certificate.

The dbtls9.dll and dbrsa9.dll files contain the Certicom code used for encryption and decryption. When you connect to the server, if the appropriate file cannot be found, or if an error occurs, a message appears on the server console in debug mode. The server does not start if the types of encryption specified cannot be initiated.

The client's and the server's encryption settings must match or the connection will fail. The server automatically encrypts client transmissions that request encryption if -ec none is not specified.

The following sample startup lines show how you start the server with each of the three types of encryption:

```
start_asiq -ec simple,certicom(certificate=sample.crt;
certificate_password=tJ1#m6+W) -x tcpip asiqdemo.db

start_asiq -ec ecc_tls(certificate=sample.crt;
certificate_password=tJ1#m6+W) -x tcpip asiqdemo.db

start_asiq -ec rsa_tls(certificate=rsaserver.crt;
certificate_password=test) -x tcpip asiqdemo.db
```

For more information, see "Encryption connection parameter [ENC]" in *Sybase IQ System Administration Guide*.

-ek Provided after the filename of a strongly encrypted database. Requires the key value as an argument to start an encrypted database. The key value is a string, including mixed cases, numbers, letters, and special characters. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way. For a strongly encrypted database, you must specify either -ek or -ep, but not both. If you do not specify a key for a strongly encrypted database, the command will fail.

-ep (Windows) Displays a dialog box that prompts user to enter encryption key for running a strongly encrypted database. This option provides an extra measure of security by never allowing the encryption key to be seen in clear text. For a strongly encrypted database, you must specify either -ep or -ek, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

The engine must either not be a Windows service, or be a Windows service with the interact with desktop option turned on.

When used with supported tools, this option always prompts the user for the encryption key, even if a key is not necessary. If you know that a key is not necessary, you can click Cancel to continue when the dialog box prompt appears.

-ga Specifying this switch on the network server causes each database to be unloaded after the last connection to the database is dropped. The database server itself does not shut down.

-gb *level* (Windows) Sets the database process priority class to *level*. Level must be one of idle, normal (the default), high, or maximum. The value idle is provided for completeness, and maximum may interfere with the running of your computer. The commonly used settings are normal and high.

-gc *num* Sets the maximum number of minutes the database server runs without doing a checkpoint on each database. The default value is 20 minutes.

-gd *level* Sets the permission level required to start a database on the server, or to stop a database:

- **DBA** Only users with DBA authority can start a database or stop a database.
- **ALL** All users can start a database or stop a database (default in start_asiq and default.cfg). Use this setting so that the DBA does not need to issue START DATABASE commands. (Note that users still need permission to access a database once they have started the database.)
- **NONE** Starting a database from Interactive SQL is not allowed. (Running stop_asiq on UNIX or Shutdown on Windows still allows you to stop the server and database.)

Both uppercase and lowercase syntax are acceptable.

-ge *size* (Windows) Sets the stack size for threads running external functions, in bytes. The default is 16384 (16KB).

-gk *level* Sets the permission level required to stop the database server:

- **DBA** Only users with DBA authority can use dbstop to stop the server (the default).
- **ALL** All users can use dbstop to stop the server.
- **NONE** The server cannot be stopped using dbstop.

Both uppercase and lowercase syntax are acceptable.

-gl *level* Sets the permission required to load data using LOAD TABLE.

The LOAD TABLE statement reads files from the database server machine.

To control access to the file system using these statements, the -gl command-line switch allows you to control the level of database permission that is required to use these statements.

The allowed values are as follows:

- **DBA** Only users with DBA authority can load data.
- **ALL** All users can load data.
- **NONE** Data cannot be loaded.

Both uppercase and lowercase syntax are acceptable.

The default settings are all for servers started with start_asiq and dba for other servers. Sybase recommends that you use the setting all on all systems for consistency with earlier versions. The all setting is used in the asiqdemo.cfg and default.cfg configuration files. For more about these configuration files, see *Table 2-1: Configuration files* in *Sybase IQ System Administration Guide*.

-gm *num* Limits the number of connections to the server that can be active at one time. If this number is greater than the number that is allowed under licensing and memory constraints, this switch has no effect. The default varies by machine capacity, but 15 is recommended. The value should approximate the number of users expected to connect to the server.

The database server allows one extra DBA connection above the connection limit to allow a DBA to connect to the server and drop other connections in an emergency.

-gn *num* Sets the number of execution threads that will be used for the Catalog Store and connectivity while running with multiple users. This parameter applies to all operating systems and servers.

Each connection uses a thread for each request, and when the request is completed the thread is returned to the pool for use by other connections. As no connection can have more than one request in progress at one time, no connection uses more than one thread at a time.

An exception to this rule is if a Java application uses threads. Each thread in the Java application is a database server execution thread.

On Windows you need to specify this parameter in the start_asiq command. To calculate its value use the following formula:

```
gn_value = gm_value + 5
```

Specify a minimum of 25. The total number of threads (-iqmt plus -gn) must not exceed a platform-specific maximum; see "-iqmt num" on page 18 for details.

The start_asiq utility sets this parameter. See the *Sybase IQ Installation and Configuration Guide* for your platform for more information.

-gp *size* Sets the maximum page size allowed, in bytes, for the Catalog Store. The size specified must be 1024, 2048, 4096, 8192, 16384, or 32768. The minimum page size on all UNIX platforms is 2048 bytes. *Sybase highly recommends that you set the -gp switch to 4096.*

-gr *num* Sets the maximum length of time, in minutes, that the database server will take to recover from system failure.

For more information, see "RECOVERY_TIME option" in Chapter 2, "Database Options" in the *Sybase IQ Reference Manual*.

-gu *level* Sets permission levels for commands such as CREATE DATABASE and DROP DATABASE. The level can be set to one of following: utility_db, all, none, dba.

The utility_db level restricts the use of these commands to only those users who can connect to the utility database. The all, none, and dba levels respectively permit all users, no users, or connected users with dba authority to execute utility commands.

To connect to the utility database, you must specify utility_db as the database name (that is, there is no database file) and the utility database password (stored in the file util_db.ini) must be known.

-h Displays usage information. Sybase IQ also displays usage information if you specify any non-valid switch, such as -help.

-iqgovern *num* Sets the number of concurrent queries allowed by the server. The number of concurrent queries is not the same as the number of connections. This switch can help Sybase IQ optimize paging of buffer data out to disk and avoid overcommitting memory. The default value of this switch is equal to 2 times the number of CPUs on your machine plus 10. You may find that another value, such as 2 times the number of CPUS plus 4, provides better throughput, especially when large numbers of users are connected.

-iqlocalreplay { all | none | [grants | procedures | views] ... } Specifies what is reconstituted on the query server during TLV log replay from a Local IQ Store after query server synchronization. IQ tables and user-defined data types must always be enabled for replay.

The allowed values are as follows:

- **all** The log replays all DDL commands.
- **none** The log replays no DDL commands.
- **grants** The log replays GRANT and REVOKE commands, reflecting changes in user names, groups, and permissions.
- **procedures** The log replays actions that affect stored procedures and functions, including user messages.
- **views** The log replays actions that affect views.

For example,

```
-iqlocalreplay "all"
```

This example replays all DDL commands from the local TLV log if the server was just synchronized.

-iqmc *size* Specifies main IQ Store cache size in MB. Overrides default of 16MB, or value, if any, set by MAIN_CACHE_MEMORY_MB database switch. Applies to all databases started from the time the IQ server is started until the IQ server is shut down. In other words, if you start one database at server startup and another later, you need 2 * -iqmc available for the main cache. In general, *Sybase does not recommend running multiple databases with a Sybase IQ server.*

-iqmpx_ov 1 *For use starting multiplex databases only.* Starts the server with override to acknowledge that the write server is starting (1) on a different host, (2) with a different server name, or (3) using a different path to its catalog (.db) file. Do not start two write servers against the same database. For details about using this parameter, see "Multiplex options" on page 4

-iqmpx_sn 1 *Use only with a multiplex database.* Starts the write server in single-node operating mode for certain types of recovery. Do not use to start query servers.

-iqmt *num* Specifies the number of Sybase IQ threads to create. The default is `60*numCPU+2*num_conn+1`. The minimum value is `2*num_conn+1`. The total number of threads (-iqmt plus -gn) must not exceed 4096 on 64-bit platforms, 1000 on AIX 32-bit servers, or 2048 on all other 32-bit platforms. The default -iqtss setting should be adequate to support these maximum numbers of threads.

-iqnumbercpus *num* Specifies the number of CPUs available to IQ, overriding the physical number of CPUs for resource planning purposes. The value of the parameter defaults to the total number of CPUs, but the range of available values is 1 through 128.

Sybase recommends using -iqnumbercpus only in the following situations:

- On machines with Intel® CPUs and hyperthreading enabled, setting -iqnumbercpus to the number of CPUs available
- On machines where an operating system utility has been used to restrict Sybase IQ to a subset of the CPUs within the machine

Setting -iqnumbercpus higher than the number of available CPUs may affect performance.

-iqpartition Specifies the number of partitions in the IQ main and temp buffer caches. Must be a power of 2. Allowed values are: 0 (default), 1, 2, 4, 8, 16, 32, 64. By default, IQ computes the number of partitions automatically as *number_of_cpus/8*, rounded to the nearest power of 2, up to a maximum of 64. You may be able to improve performance by adjusting the number of cache partitions. The -iqpartition switch sets this value for an IQ server, and overrides the value set by the Cache_Partitions database option. To set the number of partitions for a particular database, use the Cache_Partitions database option. See "CACHE_PARTITIONS option" in Chapter 2, "Database Options" of the *Sybase IQ Reference Manual*, and "Managing lock contention" in Chapter 10, "Transactions and Versioning" of the *Sybase IQ System Administration Guide* for more information on whether you should set one of these options, and factors that affect them.

-iqtc *size* Specifies IQ Temporary Store cache size in MB. Overrides default of 8MB, or value, if any, set by TEMP_CACHE_MEMORY_MB database option. Applies to all databases started from the time the IQ server is started until the IQ server is shut down. In other words, if you start one database at server startup and another later, you need 2 * -iqtc available for the temp cache. In general, *Sybase does not recommend running multiple databases with a Sybase IQ server.*

-iqtss *size* Sets the stack size of the internal execution threads in the server. The default value is 200 KB for 32-bit platforms and 300 for 64-bit platforms, which is the recommended value. However, some very complex queries may return an error indicating that the depth of the stack exceeded this limit, so the DBA may need increase this value. On 32-bit platforms, the best setting depends on the available memory, number of CPUs, number of connections and type of queries run. On 64-bit platforms, the value 1000 is reasonable because the memory of 64-bit systems is not constrained as it is in 32-bit systems. Do not choose a value less than the default or you may find that some queries cannot run.

-iqwmem *size* Creates a pool of "wired" memory on HP and Sun UNIX systems. This memory is locked down so it cannot be paged by the operating system. You specify the size in MB of memory. Use this switch *only if you have enough memory to dedicate for this purpose*. Otherwise, you can cause serious performance degradation.

-m Deletes the transaction log when a checkpoint is done, either at shutdown or as a result of a checkpoint scheduled by the server. This provides a way to automatically limit the growth of the transaction log. Checkpoint frequency is still controlled by the CHECKPOINT_TIME and RECOVERY_TIME options (which also can be set in the command line).

The -m server switch is useful where high volume transactions requiring fast response times are being processed, and the contents of the transaction log are not being relied upon for recovery or replication.

Warning! When you select the -m server switch, there is no protection against media failure on the device that contains the database files.

To avoid database file fragmentation, when you use this switch place the transaction log on a separate device or partition from the database itself.

If you start the server with the -m switch, you cannot create a database.

Note Do not use the -m switch with databases that are being replicated, as replication inherently relies on transaction log information. For this reason, *never* use the -m switch on a multiplex database.

For information on truncating the transaction log file using the-m switch, see "The transaction log file" in Chapter 4, "Managing System Resources" of the *Sybase IQ Performance and Tuning Guide*.

-n *name* Sets the name of the database server. By default, the database server receives the name of the database file with the path and extension removed. For example, if the server is started on the file *c:\sybase\ASIQ-12_6\demo\asiqdemo.db* and no -n switch is specified, then the name of the server is asiqdemo. You should, however, change the default name.

Each server name must be unique across the local area network (domain). This prevents you from unintentionally connecting to the wrong server. The host name and port number combination does not uniquely identify the server.

The server name is interpreted according to the character set of the machine, as no database collation exists at startup time. The server name must be a valid identifier. Long server names are truncated to 40 characters. On NetBIOS, 16 characters is the maximum length. For multiplex server names, 30 characters is the maximum.

The server name specifies the name to be used on client application connection strings or profiles.

Note *There are two -n switches.*The -n switch is positional. If this switch appears after a database file name, it is a database option. Otherwise, it is a server option. For example, in the following start_asiq command line, the first -n indicates a server name and the second -n, which follows the database file name mydb.db, indicates a database name:

```
start_asiq -n svrname mydb.db -n dbname
```

The server name must be used on the connect statement to specify to which server you wish to connect. In all environments, there is always a default database server that will be used if no server name is specified provided at least one database server is running on the system. Always specify a server name to avoid using the default name. *Each server name must be unique across the local area network (domain).* This prevents you from unintentionally connecting to the wrong server. The host name and port number combination does not uniquely identify the server.

Multiple database servers with the same name are not allowed to run on TCP/IP anywhere on the network, even on separate ports.

-o *filename* Prints all server message window output to a file, in addition to displaying the output on the screen. You can use this option to specify a nondefault name for the server log file, and keep using the same file after restarting the server. Be careful that you do not fill this file.

Note If the -o file is located within a file system that fills up, then the IQ server will hang. Once this condition exists, the only way to bring down the server is to kill it. To prevent this problem from occurring, specify the -os switch.

-os *size* Limits the size of the log file used by the -o switch. The default is no limit. The units G, K, and M can be either lower case or upper case. If G, K, or M is not supplied, any integer less than 10 000 is assumed to be in kilobytes, and any integer 10 000 or greater is assumed to be in bytes.

-p *packet-size* Sets the maximum size of communication packets. The default is 1460 bytes. The minimum value is 300 bytes and the maximum value is 16000 bytes. To set this value for a connection, see "CommBufferSize connection parameter [CBSize]" in the *Sybase IQ System Administration Guide*.

-qi On Windows servers, controls whether database server tray icon and window appear. This option leaves no visual indication that the server is running, other than possible startup error dialogs. You can use either (or both) the -o or -oe logs to diagnose errors.

-qp Disables display of messages about performance in the database server window. Messages that are suppressed include the following:

- No unique index or primary key for table '*table_name*'
- Database file "mydatabase.db" consists of *nnn* fragments

Specifies no server screen display (no console for the server).

-qs On Windows servers, suppresses startup error dialogs. Startup errors include errors such as:

- Could not open/read database file: <*database file*>
- A database server with that name has already started

On Windows platforms, if the server is not being autostarted, these errors appear in a dialog and must be cleared before the server stops. These dialogs do not appear if the -qs option is used.

If there is an error loading the language DLL, no dialog appears if -qs was specified on the command line and not in *@environment-variable* or *@filename* syntaxes. This error is not logged to the -o or -oe logs, but rather to the Windows Application Event Log.

Usage errors are suppressed if -qs is on the command line, but not in *@filename* or *@environment-variable* expansion.

-qw Suppresses the database server window (Windows platforms) and displays messages on the console (non-Windows platforms).

-s For UNIX servers, sets the system user ID used in messages to the syslog facility. The default is user, which uses the user ID for the database server process. A value of none prevents any syslog messages from being logged.

-sb { 0 | 1 } Specifies how the server reacts to broadcasts on TCP/IP.

Using -sb 0 causes the server not to start up any TCP/UDP broadcast listeners. In addition to forcing clients to use the DoBroadcast=NONE and HOST= options to connect to the server, this option causes the server to be unlisted when using dblocate.

Using -sb 1 causes the server to not respond to broadcasts from dblocate, while leaving connection logic unaffected. You can connect to the server by specifying LINKS=tcpip and ENG=<*name*>.

-ti *minutes* Disconnects connections that have not submitted a request for *minutes* minutes. The default is 4400 (72 hours), so that a user with a long query will not be logged off over a long weekend. A client machine in the middle of a database transaction holds locks until the transaction is ended or the connection is terminated. The -ti switch is provided to disconnect inactive connections, freeing their locks. The -ti switch does not disconnect clients that use the shared memory communications link. Setting the value to zero disables checking of inactive connections, so that no connections are disconnected.

You can set the connection timeout for individual connections using the IDLE connection parameter. For more information, see "Idle connection parameter [IDLE]" in *Sybase IQ System Administration Guide*.

-tl *seconds* Sets the liveness timeout period for the server. A liveness packet is sent periodically across a client/server TCP/IP communications protocol to confirm that a connection is intact. If the server runs for a liveness timeout period (default 2 minutes) without detecting a liveness packet, the communication is severed. The server drops any connections associated with that client. UNIX non-threaded clients and TDS connections do not do liveliness checking.

The -tl switch on the server sets the liveness timeout for all clients that do not specify a -tl switch.

When there are more than 200 connections, the server automatically calculates a higher LivenessTimeout value based on the stated LivenessTimeout value. This enables the server to handle a large number of connections more efficiently.

Clients send liveness packets at an interval of between one third and two thirds of the LivenessTimeout on each idle connection. Large numbers of liveness packets are not sent at the same time. If liveness packets take a long time to send (depending on the network, the machine's hardware, and the CPU and network load on the machine), it is possible that liveness packets will be sent after two thirds of the LivenessTimeout. A warning appears in the server console if the liveness sends take a long time. If this warning occurs, consider increasing the LivenessTimeout value.

-tq *time* Shuts down the server at a specified time. The format for the time is in HH:MM (24 hour clock), and can be preceded by an optional date. If a date is specified, the date and time must be enclosed in double quotes and be in the format “YYYY/MM/DD HH:MM”.

-ud For UNIX servers, causes the process to run as a daemon in the root directory. (Not recommended for IQ.)

-ut min For UNIX servers, causes the server to touch Catalog Store temporary files at intervals specified by *min*.

-v Displays the database server version in a message box, and then stops.

-x *list* Specifies server side network communications protocols.

The *list* is a comma-separated list of settings taken from the following list: TCPIP, or NamedPipes.

For example,

```
-x tcpip,ipx
```

allows only TCP/IP and IPX communications.

The default is to try all settings supported by database server on your operating system.

For some protocols, additional parameters may be provided, in the format

```
-x tcpip(PARM1=value1;PARM2=value2;...)
```

For UNIX, quotation marks are required if more than one parameter is supplied:

```
-x "tcpip(PARM1=value1;PARM2=value2;...)"
```

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

-xs Specifies server side web services communications protocols.

{ **dbeng9** | **dbsrv9** } **-xs** { **all** | **none** | *web-protocols* } …

web-protocols: { [**http** | **https**] *parmlist* },… *parmlist*: **(** *parm*=*value*;…**)**

Use the -xs option to specify which web protocols you want to use to listen for client connection broadcasts.

If you do not specify the -xs option, the server does not attempt to listen for web requests.

If you specify the -xs option with one or more protocols, the server attempts to listen for client requests using the specified protocol(s).

Regardless of which settings you choose for the -xs option, the server always listens for connection broadcasts using the shared memory protocol. You can specify any of the following:

- **ALL** Listen for connection attempts by the client using all communications protocols that are supported by the server on this platform, including shared memory.
- **HTTP** Listen for connection attempts by the client using the HTTP protocol. The default port on which to list is 80.
- **HTTPS** Listen for connection attempts by the client using the HTTPS protocol. The default port on which to list is 443.
- **NONE** Listen for connection attempts by the client using only the shared memory protocol. This is the default.

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

For UNIX, quotation marks are required if more than one parameter is supplied:

```
-xs "http(PARM1=value1;PARM2=value2;...)"
```

The following server command line allows only shared memory and TCP/IP communications:

```
start_asiq web.db -xs http(port=80)
```

For more information, see "CommLinks connection parameter [Links]" in *Sybase IQ System Administration Guide*.

-z Provides diagnostic information on communications links on startup. This should only be used when tracking problems.

-zl Enable capturing of the most recently-prepared SQL statement for each connection to a database on the server.

This feature can also be turned on using the remember_last_statement server setting. You can obtain the most recently-prepared SQL statement for a connection using the LastStatement property function. The sa_conn_activity stored procedure allows you to obtain the most recently-prepared SQL statement for all current connections to the database on the server.

For stored procedure calls, only the outermost procedure call appears, not the statements within the procedure.

-zo Redirects request-level logging information to a file separate from the regular log file. Request-level logging is turned on using the -zr switch. The -zo switch directs the output from this file to a separate file from that specified on a -o switch. This switch also prevents request-level logging from being displayed in the console.

-zr *level* Enables request-level logging of operations:

- ALL logs all SQL statements and other requests to the server.
- NONE turns off SQL statement logging. This is the default.
- SQL logs the following types of requests only:
  - CONTROL_START_DATABASE
  - CONTROL_STOP_ENGINE
  - CONTROL_STOP_DATABASE
  - STMT_PREPARE
  - STMT_EXECUTE
  - STMT_EXECUTE_IMM
  - STMT_EXECUTE_ANY_IMM
  - SQL_OPTION_SET
  - BACKUP
  - DELETE_FILE
  - COMMIT

- ROLLBACK
- PREPARE_TO_COMMIT
- CONNECT
- DISCONNECT
- BEGIN_TRANSACTION
- STMT_DROP
- CURSOR_OPEN
- CURSOR_EXPLAIN
- CURSOR_CLOSE
- CURSOR_RESUME
- Errors

This switch also prevents request-level logging from being displayed in the console.To correlate connection information in the -zr log file with that in the .iqmsg file, see "Correlating connection information" in Chapter 1, "Troubleshooting Hints" of the *Sybase IQ Troubleshooting and Error Messages Guide*.

See also "-zo" and "-zs { integer | integerG | integerK | integerM } …" command-line switches.

-zs { *integer* | *integerG* | *integerK* | *integerM* } … Limits the size of the request-level logging file. Request-level logging is turned on using the -zr switch, and redirected to a separate file using the -zo switch. You can limit the size of the file using the -zs switch.

The units G, K, and M can be either lower case or upper case. If G, K, or M is not supplied, any integer less than 10 000 is assumed to be in kilobytes, and any integer 10 000 or greater is assumed to be in bytes.

When the request log file reaches the size specified by either the -zs option or the sa_server_option system procedure, the file is renamed with the extension *.old* appended (replacing an existing file with the same name if one exists). The request-level log file is then restarted.

By default there is no limit. The value is in kilobytes.

The following example shows how the -zs option is used to control log file size. Suppose you start a database server with the following options on the command line:

```
-zr all -zs 10 -zo mydatabase.log
```

A new log file mydatabase.log is created. When this file reaches 10K in size, any existing mydatabase.old files are deleted, mydatabase.log is renamed to mydatabase.old, and a new mydatabase.log file is started. This process is repeated each time the mydatabase.log file reaches the specified size (in this case 10K).

Database options

-m Truncates (deletes) the transaction log when a checkpoint is done, either at shutdown or as a result of a checkpoint scheduled by the server. This provides a way to automatically limit the growth of the transaction log. Checkpoint frequency is still controlled by the CHECKPOINT_TIME and RECOVERY_TIME options (also definable on the command line).

The -m option is useful where high volume transactions requiring fast response times are being processed, and the contents of the transaction log are not being relied upon for recovery or replication. When this option is selected, there is no protection provided against media failure on the device containing the database files.

To avoid database file fragmentation, it is recommended that where this option is used, the transaction log be placed on a separate device or partition from the database itself.

This option is the same as the -m server option, but applies only to the current database or the database identified by the *database-file* command-line variable.

Note Do not use the -m option with databases that are being replicated, as replication inherently relies on transaction log information. For this reason, *never* use the -m option on a multiplex database.

-n name Provides a nickname, an alternate name for the database. Using the -n nickname simplifies connections. For Open Client the -n nickname must be the same as the entry in the *interfaces* file.

Both database servers and databases can be named. Since a database server can load several databases, the database name is used to distinguish the different databases. However, *Sybase strongly recommends that you run only one database on an IQ server.* If you must run two databases, start two IQ database servers on different ports.

By default, the database receives as a name the file name with the path and extension removed. For example, if the server is started on *c:\sybase\ASIQ-12_6\demo\asiqdemo.db* and no -n option is specified, then the name of the database is asiqdemo.

For naming conventions, see -n server option.

Note *The -n option is positional.* After a database file name, this option names the database. If the option does not follow a database file name, it names the server.

Recovery options

-iqdroplks name Allows the sp_iqcheckdb stored procedure to recover leaked storage within the IQ portion of the specified database. Use this option only while recovering leaked storage. Do not use this option during normal operations. The *dbname* must be the physical database name, not a logical name or nickname.

-iqfrec name Marks the specified database as in use and restores the IQ portion of the database to its last known consistent state. Use this option only while force recovering a database. Do not use this option during normal operations. The *dbname* must be the physical database name, not a logical name or nickname.

Note The options -iqfrec and -iqdroplks apply only to the IQ part of the database, not to the Catalog Store. -iqfrec does *not* enable a forced recovery on the Adaptive Server Anywhere part of the database (the Catalog Store).

Be sure to follow correct procedures when using -iqfrec and -iqdroplks. See *Sybase IQ Troubleshooting and Error Messages Guide* for details.

See also

Chapter 2, "Running Sybase IQ" in *Sybase IQ System Administration Guide*

Stopping the database server

There are several ways to stop the database server, some of which are platform-specific. This section describes the two operating system level server shutdown utilities dbstop and stop_asiq. For a full discussion of all of the server shutdown methods and when you need to stop the database server, see the section "Stopping the database server" in Chapter 2, "Running Sybase IQ" of the *Sybase IQ System Administration Guide*.

The Stop utility (dbstop)

The interactive Stop utility dbstop stops a database server.

The Stop utility is a command-line utility only, available on both UNIX and Windows. On Windows platforms, you can quickly stop a database server by clicking Shutdown on the server window or by right-clicking the server icon in the taskbar and choosing Exit from the popup menu.

Syntax

dbstop [*options*] *server-name*

Parameters

Table 1-6: dbstop options

Option	Description
-c *"keyword=value; ..."*	Connection parameters
-o *filename*	Log output messages to a file
-q	Quiet mode—do not print messages
-x	Do not stop if there are active connections
-y	Stop without prompting even if there are active connections

Usage

In UNIX, dbstop can shut down a server on any node on the network. The *name* is necessary to specify the name of the server that you wish to stop. However, *any* connection parameters you specified to start the server *must also* be specified with dbstop to stop the server. Without the proper connection parameters, dbstop does not know how to connect to the server to tell it to shutdown.

The dbstop command line options let you control whether a server is stopped, even if there are active connections.

server-name Name of the server to stop.

-c When stopping a network server, you must supply a connection string with a user ID that has permissions to stop the server. By default, DBA permission is required on the network server, and all users can shut down a personal server, but the -gk server command-line option can be used to change this.

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

The behavior of dbstop can be controlled if there are active connections on a server. If there are active connections, dbstop provides a prompt asking if you wish to shut down the server. If you specify `unconditional=true` on the command line, the server shuts down without prompting, even if there are active connections.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not print a message if the database was not running.

Do not stop if there are active connections (-x) Do not stop the server if there are still active connections to the server.

Stop without prompting (-y) Stop the server even if there are still active connections to the server.

Examples

To stop a database named asiqdemo on the server myserver:

```
dbstop -c "uid=DBA;pwd=SQL;eng=myserver;dbn=asiqdemo"
```

To stop a server myserver, regardless of the database running:

```
dbstop -c
"uid=DBA;pwd=SQL;eng=myserver;dbn=utility_db"
```

See also

Chapter 2, "Running Sybase IQ" in *Sybase IQ System Administration Guide*

The stop_asiq utility (UNIX only)

In addition to using the dbstop command, on UNIX platforms you can also stop the database server using the stop_asiq command. For full details on when to use stop_asiq, see the section "Stopping the database server" in Chapter 2, "Running Sybase IQ" in the *Sybase IQ System Administration Guide*.

Syntax

stop_asiq [*-agent* | *-stop* { one | all }]

Usage

When used in a UNIX operating system command line, stop_asiq is an interactive utility that lists all of the servers owned by other users, lists the server(s) you own, then asks you if you want to stop your server. If you respond `Y` (yes), stop_asiq shuts down your server and closes all users connections to your server. If you respond `N` (no), stop_asiq returns to the operating system command prompt level and your server continues to run.

Note that when stop_asiq is used, the following message appears:

```
"Please note that 'stop_asiq' will shutdown a server
```

```
    completely without regard for users connections or
    load processes status. For a finer level of detail
    the utility 'dbstop' has the options to control
    whether a server is stopped based on active
    connections."
```

Normally, you should not shut down a server while the server is still connected to one or more clients. If you try this, you get a warning that any uncommitted transactions will be lost. Disconnect or close all the clients and try again.

The -agent option shuts down the IQ Agent. on UNIX. On Windows, the IQ Agent runs as a service and the Service Manager administers it. For more information about stopping the IQ Agent, see "Troubleshooting IQ Agent startup"in "Running Sybase IQ," *Sybase IQ System Administration Guide*.

You can also use stop_asiq in a UNIX cron or at job with the -stop option. No operator prompting occurs and no operator action is required when the -stop option is specified.

-stop one In a UNIX cron or at job, shut down a single server, when exactly one running server was started by the user ID that starts the cron or at job.

-stop all In a UNIX cron or at job, shut down all servers that were started by the user ID that starts the cron or at job.

Note You must specify the full pathname to the stop_asiq executable in the cron statement.

Example

The following example illustrates using stop_asiq interactively to list all the servers that are running, then shut down one server.

```
% stop_asiq
Checking system for ASIQ 12 Servers ...
The following 3 server(s) are owned by other users.

##     Owner    PID   Started  CPU_Time
 -- ---------  -----  --------  --------  -----------------------

        hsin  19895    Mar.21      1:33
start_asiq -c 32m -gd all -gl all -gm 10 -gn 25 -gp 4096 -ti 4400
-n hsin -x tcp
qadaily  24754  01:25:07   1286:53
start_asiq -gn 25 @/exp1/new.cfg asiqdemo.db
-o /exp1/qa
wb  28350    Apr.11      0:20
start_asiq -gn 25 @asiqdemo.cfg -o
/exp1/wb/mysybase12/asiq1
```

```
The following 1 server(s) are owned by 'janed'
##     Owner    PID   Started  CPU_Time
-- ---------  -----  --------  --------  ----------------------
1:     janed   2838  15:11:37      0:07
start_asiq -c 32m -gd all -gm 10 -gn 25 -gp 4096 -ti 4400 -tl 300
@asiqdemo.cfg

--
    Please note that 'stop_asiq' will shutdown a server completely
    without regard for users connections or load processes status.
    For a finer level of detail the utility 'dbstop' has the options
    to control whether a server is stopped based on active connections.

Do you want to stop the server displayed above <Y/N>? Y

Shutting down server (2838) ...
Checkpointing server (2838) ...
Server shutdown.
```

See also Chapter 2, "Running Sybase IQ" in *Sybase IQ System Administration Guide*

Troubleshooting the database server

If you have problems with the operation of the server, including startup, shutdown, unresponsiveness, and abnormal termination, refer to Chapter 1, "Troubleshooting Hints" in the *Sybase IQ Troubleshooting and Error Messages Guide* for troubleshooting information.

CHAPTER 2

Using Interactive SQL (dbisql)

This chapter describes Interactive SQL (dbisql) in depth. In addition to the dbisql syntax and parameters, this chapter includes a tutorial on the dbisql tasks starting Sybase IQ, opening the sample database, accessing online help, using the dbisql command window, cancelling commands, and stopping dbisql.

Introduction to dbisql

Interactive SQL (dbisql) is a utility for sending SQL statements to the database server. You can use it for:

- Browsing the information in a database.
- Trying out SQL statements that you plan to include in an application.
- Loading data into a database and carrying out other administrative tasks.

In addition, dbisql can run command files. You can build repeatable scripts to run against a database and then use dbisql to execute these scripts. For more information, see "Saving, loading, and running command files" on page 54.

dbisql and dbisqlc

dbisql is a Java-based application. In this book, Interactive SQL always refers to this Java-based application.

All new Interactive SQL development is on dbisql. dbisql has several advantages over the Windows-based Interactive SQL application shipped with pre-12.4.3 versions of Sybase IQ:

- Find server capability — The Connect dialog now features a Find button beside the Server Name box on the Database tab. This allows you to browse and select the server name.
- Mouse buttons on UNIX — You can now select items from the dbisql menus using the mouse buttons.
- Cut and paste capability —available from both the main menu Edit item and a dropdown that appears when you select and right-click on text in the SQL Statements window.

To start dbisql, type dbisql at a UNIX command prompt or choose Interactive SQL Java from the Sybase IQ 12 program group on Windows.

Sybase IQ still includes the previous version of Interactive SQL in case you need a feature not yet implemented in Interactive SQL Java. The name of the previous version has been changed from dbisql to dbisqlc. See the section "The Interactive SQL Classic (dbisqlc) utility" on page 65 for dbisqlc syntax.

To start the previous version, enter dbisqlc at a UNIX command prompt or choose Interactive SQL Classic from the Sybase IQ 12 program group on Windows. For information specific to dbisqlc, see "The Interactive SQL Classic (dbisqlc) utility" on page 65.

The Interactive SQL (dbisql) utility

dbisql provides an interactive environment for database browsing and for sending SQL statements to the database server.

dbisql allows you to type SQL commands or run command files. It also provides feedback about the number of rows affected, the time required for each command, the execution plan of queries, and any error messages.

You can start dbisql in the following ways:

- from Sybase Central, using the Open Interactive SQL menu item.
- from the Start menu by choosing Start → Programs → Sybase → Adaptive Server IQ 12.6 → Interactive SQL Java.
- at a command prompt, using the *dbisql* command.

Opening Interactive SQL using the dbisql command-line utility

Syntax

dbisql [*options*] [*dbisql-command* | *command-file*]

Parameters

Table 2-1: dbisql options lists the available options for dbisql.

Table 2-1: dbisql options

Option	Description
-c *"keyword=value; ..."*	Supply database connection parameters
-codepage *codepage*	Specify a codepage to use when reading or writing files
-d *delimiter*	Use the given string as the command delimiter
-d1	Print statements as they are executed [command-prompt mode only]
-datasource *dsn-name*	Supply an ODBC data source
-f	Open (without running) the file called *filename*.
-host *hostname*	Specify the hostname or IP address of the machine running a database server.
-jConnect	Use jConnect to connect to the database
-nogui	Run in command-prompt mode
-ODBC	Use the iAnywhere JDBC driver to connect to the database
–onerror { continue | exit }	Override the ON_ERROR option for all users
-port *portnumber*	Look on the specified port number for the database server.
-q	Run in quiet mode — No windows or messages.
-x	Run in syntax check mode — No commands executed.

Usage

Note that for Sybase IQ 12, the Help menu item does not open online help for dbisql, but help is available in Sybase Central. In Sybase Central, choose Help → Sybase IQ Help, then click the Index tab. Type dbisql in the Topic or Keyword box and click the Go To button.

Exit codes are 0 (success) or non-zero (failure).

Interactive SQL requires that the QUOTED_IDENTIFIER database option be set to ON since a number of database functions, including some statements, rely on this setting to function properly. Interactive SQL automatically sets it ON when connecting to a database.

This utility does *not* accept @filename parameters.

dbisql-command | command-file If *dbisql-command* is specified, then dbisql executes the command. You can also specify a command file name. If no *dbisql-command* is specified, dbisql enters the interactive mode where you can type a command into a command window. From the Windows Start menu, you can enter interactive mode by selecting Programs → Sybase → Adaptive Server IQ 12.6 → Interactive SQL Java.

–c "keyword=value; ..." Specify connection parameters. See Chapter 4, "Connection and Communication Parameters" in the *Sybase IQ System Administration Guide* for a description of the connection parameters. If this option is not specified, the environment variable SQLCONNECT is used. If Interactive SQL cannot connect, you are presented with a dialog box where you can enter the connection parameters.

Note Sybase recommends that, to avoid ambiguity, you always specify connection parameters for dbisql instead of relying on defaults, whether you specify them in a command line or an initialization file such as .odbc.ini. If more than one database is started on a server, for example, you should specify the database name, and in a network with subnets, specify the communications protocol parameter with host number. For details, see Chapter 3, "Sybase IQ Connections" in the *Sybase IQ System Administration Guide*.

–codepage Specify the codepage to use when reading or writing files. The default code page is the default code page for the platform you are running on.

For example, on an English Windows machine, Interactive SQL uses the 1252 (ANSI) code page. If you want Interactive SQL to read files created using the 297 (IBM France) code page, specify the following option.

```
-codepage 297
```

For a list of supported code pages, see "Supported code pages" in *Sybase IQ System Administration Guide*.

–d Specify a command delimiter. Quotation marks around the delimiter are optional, but required when the command shell itself interprets the delimiter in some special way.

Command delimiters are used for all connections in that Interactive SQL session, regardless of the setting stored in the database (for the user, or the PUBLIC setting).

–d1 (The final character is a number 1, not a lower-case L). Interactive SQL echoes all statements it executes to the Command window (STDOUT). This can provide useful feedback for debugging SQL scripts, or when Interactive SQL is processing a long SQL script.

–datasource Specify an ODBC data source to connect to. You do not need to be using the iAnywhere JDBC driver to use this option. However, if the data source to which you are connecting is not configured to use TCP/IP, you must use the iAnywhere JDBC driver to connect. Sybase IQ data sources are configured to use TCP/IP by default.

–f *filename* Open (but do not run) the file called *filename*. The file name can be enclosed in quotation marks, and MUST be enclosed in quotation marks if the file name contains a blank. If the file does not exist, or if it is really a directory instead of a file, Interactive SQL prints an error message to the console and then quits. If the file name does not include a full drive and path specification, it is assumed to be relative to the current directory.

–host Specify the hostname or IP address of the computer on which the database server is running. You can use the name localhost to represent the current machine.

–jConnect Use the Sybase jConnect JDBC driver to connect to the database. This is the method recommended in most circumstances.

–nogui Run Interactive SQL in a command-prompt mode, with no windowed user interface. This is useful for batch operations. If you specify either *dbisql-command* or *command-file*, then –nogui is assumed.

In this mode, Interactive SQL sets the program exit code to indicate success or failure. On Windows operating systems, the environment variable ERRORLEVEL is set to the program exit code. The exit codes are listed in *Table 2-2: dbisql program exit codes*.

Table 2-2: dbisql program exit codes

Program Exit code	Description
0	Success.
1	General failure. At some point, a SQL or Interactive SQL statement did not execute successfully and the user chose to stop executing SQL statements. Alternatively, Interactive SQL noted an internal error.
5	User terminated Interactive SQL. When an error occurs during execution, the user is prompted to ignore it, stop, or exit Interactive SQL. If the user opts to exit, the program returns code 5. Code 5 is also returned if an error occurs and the Interactive SQL option ON_ERROR is set to EXIT.
9	Unable to connect.
255	Bad command line. The command line contained incomplete or invalid switches.

–ODBC Connect using the iAnywhere JDBC driver. (The iAnywhere JDBC driver was formerly called the JDBC-ODBC bridge.)

–onerror Controls what happens if an error is encountered while reading statements from a command file. This option overrides the ON_ERROR setting. It is useful when using Interactive SQL in batch operations.

For more information, see "ON_ERROR option [DBISQL]" in *Sybase IQ Reference Manual*.

–port Specify the port number on which the database server is running. The default port number for Sybase IQ is 2638.

–q Do not display output messages. This option is useful only if you start Interactive SQL with a command or command file. Specifying this option does not suppress error messages.

If you run Interactive SQL Java (dbisql) with the -q option and data extraction commands (primarily setting the option TEMP_EXTRACT_NAME1 to an output file) are in a command file, you must first set and make permanent the dbisql option "Show multiple result sets." If this option is not set, the data extraction output file is not created.

To set the "Show multiple result sets" option, click Tools → Options → Results in the dbisql window, then check the box "Show multiple result sets" and click "Make permanent."

–x Scan commands but do not execute them. This option is useful for checking long command files for syntax errors.

Examples

The examples in this section illustrate using dbisql at the system command prompt level. The rest of the dbisql sections in this chapter discuss how to use dbisql interactively.

- The following command, entered at a system prompt, runs the command file mycom.sql against the current default server, using the user ID DBA and the password SQL. If there is an error in the command file, the process terminates.

  ```
  dbisql -c "uid=DBA;pwd=SQL" –onerror exit mycom.sql
  ```

- The following command, when entered on a single line at a command prompt, adds a user to the current default database:

  ```
  dbisql -c "uid=DBA;pwd=SQL" grant connect to joe
  identified by passwd
  ```

See also

For detailed descriptions of SQL statements and dbisql commands, see Chapter 6, "SQL Statements" in the *Sybase IQ Reference Manual*.

Starting dbisql

In this section, you can practice starting IQ and dbisql.

You can start Interactive SQL in two ways: from Sybase Central or on its own. The way that you start Interactive SQL on its own depends on your operating system.

For detailed information on connecting to databases, see Chapter 3, "Running and Connecting to Servers" in the *Introduction to Sybase IQ*.

❖ **To open Interactive SQL from Sybase Central**

1 In the left pane, select the Sybase IQ plug-in.

2 In the right pane, click the Utilities tab.

3 Double-click Open Interactive SQL in the right pane.

  The Interactive SQL window appears.

4 Supply parameters following the steps in "Supplying connection parameters" on page 43.

Tip
You can also access Interactive SQL by choosing Start → Programs → Sybase → Adaptive Server IQ 12.6 → Interactive SQL Java, or from within Sybase Central by choosing any of the following options:

- Choosing Tools → Sybase IQ → Open Interactive SQL.
- Selecting a database in the left pane, and choosing Open Interactive SQL from the File menu.
- Right-clicking a database, and choosing Open Interactive SQL from the pop-up menu.
- Right-clicking a stored procedure, and choosing Execute from Interactive SQL from the pop-up menu. Interactive SQL opens with a CALL to the procedure in the SQL Statements pane and executes the stored procedure.

❖ **To start Interactive SQL from Sybase Central:**

- To start Interactive SQL, right-click the server and choose Interactive SQL from the pop-up menu. In this case, Interactive SQL automatically connects to the database.

  To start Interactive SQL without a connection to a database, choose Sybase IQ 12 → Utilities → Open Interactive SQL. The Connect dialog automatically appears.

❖ **To start dbisql on UNIX:**

1. Start the sample database server, if it isn't already running. To do this, change to the demo directory and use the following command format:

   ```
   % start_asiq -n servername asiqdemo.db
   ```

   For *servername*, substitute a unique server name, for example, your system name, database name, and port number, separated by an underscore.

   Note If you specify -n *servername* without a dbname, it connects to the default database on the current server. If you specify -n *dbname* without a *servername*, it connects to that database on the current server.

   For example, enter the following commands at system prompts:

   ```
   cd $SYBASE/ASIQ-12_6/demo
   start_asiq -n fiona_asiqdemo_1870 asiqdemo.db
   ```

2 To start dbisql, type the following command at a system prompt: dbisql

❖ **To start dbisql on Windows:**

- From the Start menu, choose Programs → Sybase → Adaptive Server IQ 12.6 → Interactive SQL Java.

  The Connect dialog appears.

Note Once the database server is running, you can also start dbisql by typing *dbisql* at the Command Prompt or by right-clicking the server name in Sybase Central and choosing Interactive SQL from the pop-up menu.

Supplying connection parameters

When no database is connected, Sybase IQ displays the Connect dialog box to request connection parameters.

❖ **To connect to a database:**

1 Identify yourself in the Connect dialog box on the Identification tab, using any of these methods:

- Enter a User ID and password. The password is case sensitive; the User ID is not. Press TAB after entering each.

  User ID — For the sample database, use the user ID DBA

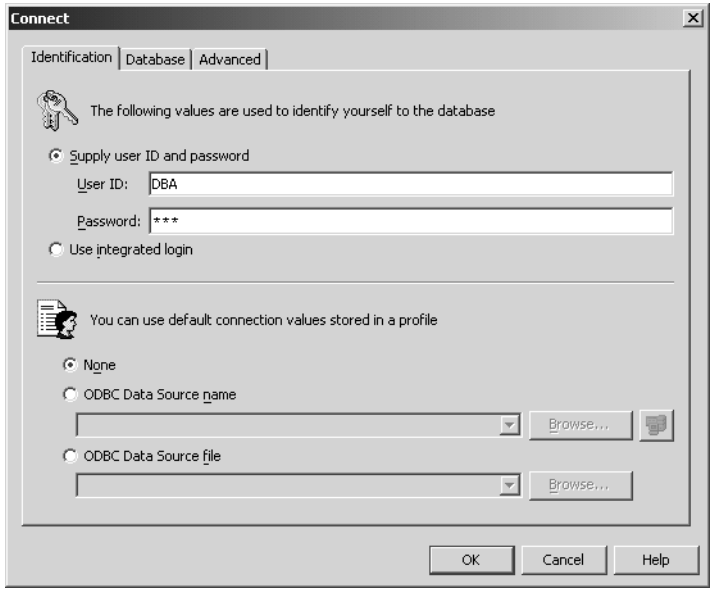
  Password — For the DBA user ID, use the password

  SQL

- Choose Integrated login.

  To use this option, the DBA must define an integrated login for you. For more information, see "Using an integrated login" in *Sybase IQ System Administration Guide*.

Figure 2-1: dbisql Connect dialog box



The password does not appear when you type it. This prevents someone else from seeing your password.

2 After entering the password, click the Database tab.

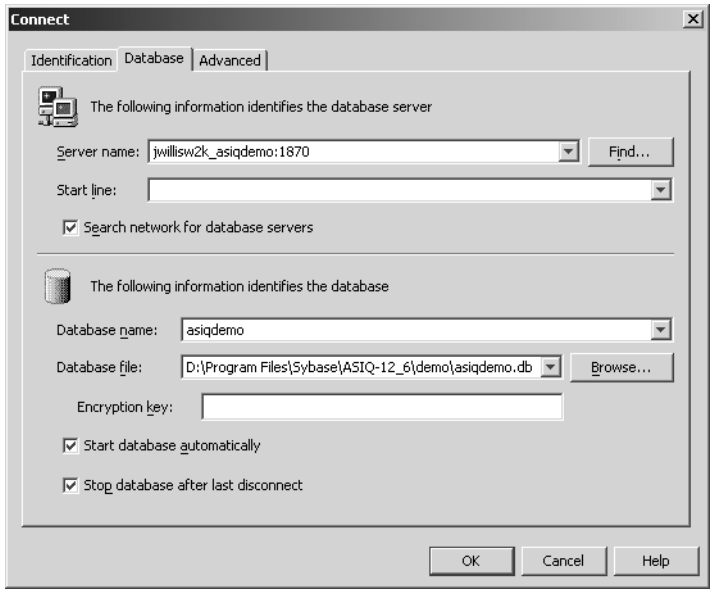
3 Enter the Server name you used when starting the server. Be sure that it is unique on your local area network.

For remote servers, specify the server as *host name*:*port number*.

4 Tab to the Database name and enter the database filename, without any suffix.

5 Tab to the Database file and enter the full pathname to the .db file. You can use the Browse key to locate the file, as shown in Figure 2-2.

Figure 2-2: dbisql Connect dialog Database tab



You can bypass the Connect dialog box by supplying the connection parameters on the command line with the -c option as shown in "Examples" on page 41.

You can also click Cancel to start Interactive SQL with no database connected. You can then enter a CONNECT command in the SQL Statements window. For example, to connect to the sample database on a local server, you could use:

```
CONNECT USING 'UID=DBA;PWD=SQL'
```

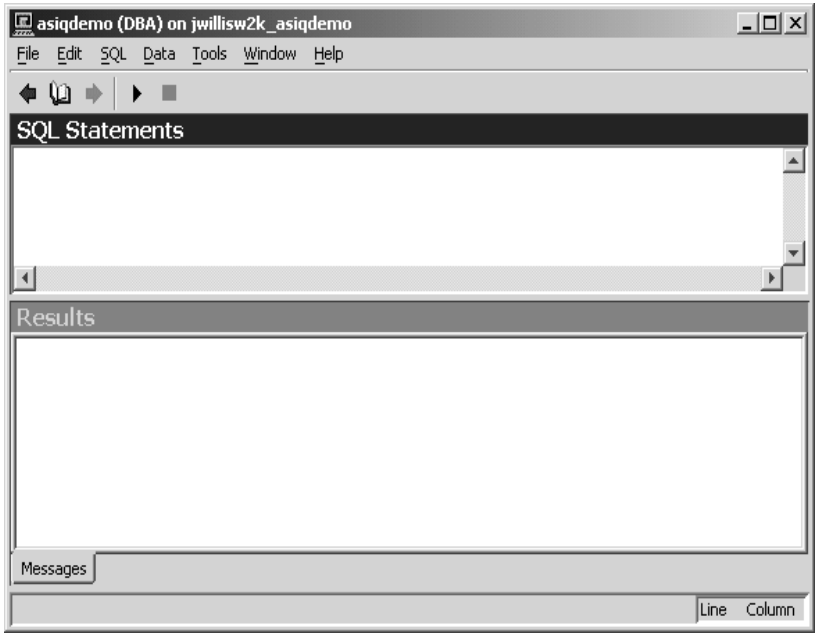
If the CONNECT dialog or an error message about missing information pops up, you may need to enter the -host and -port or other missing information in the Advanced tab. If your database is on a remote server, enter the -host and -port parameters on separate lines, as in:

```
-host fiona
-port 1870
```

For a complete list of connection parameters, see Chapter 4, "Connection and Communication Parameters" in the *Sybase IQ System Administration Guide*.

After connecting to the database, Interactive SQL displays the database name, user ID, and server name for the connection on the title bar, as shown in Figure 2-3.

Figure 2-3: dbisql window



Note For Sybase IQ 12, the Help menu item does not open dbisql online help, but dbisql help is available in Sybase Central. In Sybase Central, choose Help → Sybase IQ Help, then click the Index tab. Type dbisql in the Topic or Keyword box and click the Go To button.

Main dbisql window description

Interactive SQL has the following panes:

- **SQL Statements** provides a place for you to type SQL statements. If the code you type exceeds the size of the pane, scroll bars appear automatically.
- **Messages** displays information about execution. You can specify Messages options from Tools → Options dialog, including whether to display it as a separate pane or as a tab on the Results pane (the default).

- **Results** displays the results of commands that you execute. For example, if you use SQL statements to retrieve specific data, this pane displays the columns and rows that match the search criteria. If the information exceeds the size of the pane, scroll bars appear automatically.

Opening multiple windows

You can open multiple Interactive SQL windows, each corresponding to a separate database connection. Use multiple windows to connect simultaneously to two (or more) different databases on different servers or to open concurrent connections to a single database.

❖ **To open a new Interactive SQL window:**

1. Choose File → New Window.
2. In the resulting Connect dialog, enter connection options, and click OK to connect.

You can also connect to or disconnect from a database with the Connect and Disconnect commands in the SQL menu, or by executing a CONNECT or DISCONNECT statement.

Keyboard shortcuts

Interactive SQL provides the keyboard shortcuts listed in *Table 2-3: dbisql keyboard shortcuts*.

Table 2-3: dbisql keyboard shortcuts

Function key	Description
ALT+F4	Exits Interactive SQL.
ALT+LEFT CURSOR	Displays the previous SQL statement in the history list.
ALT+RIGHT CURSOR	Displays the next SQL statement in the history list.
CTRL+BREAK	Interrupts the SQL statement that is being executed.
CTRL+C	Copies the selected row(s) and column headings to the clipboard in the Results pane.
	In the SQL Statements pane, copies the selected text to the clipboard.
CTRL+END	Moves to the bottom of the current pane.
CTRL+H	Displays the history of your executed SQL.

Function key	Description
CTRL+HOME	Moves to the top of the current pane.
CTRL+N	Clears the contents of the Interactive SQL window.
CTRL+P	Prints the contents of the SQL Statements pane. You can configure the appearance of the printed text in the Interactive SQL Options dialog.
CTRL+Q	Displays the Query Editor. The Query Editor helps you build SQL queries. When you have finished building your query, click OK to export it back into the SQL Statements pane.
CTRL+S	Saves the contents of the SQL Statements pane.
ESC	Clears the SQL Statements pane.
F1	Opens Help.
F2	Edits the selected value in the result set. You can tab from column to column within the row.
F5	Executes all text in the SQL Statements pane. You can also perform this operation by clicking the Execute SQL Statement button on the toolbar.
F7	Displays the Lookup Table Name dialog. In this dialog, you can find and select a table and then press ENTER to insert the table name into the SQL Statements pane at the cursor position. Or, with a table selected in the list, press F7 again to display the columns in that table. You can then select a column and press ENTER to insert the column name into the SQL Statements pane at the cursor position.
F8	Displays the Lookup Procedure Name dialog. In this dialog, you can find and select a procedure and then press ENTER to insert the procedure name into the SQL Statements pane at the cursor position.
F9	Executes the text that is selected in the SQL Statements pane. If no text is selected, all of the statements are executed.
PGDN	Moves a page down in the current pane.
PGUP	Moves a page up in the current pane.
SHIFT+F5	Displays the plan for the statement in the SQL Statements pane without executing the statement.

The keyboard shortcuts in Table 2-4 are available when the SQL Statements pane has the focus:

Table 2-4: dbisql keyboard shortcuts for SQL Statements pane

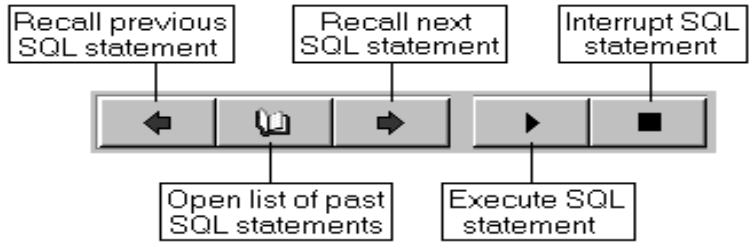
Function key	Description
CTRL+]	Moves the cursor to the matching brace. Brace matching matches parentheses, braces, brackets, and angle brackets.
CTRL+BACKSPACE	Deletes the word to the left of the cursor.

Function key	Description
CTRL+DEL	Deletes the word to the right of the cursor.
CTRL+G	Opens the Go To dialog where you can specify the line you want to go to.
CTRL+L	Deletes the current line from the SQL Statements pane and puts the line onto the clipboard.
CTRL+SHIFT+]	Extends the selection to the matching brace. Brace matching matches parentheses, braces, brackets, and angle brackets.
CTRL+SHIFT+L	Deletes the current line.
CTRL+SHIFT+U	Changes the selection to upper case characters.
CTRL+U	Changes the selection to lower case characters.
F3	Finds the next occurrence of the selected text.
HOME	Moves the cursor to the start of the current line or to the first word on the current line.
SHIFT+F3	Finds the previous occurrence of the selected text.
SHIFT+HOME	Extends the selection to the start of the text on the current line.

Using the Interactive SQL toolbar

The Interactive SQL toolbar (shown in Figure 2-4) appears at the top of the Interactive SQL window. This toolbar provides you with buttons for executing common commands.

Figure 2-4: Interactive SQL toolbar



With the buttons on this toolbar, you can:

- Recall the executed SQL statement immediately before your current position in the history list.
- View a list of up to 50 previously executed SQL statements.

- Recall the executed SQL statement immediately after your current position in the history list.
- Execute the SQL statement currently showing in the SQL Statements pane.
- Interrupt the execution of the current SQL statement.

As an easy reminder of what these buttons do, you can hold your cursor over each button to see a pop-up description.

Using Interactive SQL to display data

One of the principal uses of Interactive SQL is to browse the information in databases. This section shows how to query the information in the sample database.

You can display database information using the SELECT statement in Interactive SQL. The following example shows the command to type in the SQL Statements pane. Once you have typed the command, you must click the Execute SQL Statement button on the toolbar to carry out the command.

After you execute the statement, the data (called a result set) appears in the Results pane. You can use the scroll bars to see areas of the table that are outside your current view of the pane.

❖ **To list all the columns and rows of the employee table:**

1 Start Interactive SQL and connect to the sample database.

2 Type the following in the SQL Statements pane:

```
SELECT *
FROM employee
```

3 On the toolbar, click the Execute SQL Statement button.

Emp_id	Manager_id	Emp_lname	emp_fname	...
102	501	Fran	Whitney	...
105	501	Matthew	Cobb	...
129	902	Philip	Chin	...
148	1293	Julie	Jordan	...
160	501	Robert	Breault	...
...				

For more information on SELECT statements, see Chapter 1, "Selecting Data from Database Tables" in the *Sybase IQ Performance and Tuning Guide*.

Working with commands in Interactive SQL

The following chapters describe some of the commands you can use in Interactive SQL. This section describes general tasks for working with commands in Interactive SQL.

All SQL statements can be entered as commands in the top pane of the Interactive SQL viewer. When you are finished typing, you need to execute the statement to run it.

❖ **To enter a command:**

- In the SQL Statements pane, type the command.

❖ **To execute a command, do one of the following:**

- Press the Execute SQL Statement button *or* choose SQL → Execute *or* press F5.

❖ **To clear the SQL Statements pane:**

- Choose Edit → Clear SQL *or* press ESCAPE.

Tips

You can execute only a selection of text in the SQL Statements button by pressing F9.

Syntax error messages appear in the ISQL Error pop-up window. To see more information on an error, click Help. (Do not use the Online Books button in the Help window.)

Other messages about an executed statement, such as execution times, can appear in the Messages pane. You control the display and content of Message information in Tools → Options → Messages.

Combining multiple statements

The Interactive SQL environment allows multiple statements to be entered at the same time. This can be done by ending each statement with a semi-colon (;).

❖ **To enter multiple statements in SQL Statements pane:**

1. Try entering the following three commands into the SQL Statements pane.

```
UPDATE employee
SET dept_id = 400,
    manager_id = 1576
WHERE emp_id = 467;

UPDATE employee
SET dept_id = 400,
    manager_id = 1576
WHERE emp_id = 195;

SELECT *
FROM employee
WHERE emp_id IN ( 195, 467 );
```

2. On the toolbar, click the Execute SQL Statement button. All three statements are executed. After execution, the commands remain in the SQL Statements pane. If you want to clear this pane, press the ESC key.

Looking up tables, columns, and procedures

While you are entering commands in Interactive SQL, you can look up the names of tables, columns, or procedures stored in the current database and insert them at your cursor position.

❖ **To look up the names of tables in the database:**

1. Choose Tools → Lookup Table Name.
2. Find and select the table.
3. Click OK to insert the table name into the SQL Statements pane.

❖ **To look up column names in the database:**

1. Choose Tools → Lookup Table Name.
2. Find and select the table containing the column.

3 Click Show Columns.

4 Select the column and click OK to insert the column name into the SQL Statements pane.

❖ **To look up the names of procedures in the database:**

1 Choose Tools → Lookup Procedure Name.

2 Find and select the procedure.

3 Click OK to insert the procedure name into the SQL Statements pane.

In the tables and procedures lookup dialogs, you can enter the first few characters of the table or procedure you are looking for. After you type something in the field, the dialog waits a short time, then narrows the list to include only those items that start with the text you entered.

You can use the standard SQL wild card metacharacter `%` to mean "match anything". For example, to list only those tables ending in order, type `%order` into the look up field. To list all items containing the word java, type `%java%`. Clearing the search criteria from the look up field resets the filtering to display all the items.

Recalling commands

When you execute a command, Interactive SQL automatically saves it in a history list that lasts for the duration of the current session. Interactive SQL maintains a record of up to 50 of the most recent commands.

You can view the entire list of commands in the Command History dialog. To access the Command History dialog, press CTRL + H, or click the book icon in the toolbar.

The most recent commands appear at the bottom of the list. To recall a command, highlight it and click OK. It will appear in the SQL Statements pane of Interactive SQL.

You can also recall commands without the Command History dialog. Use the arrows in the toolbar to scroll back and forward through your commands, or press ALT + RIGHT ARROW and ALT + LEFT ARROW.

You can also save commands in text files, so that you can use them in a subsequent Interactive SQL session. as described in the next section.

Saving, loading, and running command files

You can save the commands currently present in the SQL Statements pane so that they are available for future Interactive SQL sessions. The file in which you save them is called a command file (also known as a script).

When you begin a new session, you can load the contents of a command file into the SQL Statements pane, or you can run the contents immediately.

❖ **To save the commands in the SQL Statements pane:**

1. Choose File → Save As.
2. In the Save dialog, specify a location, name and format for the file. Click Save when finished.

❖ **To load commands from a file into the SQL Statements pane:**

1. Choose File → Open.
2. In the Open dialog, find and select the file. Click Open when finished.

❖ **To run a command file immediately:**

1. Choose File→Run Script.
2. In the Open dialog, find and select the file. Click Open when finished.

You can supply a command file as a command line argument for Interactive SQL.

The Run Script menu item is the functional equivalent of a READ statement. For example, in the SQL Statements pane, you can also run a command file by typing:

```
READ 'c:\filename.sql'
```

where *c:\filename.sql* is the path, name, and extension of the file. You can enclose the string in single quotation marks (as shown) if you want, but they are only required if the path contains embedded spaces.

Logging commands

With the Interactive SQL logging feature, you can record commands as you execute them. Interactive SQL continues to record until you stop the logging process, or until you end the current session. The recorded commands are stored in a log file.

❖ **To begin logging Interactive SQL commands:**

1 Choose SQL → Start Logging.

2 In the Save dialog, specify a location and name for the log file.

3 Click Save when finished.

❖ **To stop logging Interactive SQL commands:**

- Choose SQL → Stop Logging.

Tips

You can also start and stop logging by typing in the SQL Statements pane. To start logging, type and execute START LOGGING *'c:\filename.sql'*, where *c:\filename.sql* is the path, name, and extension of the log file. A log file must have the .SQL extension. You only need to include the single quotation marks if the path contains embedded spaces. To stop logging, type and execute STOP LOGGING.

Once you start logging, all commands that you try to execute are logged, including ones that do not execute properly.

Canceling an Interactive SQL command

An Interrupt or Stop operation stops current processing and prompts for the next command or action.

❖ **To interrupt an Interactive SQL command:**

- On the toolbar, click the Interrupt the SQL statement button.

  or

  Choose SQL → Stop.

If a command file was being processed, you are prompted for an action to take (Stop command file, Continue, or Exit Interactive SQL). These actions can be controlled with the Interactive SQL ON_ERROR option. For more information, see ON_ERROR option [DBISQL] in Chapter 2, "Database Options" in the *Sybase IQ Reference Manual*).

Reported errors

When an abort is detected, one of three different errors will be reported depending upon when the abort is detected.

1 If the abort is detected when Interactive SQL is processing the request (as opposed to the database engine), then the following message is displayed:

ISQL command terminated by user

Interactive SQL stops processing immediately and the database transaction is left alone.

2 If the cancel is detected by the database engine while processing a standard data manipulation command (SELECT, INSERT, and DELETE) and the engine is not running in bulk operations mode, then the following message is displayed.

Statement interrupted by user.

The effects of the current command are undone, but the rest of the transaction is left intact.

3 If the abort is detected while the database engine is processing a data definition command (CREATE, DROP, ALTER, etc.), the following message appears:

```
Terminated by user -- transaction rolled back
```

Since data definition commands all perform a COMMIT automatically before the command starts, the effect of the ROLLBACK is to just cancel the current command.

This message also occurs when the database engine is running in bulk operations mode executing a command that modifies the database (INSERT, UPDATE, and DELETE). In this case, ROLLBACK cancels not only the current command, but everything that has been done since the last COMMIT. In some cases, it may take a considerable amount of time for the database engine to perform the automatic ROLLBACK.

Configuring Interactive SQL

You can configure different Interactive SQL viewer options in the Options dialog. This dialog provides settings for commands, viewer appearance, import/export features and messages.

Each option can be set from the interface, or using the Interactive SQL SET OPTION statement. For more information, see SET OPTION statement [DBISQL] in *Sybase IQ Reference Manual*.

To access the Options dialog:

- Choose Tools → Options.

The tabs of the Options dialog are described in the following sections.

Note Some Interactive SQL options are for use with Adaptive Server Anywhere only, and are not discussed here.

Options dialog: General tab

The Commands tab of the Options dialog has the following components:

- **Commit** The following options let you select when to commit changes to the database. You can also commit manually by entering an explicit COMMIT command whenever appropriate.
  - **After each command** Select this option to commit changes to the database after each SQL statement is executed.
  - **On exit** Select this option to commit changes to the database when you exit your Interactive SQL session. This is the default setting.

    For more information, see "AUTO_COMMIT option [DBISQL]" in Chapter 2, "Database Options" in the *Sybase IQ Reference Manual*.
- **Command files** The following options control Interactive SQL's behavior when running command files.
  - **When an error occurs** Select an option to control how Interactive SQL responds when it encounters an error while executing statements from a command file. Depending on the option you choose, Interactive SQL can continue executing the file, stop executing the file, or shut down. The default setting is PROMPT.

    For more information about configuring how Interactive SQL responds to errors while executing statements from a command file, see the ON_ERROR option in Chapter 2, "Database Options" in the *Sybase IQ Reference Manual*.
  - **Echo command files to log** If you select this option, Interactive SQL logs SQL statements that are executed from command files to the log file.

    By default, command files are copied to the log.

    For more information, see the ECHO option in Chapter 2, "Database Options" in the *Sybase IQ Reference Manual*.

- **When browsing for files, where should the browser start?** Select one of the following options to specify which directory Interactive SQL uses first when browsing for files. If you wish to use this setting in subsequent Interactive SQL sessions, click Make Permanent after selecting one of the following options.
  - **Last folder used** If you select this option, the initial browser directory is the one where you last used the file browser. This is the default setting.
  - **Current folder** If you select this option, the initial directory used by the browser is the current folder, as defined by the operating system.
- **File association** On Windows platforms, you can make Interactive SQL the default editor for *.SQL* files.
  - **Make DBISQL the default editor for .SQL files** Select this option to make Interactive SQL the default editor for *.SQL* files on Windows.

    Windows uses Interactive SQL to automatically open the file when you double-click on it. Note that Interactive SQL does not run the file automatically.

Options dialog: Results tab

The Results tab of the Options dialog in Interactive SQL has the following components:

- **Display null values as** Specify how you want nulls to display in table columns. You can use any string for this value. The default value is (NULL). If this field is blank, null values appear as an empty string.

  For more information, see NULLS option [DBISQL] in Chapter 2, "Database Options" in *Sybase IQ Reference Manual*.
- **Maximum number of rows to display** Specify the maximum number of rows that appear in the Results pane. The default is 500.
- **Truncation length** Specify the number of characters that are displayed in each column in the Results pane. If you enter a value of 0, the columns are not truncated. The default is 256.

  For more information, see TRUNCATION_LENGTH option [DBISQL] in Chapter 2, "Database Options."

- **Show multiple result sets** Select this option if you want Interactive SQL to display multiple result sets in the Results pane when you execute a procedure that returns multiple SELECT statements. Each result set appears on a separate tab in the Results pane. By default, Interactive SQL does not display multiple result sets.

  If you use the jConnect driver and select the Show Multiple Result Sets option, Interactive SQL must wait until the entire result set is retrieved before any rows appear in the Results pane. This can result in slower processing of large result sets.

- **Show row number**

  Select this option if you want row numbers to appear beside your results in the Results pane. This option is on by default.

- **Automatically refetch results** Select this option if you want Interactive SQL to automatically regenerate the result set after you execute an INSERT, UPDATE, or DELETE statement. By default, Interactive SQL refetches result sets.

  For more information, see the AUTO_REFETCH option in Chapter 2, "Database Options" in the *Sybase IQ Reference Manual*.

- **Which font do you want to use to show table data?** Select one of the following options to specify the font that is used for table data in the Interactive SQL Results pane.

  - **System** Select this option to use the machine's normal text font. This is the default setting.
  - **Editor** Select this option to use the same font as the Code Editor.

    For more information about the Code Editor, see "Format tab" on page 62.
  - **Custom** Select this option to specify the font, font style, and point size you want to use. Click Browse to select the desired settings in the Font dialog.

Options dialog: Import/Export tab

The Import/Export tab of the Options dialog in Interactive SQL has the following components:

- **Default export format** To choose a format for exporting files, select a file format from the dropdown list. By default, the export format is ASCII.
- **Default import format** This option is not supported for use with Sybase IQ databases.

  For more information about the file formats supported by Interactive SQL, see Chapter 7, "Moving Data In and Out of Databases," *Sybase IQ System Administration Guide*.

  Note The IQ data extraction facility exports data in binary or ASCII format, which can be loaded into another database. Use this facility for high-volume data movement, or when you need an output file that can be used for loads. See "Data extraction options" in Chapter 7, "Moving Data In and Out of Databases," *Sybase IQ System Administration Guide* for details and advantages of this feature.

- **ASCII Options** Specify the default symbols used for the field separator, quote string, and escape character when you import or export data in ASCII format.
  - **Default field separator** The symbol used to separate values in ASCII files. The default value is a comma (,).
  - **Default quote string** The symbol used to enclose strings in ASCII files. The default value is a single quote (').
  - **Default escape character** The symbol used in place of unprintable characters in ASCII files. The escape character must be one, single-byte character. The default value is a backslash (\).
  - **Default encoding** The encoding used when importing and exporting files. If you change this value, it is only changed for the current Interactive SQL session. When you start a new Interactive SQL session, it is restored to its default value. The default value is (Default). If you select (Default), the encoding is determined as follows:
    - the code page specified in the ENCODING clause of the INPUT, OUTPUT, or READ statement
    - the code page specified with the DEFAULT_ISQL_ENCODING option (if this option is set)
    - the code page specified with the -codepage command-line option when Interactive SQL was started

- the default code page for the computer Interactive SQL is running on

Options dialog: Messages tab

The Messages tab of the Options dialog in Interactive SQL has the following components:

- **Measure execution time for SQL statements** Select this option if you want Interactive SQL to measure the time it takes for a statement to execute. The time appears on the Messages tab. By default, this option is selected.
- **Show separate Messages pane** Select this option if you want information from the database server, such as execution time, to appear in the Messages pane between the SQL Statements and Results panes rather than on the Messages tab in the Results pane. By default, database server information appears on the Messages tab in the Results pane.
- **Default number of lines in Messages pane** Type the number of lines you want returned in the Messages pane. The default number is 7. If you select Show Separate Messages Pane, the number is also the height (in lines) of the Messages pane.

Options dialog: Editor tab

This tab allows you to configure the appearance of text typed in the SQL Statements pane. Note that any settings you specify on this tab also apply to the Code Editor when it is used in Sybase Central.

The Editor tab of the Options dialog in Interactive SQL consists of four tabs: Editor, Tabs, Format, and Print.

Editor tab

The Editor tab has the following components:

- **Vertical scroll bar** Shows or hides a vertical scroll bar when the window is too small to contain all the text.

- **Horizontal scroll bar** Shows or hides a horizontal scroll bar when the window is too small to contain all the text.

Tabs tab

The Tabs tab has the following components:

- **Tab size** Lets you set the tab size (in number of spaces).
- **Indent size** Lets you set the size of indents (in number of spaces).
- **Insert spaces** Inserts *n* spaces when you press Tab instead of inserting one tab character. The value for *n* will be between one and the number of indent size spaces, depending on how many spaces are required to move the cursor forward to the next tab stop.
- **Keep tabs** Inserts a tab character into the document and moves the cursor forward to the next tab stop when you press Tab.
- **Auto indent** Lets you set the auto indent feature. You have the following options:
  - **None** disables the feature.
  - **Default** uses the tab and indent sizes that are set.
  - **Smart** uses previous lines of code as a guideline for indenting open and closing braces.

    •Indent opening brace Select this option if you wish opening braces to be indented. This option is enabled when the Smart option is selected.

    •Indent closing brace Select this option if you wish closing braces to be indented. This option is enabled when the Smart option is selected.
- **Sample** The Sample field shows an example of how code is formatted based on the options you choose for indenting opening and closing braces.

Format tab

The Format tab has the following components:

- **Text Highlighting** Lets you specify the color and style of different types of text in the main editing window. Choose a type of text and then set the foreground, background, and style for that text type.
  - **Foreground** Foreground refers to the color of the text.

- **Background** Background refers to the color of the screen behind the text.
- **Style** Lets you specify the type of formatting for a text type. You have the following choices:
  - Plain
  - Italic
  - Bold
  - Italic and Bold

- **Font size** Lets you specify the font point size of the text that appears in the SQL Statements pane.
- **Caret color** Lets you specify the color of the blinking on-screen cursor indicator.
- **Sample** Shows an updated sample of the text with the settings you configure above.
- **Reset All** Returns all settings to their default values.

Print tab

The Print tab has the following components:

- **Header** Lets you specify what information appears in the header, as well as how it is formatted, when you print the contents of the SQL Statements pane. The header text is left aligned by default. Press the > button for a list of available options.
- **Footer** Lets you specify what information appears in the footer, as well as how it is formatted, when you print the contents of the SQL Statements pane. The footer text is left aligned by default. Press the > button for a list of available options.
  - **> button** Pressing the > button lets you choose from the following options for the header or footer:
    - File Name
    - File Time
    - File Date
    - Page Number
    - Page Count

- Current Time
- Current Date
- Left Align
- Center
- Right Align

The items that you select do not all need to have the same alignment. For example, you can choose to left align the file name and right align the date in the header. By default, all text in both the header and the footer is left aligned. You must specify the alignment before the type of text. For example, if you want the file name to be centered in the header, type &C&F in the Header field, or press the > button and select the Center option and then press the > button again and select the Filename option.

In addition to specifying these options, you can type text that you want to appear in the header and footer fields. For example, if you type Page &P of &p in the Footer field, Page 1 of 1 appears in the footer of the printed document.

- **Font size** Lets you select the font point size for the printed text.

Options dialog: Query Editor tab

This tab allows you to configure settings for the Query Editor.

The Query Editor tab of the Interactive SQL Options dialog has the following components:

- **Fully qualify table and column names** Select this option if you want table and column names to be fully qualified with their owner names when constructing queries in the Query Editor.
- **Quote names** Select this option if you want the names of identifiers to be enclosed in double quotes when constructing queries in the Query Editor.

  For information about quoting identifiers, see "Identifiers" in Chapter 3, "SQL Language Elements," *Sybase IQ Reference Manual*.

Leaving dbisql

When you have finished working with dbisql, the EXIT command returns you to the operating system.

If you leave dbisql now, you will have to restart dbisql to continue with the tutorial.

The Interactive SQL Classic (dbisqlc) utility

Note The dbisql command now runs an enhanced version of Interactive SQL that is a Java application. The Java version has many advantages, including multiple windows, each opening a separate database connection. The version of the Interactive SQL utility described in this section is now called dbisqlc to distinguish it from the Java version. For information about the Java version, see "The Interactive SQL (dbisql) utility" on page 36.

dbisqlc provides the user with an interactive environment for database browsing and for sending SQL statements to the database server.

dbisqlc allows you to type SQL commands, or run command files. It also provides feedback about the number of rows affected, the time required for each command, the execution plan of queries, and any error messages.

Syntax

dbisqlc [*options*] [*dbisqlc-command* | *command-file*]

Parameters

Table 2-5: dbisqlc options lists the available options for dbisqlc.

Table 2-5: dbisqlc options

Option	Description
-c *"keyword=value; ..."*	Supply database connection parameters
-d *delimiter*	Specify command delimiter
-q	Quiet mode—no windows or messages
-r	Returns the error "Not enough fields allocated in sqlda," if the defined result set of the stored procedure does not match the actual result set.
-x	Syntax check only—no commands executed

Usage

Note that for Sybase IQ 12, the Help menu item does not open online help for dbisqlc, but help is available in Sybase Central. In Sybase Central, choose Help → Sybase IQ Help, then click the Index tab. Type dbisqlc in the Topic or Keyword box and click the Go To button.

If you would like to customize dbisqlc, change to your home directory ($HOME) and issue the following command to copy the terminfo extension file default.tix into it:

```
% cp $ASDIR/tix/default.tix .
```

This file controls key sequences for dbisqlc and improves the command window display. For more information, see "Shift and Control Keys" on page 72.

dbisqlc-command | command-file If *dbisqlc-command* is specified, then dbisqlc executes the command. You can also specify a command file name. If no *dbisql-command* is specified, dbisqlc enters the interactive mode where you can type a command into a command window. From the Windows Start menu, you can enter interactive mode by selecting Programs → Sybase → Adaptive Server IQ 12.6 → Interactive SQL Classic.

-c "keyword=value; ..." Specify connection parameters. See Chapter 4, "Connection and Communication Parameters" in the *Sybase IQ System Administration Guide* for a description of the connection parameters. If this option is not specified, the environment variable SQLCONNECT is used. If Interactive SQL cannot connect, you are presented with a dialog box where you can enter the connection parameters.

Note Sybase recommends that, to avoid ambiguity, you always specify connection parameters for dbisqlc instead of relying on defaults, whether you specify them in a command line or an initialization file such as .odbc.ini on UNIX, or *odbc.ini* on Windows. If more than one database is started on a server, for example, you should specify the database name, and in a network with subnets, specify the communications protocol parameter with host number. For details, see Chapter 3, "Sybase IQ Connections" in the *Sybase IQ System Administration Guide*.

-d Specify a command delimiter. By default, the delimiter is the semi-colon.

-q Do not display output messages. This option is useful only if you start Interactive SQL with a command or command file.

-r Returns the error "`Not enough fields allocated in sqlda`" if the defined result set of the stored procedure does not match the actual result set. This option is useful when querying stored procedures.

-x Scan commands but do not execute them. This option is useful for checking long command files for syntax errors.

Examples

The examples in this section illustrate using dbisqlc at the system command prompt level. The rest of the dbisqlc sections in this chapter discuss how to use dbisqlc interactively.

- The following command, entered at a system prompt, runs the command file *mycom.sql* against the current default server, using the user ID *DBA* and the password *SQL*. Note that the default password is case sensitive for a case sensitive database; the user ID is not.

  ```
  dbisqlc -c "uid=DBA;pwd=SQL" mycom.sql
  ```

- The following command, entered at a system prompt, adds a user to the current default database:

  ```
  dbisqlc -c "uid=dba;pwd=SQL" grant connect to joe
  identified by passwd
  ```

See also

For detailed descriptions of SQL statements and dbisql commands, see Chapter 6, "SQL Statements" in the *Sybase IQ Reference Manual*.

Starting dbisqlc

Some users may prefer to use the older version of Interactive SQL that shipped with versions of IQ prior to 12.6. Once a database server is running, you can connect to it from dbisqlc. This example describes connecting to the sample database asiqdemo.

❖ **To start dbisqlc and connect to a database from the command prompt:**

1 Ensure that the sample database is loaded on a running server, as described in "Starting the database server" in Chapter 3, "Running and Connecting to Servers" of the *Introduction to Sybase IQ*.

2 Start dbisqlc by entering:

```
dbisqlc -c "uid=DBA;pwd=SQL"
```

at the command prompt on your UNIX or Windows system.

The –c parameter specifies connection parameters. For a complete list of connection parameters, see Chapter 4, "Connection and Communication Parameters" in the *Sybase IQ System Administration Guide*.

Note The default user ID and password for the asiqdemo database and other IQ databases are "DBA" and "SQL." You must always type the password in uppercase, because the default setting is CASE RESPECT. (User IDs are never case sensitive.) However, if you connect to a database created with Adaptive Server Anywhere, the default may be lowercase. If you have trouble connecting to a database, try changing the case of the user ID and password that you supply.

❖ **To start dbisqlc and connect to a database on Windows**

1 Start the server, as described in "Starting the database server" in Chapter 3, "Running and Connecting to Servers" of the *Introduction to Sybase IQ*.

2 Select Start → Programs → Sybase → Adaptive Server IQ 12.6 → Interactive SQL Classic

The dbisqlc Connect window appears.

3 On the Login tab, enter the user ID

```
DBA
```

and the password

```
SQL
```

This is the default user ID and password for Sybase IQ databases when they are created. (The password does not appear when you type it. This prevents someone else from seeing your password.) By default, the password is case sensitive for a case sensitive database; the user ID is not.

4 Click the Database tab and type the server name (for example, "*hostname*_asiqdemo" for the demo database).

5 On the Network tab, select TCPIP.

If your database is on a remote machine, you must add host information in the space beside TCPIP by typing "host=*servername*:*nnnn*" where *servername* is the name of your system and *nnnn* is your port number. (The default port number is 2638, but if the host was started with a different number, use that instead.)

6 Click OK to connect to the database.

The dbisqlc command window

After you enter the dbisqlc command and connect to a database, the dbisqlc command window appears. The appearance and functionality of this window are different on UNIX and Windows systems.

dbisqlc Command window on UNIX

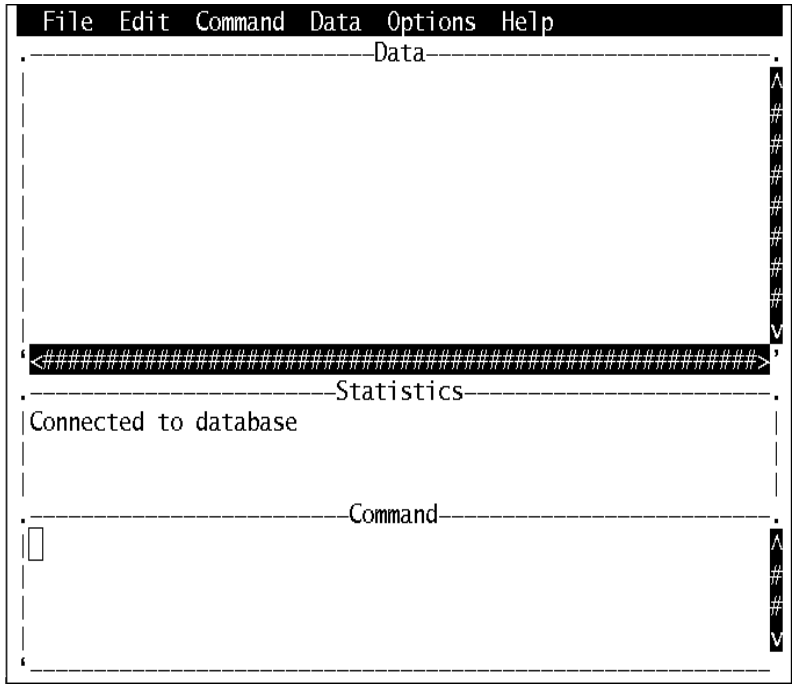
If you run Interactive SQL Classic on a Windows system, skip to the section "dbisqlc Command window on Windows" on page 74.

The dbisqlc window on UNIX is divided into three sections:

- Data section, at the top
- Statistics section, below it
- Command section, at the bottom

If you follow the instructions in "Starting dbisqlc" on page 68 correctly, you see the message "Connected to database" in the dbisqlc Statistics section.

Figure 2-5: dbisqlc window on UNIX



The dbisqlc command window on UNIX varies according to the operating system. For example, on Solaris systems, the Common Desktop Environment window differs from that on OpenWindows. The dbisqlc command window is divided into three subwindows. The top section is entitled Data, the middle Statistics, and the bottom section is Command, as shown in Figure 2-5.

For Sybase IQ 12, the Help menu item does not open online help for dbisqlc, but help is available in Sybase Central. In Sybase Central, choose Help Æ Sybase IQ Help, then click the Index tab. Type dbisqlc in the Topic or Keyword box and click the Go To button.

Pull-down menus

Tutorials throughout this chapter instruct you to choose items from pull-down menus. These menus are located at the top of the screen.

Executing Commands

Type commands in the Command section, and execute them by pressing the F9 function key on UNIX or the Execute button on Windows.

Note If your terminal does not support the function keys, type CTRL-F followed by the function key number. For example, for the F9 function key, you would type CTRL–F 9. If you still have trouble executing commands, set up the terminfo extension file as described in "Shift and Control Keys."

To enter multiple commands at once, separate them with semicolons.

You can store commands to an ASCII file by typing CTRL-A F to activate the File menu, and using the down arrow to choose File → Save from the menu bar.

To load from an ASCII file, type CTRL-A F and choose File → Open from the menu bar.

Function keys

dbisqlc uses some function keys and special keys on UNIX as described in Table 2-6.

Table 2-6: dbisqlc function and special keys on UNIX

Function key	Description
F5	Move data to the left by one column in the data window
SHIFT+F5	Move data to the left by one character
F6	Move data to the right by one column
SHIFT+F6	Move data to the right by one character
F7	Display a list of the tables in the database. The up and down arrow keys can be used to scroll through the table names changing the highlighted table name. With the list displayed, pressing enter will insert the current table name into the command window at the cursor position. The F7 key can be used while the table list is displayed, and a list of columns will be displayed for the highlighted table. Again, enter can be used to select the highlighted column name and put it into the command window at the cursor position.
CTRL+PgUp	Move to the top of data
CTRL+PgDn	Move to bottom of data

Shift and Control Keys

If necessary, key sequences can also be defined for SHIFT and CTRL to be applied to the next key (so that SHIFT-FunctionKey or CTRL-PgDown can be entered).

Some terminals do not allow dbisqlc to distinguish when the ALT key is being used. To enter an ALT- key in dbisqlc, use CTRL-A followed by the key to which the ALT modifier is to be applied. For example, to see the File menu, type CTRL-A f. If you use dbisqlc from a remote terminal, such as a VT100 emulator, you may be able to configure the terminal to use “emacs mode” so that it will send ALT-<key> as ESC <key>. dbisqlc will also recognize these escape sequences. You can type ESC <key> yourself, but they must be typed in quick succession or the keys will be interpreted individually.

If necessary, key sequences can also be defined for SHIFT and CTRL to be applied to the next key (so that SHIFT-<FunctionKey> or CTRL-PgDown can be entered).

These sequences are controlled by a terminfo extension (*.tix*) file. The dbisqlc utility first looks for *${TERM}.tix* in *${HOME}*, */opt/ASIQ-12_6/tix*, */opt/ASIQ-12_6/bin*, and then throughout *${PATH}*. If *${TERM}.tix* is not found, dbisqlc searches for *default.tix* in the same directories. A sample *default.tix* file comes with Sybase IQ.

Displaying data in dbisqlc

One of the principal uses of dbisqlc is to look at information in databases.

The database used in this tutorial is for a fictional company. The sample database contains information about employees, departments, sales orders, and so on.

All this information is organized into a number of tables, consisting of rows and columns.

To list the tables in a database

Sybase IQ provides a number of built-in functions called system stored procedures to display information about your database.

For example, you can list the tables in the sample database using the sp_iqtable stored procedure. To run a stored procedure, type its name (and any parameters) in the Command window.

For more information about system stored procedures, see Chapter 9, "System Procedures" in the *Sybase IQ Reference Manual*.

To list all the columns and rows of the employee table

To list the data stored within database tables, type a SELECT command in the dbisqlc command window. Once you have typed the command, choose the F9 function key (if supported) or Command → Execute from the menu to carry out the command. The example output that follows displays the first several columns and rows of the results of the query. (These appear in the dbisqlc data window.) The format is used throughout this manual.

Type the following:

```
SELECT *
FROM employee
```

emp_id	manager_id	emp_fname	emp_lname	...
102	501	Fran	Whitney	...
105	501	Matthew	Cobb	...
129	902	Philip	Chin	...
148	1293	Julie	Jordan	...
160	501	Robert	Breault	...
...				

Notes

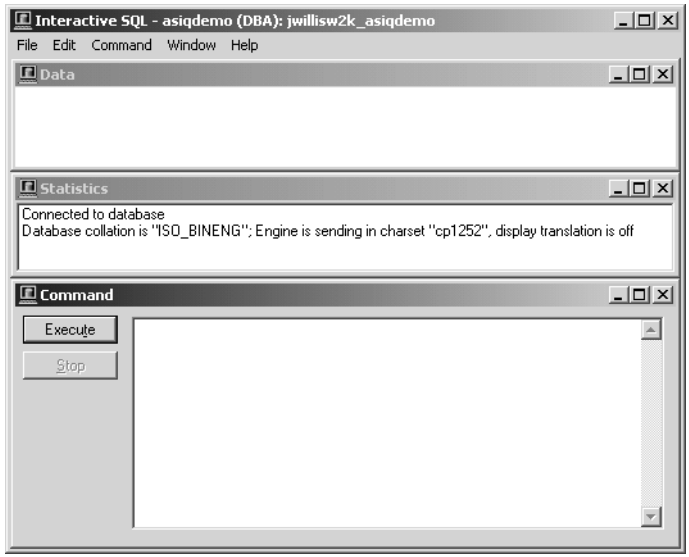
- SQL statements are case insensitive. SELECT is the same as select and the same as Select. In the examples, SQL keywords are shown in upper case, but you do not have to type them in upper case.
- SQL statements can be typed on more than one line. You can type the statements all on one line, or break them over several lines as you wish. Some SQL statements, such as the SELECT statement, consist of several parts, called clauses. In many examples, each clause is placed on a separate line, but you do not have to type them this way.

The dbisqlc Data window displays a set of rows and columns containing information about the employees. Each row contains information about one employee, and each column contains a particular piece of information for all employees.

dbisqlc Command window on Windows

After you start dbisqlc, the Interactive SQL Classic window opens. It is divided into three subwindows. The topmost is entitled Data, the second Statistics, and, at the bottom of your screen is Command, as shown in Figure 2-6.

Figure 2-6: dbisqlc window on Windows



For Sybase IQ 12, the Help menu item does not open online help for dbisqlc, but help is available in Sybase Central. In Sybase Central, choose Help → Sybase IQ Help, then click the Index tab. Type dbisqlc in the Topic or Keyword box and click the Go To button.

Entering commands

The Command window appears at the bottom of the dbisqlc screen. It is a standard edit control for typing dbisqlc commands. If you type more lines than will fit in this window, the window automatically scrolls. You can scroll the window using the cursor keys or the scroll bar on the right side of the window. This window can also be made larger and maximized to full screen size in the standard Windows fashion.

Execute commands by pressing the execute key (F9) or clicking the Execute button.

To enter multiple commands at once, separate them with semicolons. You can store commands to an ASCII file by choosing File → Save. To load them from an ASCII file, choose Open from the menu bar.

Displaying data in dbisqlc

One of the principal uses of dbisql is to look at information in databases.

The database used in this tutorial is for a fictional company. The sample database contains information about employees, departments, sales orders, and so on.

All this information is organized into a number of tables, consisting of rows and columns.

You display information from a database using the SELECT statement. The following example shows the command to type in the dbisqlc command window. Once you have typed the command, you must click Execute to carry out the command. The example displays the first several columns and rows of the results of the query, which are displayed in the dbisqlc data window. The format is used throughout this manual.

To list all the columns and rows of the employee table

Type the following:

```
SELECT *
FROM employee
```

emp_id	manager_id	emp_lname	emp_fname	...
102	501	Fran	Whitney	...
105	501	Matthew	Cobb	...
129	902	Philip	Chin	...
148	1293	Julie	Jordan	...
160	501	Robert	Breault	...
...				

Notes

- SQL statements are case insensitive. SELECT is the same as select and the same as Select. In the examples, SQL keywords are shown in upper case, but you do not have to type them in upper case.
- SQL statements can be typed on more than one line. You can type the statements all on one line, or break them over several lines as you wish. Some SQL statements, such as the SELECT statement, consist of several parts, called **clauses**. In many examples, each clause is placed on a separate line, but you do not have to type them this way.

The dbisqlc Data window displays a set of rows and columns containing information about the employees. Each row contains information about one employee, and each column contains a particular piece of information for all employees.

Scrolling the data window

When you type the command

```
SELECT * FROM employee
```

in the dbisqlc command window, the visible portion of the dbisqlc data window cannot hold the entire employee table.

The visible portion of the data window does not display all the information about each employee, and does not display the entire list of employees.

Viewing other columns

To see more information about each employee (that is, other columns) you use the scroll bar at the bottom of the data window. This is a standard Windows scroll bar.

Viewing other rows

To see more information on other employees (that is, other rows), use the scroll bar to the right of the data window. The employee table in the sample database has information on about 75 employees.

Sometimes, the vertical scroll bar behaves slightly differently than standard scroll bars, as the number of rows in the result may be unknown. In this case, a guess as to the number of rows is used. If dbisqlc determines that its guess is wrong, the guess is adjusted and the slider "jumps."

Command recall in dbisqlc

Let's execute another command.

1 Type the following:

```
SELECT * FROM department
```

2 Press F9.

The contents of the department database table are displayed in the Data window. As you execute commands with dbisqlc, they are saved in a command history.

3 To recall commands, choose Command → Recall from the menu bar. This activates the command recall window.

4 The command recall window displays the first line of the last 15 commands executed. Use the cursor up and down keys to scroll through the commands.

5 Position the cursor on the first command that you executed, which was:

```
SELECT *
FROM employee
```

and press the enter key. The cursor returns to the command window with the selected command in it. You can now re-execute that command or modify it to make a new command.

More recall keys

Table 2-7: dbisqlc recall keys lists the keys that can also be used to recall previous commands.

Table 2-7: dbisqlc recall keys

Key sequence	Description
Ctrl+r	Brings up the command recall window
Ctrl+p	Cycles backwards through previously executed commands. Retrieved commands are placed into the command window
Ctrl+n	Cycles forward through previously executed commands

Function keys

dbisqlc uses some function keys and special keys on Windows as listed in *Table 2-8: dbisqlc function and special keys on Windows*.

Table 2-8: dbisqlc function and special keys on Windows

Function key	Description
F5	Move data to the left by one column in the data window
Shift+F5	Move data to the left by one character
F6	Move data to the right by one column
shift+F6	Move data to the right by one character
F7	Display a list of the tables in the database. The cursor up and down keys can be used to scroll through the table names changing the highlighted table name. With the list displayed, pressing enter will insert the current table name into the command window at the cursor position. The F7 key can be used while the table list is displayed, and a list of columns will be displayed for the highlighted table. Again, enter can be used to select the highlighted column name and put it into the command window at the cursor position.
F9	Execute the command that is in the command window. This operation can also be performed with the mouse by clicking Execute.
F10	Activate the menus at the top of the window
Page Up	Move data up a page
Page Down	Move data down a page
Ctrl+PageUp	Move to top of data
Ctrl+PageDown	Move to bottom of data

Canceling a **dbisqlc** command

The Stop button is used to cancel a command.

A Stop operation stops current processing and prompts for the next command. If a command file was being processed, you are prompted for an action to take: Stop command file, Continue, or Exit dbisqlc. These actions can be controlled with the dbisql ON_ERROR option (see Chapter 2, "Database Options" in the *Sybase IQ Reference Manual*).

Reported errors

When a command is canceled, one of three different errors will be reported depending upon when the end is detected.

1 If the cancellation is detected when dbisqlc is processing the request (as opposed to the database engine), then the following message is displayed:

```
dbisql command terminated by user
```

dbisqlc stops processing immediately and the database transaction is left alone.

2 If the cancellation is detected while the database engine is processing a data definition command (CREATE, DROP, ALTER, etc.), the following message appears:

```
Terminated by user -- transaction rolled back
```

Since data definition commands all perform a COMMIT automatically before the command starts, the effect of the ROLLBACK is to just cancel the current command.

This message also occurs when the database engine is running in bulk operations mode executing a command that modifies the database (INSERT and DELETE). In this case, ROLLBACK cancels not only the current command, but everything that has been done since the last COMMIT or ROLLBACK. In some cases, it may take a considerable amount of time for the database engine to perform the automatic ROLLBACK.

3 If the cancel is detected by the database engine while processing a standard data manipulation command (SELECT, INSERT, DELETE, etc.) and the engine is not running in bulk operations mode, then the following message is displayed.

```
Statement interrupted by user.
```

The effects of the current command are undone, but the rest of the transaction is left intact.

CHAPTER 3

Database Administration Utilities

About this chapter

Sybase IQ includes a set of utility programs in addition to the database server, for performing database administration tasks. This chapter provides reference information for most of the database administration utilities. Utility programs that start and stop the database server are described in Chapter 1, "Running the Database Server".

The following command-line utility programs are described in this chapter:

- "The CP874toUTF8 utility"
- "The Collation utility (dbcollat)"
- "The Data Source utility (iqdsn)"
- "The Information utility (dbinfo)"
- "The Log Translation utility (dbtran)"
- "The Ping utility (dbping)"
- "The Server Location utility (dblocate)"
- "The Transaction Log utility (dblog)"
- "The Validation utility (dbvalid)"

Administration utilities overview

This chapter presents reference information on the programs and database administration utilities that are part of Sybase IQ. The utilities can be accessed from Interactive SQL, or as command-line programs.

For an introduction to Interactive SQL, also called dbisql, see Chapter 2, "Using Interactive SQL (dbisql)".

For comprehensive documentation on Sybase Central, see the Sybase Central online Help. For an introduction to the Sybase Central database administration tool, see Chapter 3, "Running and Connecting to Servers" in the *Introduction to Sybase IQ*.

The administration utilities use a set of system environment variables. These variables are described in Chapter 1, "File Locations and Installation Settings" in the *Sybase IQ Reference Manual*.

File administration statements

A set of SQL statements are available that carry out some of the tasks that the administration utilities carry out. These statements are listed in Chapter 6, "SQL Statements" of the *Sybase IQ Reference Manual*.

The CP874toUTF8 utility

The CP874toUTF8 utility converts data in the CP874 character set into UTF8 collation, the only collation supported by Sybase IQ for the Thai language. You cannot load data in the CP874 character set. Instead, you must convert it to UTF8 using this utility.

Syntax

CP874toUTF8 *[CP874InputFile]*

Usage

You can run this utility from the command prompt only.

CP874toUTF8 reads the named file in the CP874 character set (or standard input if no files are named) and prints the UTF8 conversion to standard output.

Note Files with embedded NULL characters ('\0') are not converted correctly. Remove such characters before running this utility.

Use caution with large data files; the UTF8 output can be up to 3 times larger than the input data. Input and output file size must both be within operating system limits.

CP874toUTF8 returns a 0 exit status upon successful completion. A non-zero exit status indicates an error occurred. The CP874toUTF8 utility writes all error messages to stderr.

CP874toUTF8 calls the Sybase Unicode Infrastructure Library (Unilib®) to perform the data conversion. If Unilib cannot convert the data, CP874toUTF8 reports the following conversion errors:

```
uniconv_unistrFromS failed
```

```
uniutf8_unistrToUTF8 failed
```

If the conversion fails, divide the file into smaller sections for conversion, to isolate the point of failure. Contact Sybase Technical Support, if you are unable to resolve the problem.

The Collation utility (dbcollat)

The Collation command-line utility dbcollat extracts a collation (sorting sequence) into a file suitable for creating a database using a custom collation.

The file that is produced by dbcollat can be modified and used with Sybase Central or with the COLLATION option of CREATE DATABASE to create a new database with a custom collation.

Exit codes are 0 (success) or non-zero (failure).

You must change the label on the following line in the collation file. If you do not, the custom collation will conflict with the original collation on which it is based.

```
Collation label (name)
```

If you wish to create a custom collation but have not yet created a database, you should extract a collation from the sample database.

For more information on custom collating sequences, see Chapter 11, "International Languages and Character Sets" of the *Sybase IQ System Administration Guide*.

Syntax

dbcollat *[options] output-file*

Parameters

The following table lists the available options for the dbcollat utility.

Table 3-1: dbcollat options

Option	Description
-c *"keyword=value; ..."*	Supply database connection parameters
-d *filename*	Convert definition file to INSERT statement with collation mapping placed in *mapping-file*
-e	Include empty mappings
-o *filename*	Log messages to a file
-q	Quiet mode — do not print messages
-v	Show version of IQ software.
-x	Use hex for extended characters (7F-FF)
-y	Replace file without confirmation
-z *col-seq*	Specify collating sequence label

This utility accepts @filename parameters.

Usage

Connection parameters (-c) For a description of the connection parameters, see Chapter 4, "Connection and Communication Parameters" in the *Sybase IQ System Administration Guide*. If the connection parameters are not specified, connection parameters from the SQLCONNECT environment variable are used, if set.

For example, the following Windows command extracts a collation file from the asiqdemo database running on the *sample_server* server, and connects as user ID DBA with password SQL:

```
dbcollat -c
"eng=sample_server;dbn=asiqdemo;uid=DBA;pwd=SQL"
c:\sample\col
```

Convert the definition file to an INSERT statement (-d) When a database is created, the collation is inserted into the SYS.SYSCOLLATION system table. A mapping from the collation to character sets and Sybase TDS collations is also inserted into the SYS.SYSCOLLATIONMAPPINGS system table. This collation is selected from the set of provided collations in the *collseqs.sql* file or from the custom collations in the *custom.sql* file in the *scripts* subdirectory of your Sybase IQ installation directory.

For more information about the SYSCOLLATIONMAPPINGS system table, see "SYSCOLLATIONMAPPINGS system table" in *Sybase IQ Reference Manual*.

Custom collations are added to the *custom.sql* script. The -d option converts the collation file that you edited into an INSERT statement that can be copied into *custom.sql*.

For example, you can use the -d option with the dbcollat command as follows:

```
dbcollat -d coll-defn-file custom-file
```

The *coll-defn-file* is read and parsed as a collation definition. Output is written to *custom-file*. The *custom-file* contents must be added to *custom.sql*.

For more information about creating a custom collation using the –d option, see "Creating a custom collation" in *Sybase IQ System Administration Guide*.

Include empty mappings (-e) Normally, collations don't specify the actual value that a character is to sort to. Instead, each line of the collation sorts to one position higher than the previous line. However, older collations have gaps between some sort positions. Normally, the Collation utility skips the gaps and writes the next line with an explicit sort-position. This option causes the Collation utility to write empty mappings (consisting of just a colon (:)) for each line in the gap.

Log messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display messages on a window.

Show version (-v) Display the IQ version number.

Use hexadecimal for extended characters [7F to FF] (-x) Extended single-byte characters (whose value is greater than hex 7F) may or may not appear correctly on your screen, depending on whether or not the code page in use on your computer is the same as the code page of the collation you are extracting. This option causes the Collation utility to write any character to hex 7F or above as a two-digit hexadecimal number, in the form \x*dd*. For example:

```
\x80, \xFE
```

Without the –x option, only characters from hex 00 to hex 1F, hex 7F and hex FF are written in hexadecimal form.

Operate without confirming actions (-y) Choosing this option automatically replaces an existing collation file without prompting for confirmation.

Specify collating sequence label (-z) Specify the label of the collation to be extracted. The names of the collation sequences can be found by executing the following command: `dbinit -l`. If this option is not specified, then the Collation utility extracts the collation being used by the database. For more information about collations, see "Understanding collations" in Chapter 11, "International Languages and Character Sets" in *Sybase IQ System Administration Guide*.

If the -z option is specified with one of the available collation labels, then dbcollat does not connect to a database. Otherwise, it connects to a database and extracts the collation of that database. If the collation label does not match the collation label of the database, an error is returned.

If -z is not specified, the default collation is used. Normal ASCII (binary) ordering is used for the lower 128 characters. For the upper 128 characters (also called the extended characters), any character that is an accented form of a letter in the lower 128 are sorted to the same position as the unaccented form. The determination of whether or not an extended character is an accented letter is based upon code page 850 (multilingual code page).

For a list of available collation labels, see "Supplied and recommended collations" in Chapter 11, "International Languages and Character Sets" in *Sybase IQ System Administration Guide*.

Examples

The following Windows command extracts a collation file from the asiqdemo database running on the *sample_server* server, and connects as user ID DBA with password SQL:

```
dbcollat -c
"eng=sample_server;dbn=asiqdemo;uid=DBA;pwd=SQL"
c:\sample\col
```

The following command uses the -z option of dbcollat to extract the details of collation 850:

```
dbcollat -c "uid=DBA;pwd=SQL;eng=sample_server" -z 850
c:\sample\col
```

See also

For more information on custom collating sequences, see Chapter 11, "International Languages and Character Sets" of the *Sybase IQ System Administration Guide*.

The Data Source utility (iqdsn)

The Data Source utility is a cross-platform alternative to the ODBC Administrator for creating, changing, deleting, describing, and listing Sybase IQ ODBC data sources. On Windows operating systems, the data sources are held in the registry. On UNIX operating systems, the data sources are held in the *.odbc.ini* file. This command-line utility is useful for batch operations.

Syntax

iqdsn [*modifier-options*]
{ **–l** [**u**] [**s**] [**qq**]
| **–d** [**u**] [**s**]*dsn*
| **–g** [**u**] [**s**] *dsn*
| **–w** [**u**] [**s**] *dsn* [*details-options;...*]
| **–cl** [**qq**] }

Parameters

The following tables list the available options for the iqdsn utility.

Table 3-2: iqdsn major options

Major option	Description
–l [u] [s] [qq]	List either all Sybase IQ users or all Sybase IQ system data sources. You can also list both user and system data sources. User data sources is the default. Using –qq with this option lists the DSNs without any banner or titles.
–d [u] [s] *dsn*	Delete the named Sybase IQ user or Sybase IQ system data source. User data sources is the default.
–g [u] [s] *dsn*	List (get) details about the named Sybase IQ user or Sybase IQ system data source. User data sources is the default.
–w [u] [s] *dsn* [*details-options*]	Create (write) a user or system data source definition. User data sources is the default.
–cl [qq]	List available connection parameters. Using –qq with this option lists the available connection parameters without any banner or titles

Table 3-3: iqdsn modifier-options

Modifier option	Description
–b	Brief. Print connection string for the data source
–q	Quiet. Do not print banner
–v	Verbose. Print connection parameters in tabular form
-va	Verbose All. Print connection parameters in same format as -v, but also include other hidden parameters. Use this option to display ODBC driver qualifier needed for remote data access on those UNIX platforms that support such access, or for some third party driver managers.
–y	Delete or overwrite data source without confirmation

Table 3-4: iqdsn details-options

Details option	Description
–c "keyword=*value*;..."	Supply database connection parameters
–ec *encryption type*	Encrypt all network packets
–o *filename*	Write client message to filename
–p *size*	Set maximum network packet size
–r	Disable multiple record fetching
–tl *seconds*	Client liveness timeout period
–x *list*	List network drivers to run
–z	Display debugging information
server-name	Connect to named database server

Usage

The iqdsn modifier options can occur before or after the major option specification. The order makes a difference only when a connection parameter value is specified more than once. In such a case, the last value specified is used.

Major options

List defined data sources (–l) Lists the available Sybase IQ ODBC data sources. You can modify the list format using the –b or –v options. You can modify the option using the u (user) or s system) specifiers. The default specifier is u.

Delete the named data source (–d) Deletes the named data source. You can modify the option using the u (user) or s system) specifiers. u is the default specifier. If you supply -y, any existing data source is overwritten without confirmation.

List (get) details of the named data source (–g) List the definition of the named data source. You can modify the format of the output using the –b or –v option. You can modify the option using the u (user) or s system) specifiers. The default specifier is u.

Create (write) a data source definition (–w) Creates a new data source, or overwrites one if one of the same name exists. You can modify the option using the -u (user) or s system) specifiers. u is the default specifier. If you supply –y, any existing data source is overwritten without confirmation.

List available connection parameters (–cl) This convenience option lists the connection parameters supported by the iqdsn utility.

Modifier options

Print connection string for the data source (–b) Format the output of the list as a single line connection string.

Do not print banner (–q) Suppress the informational banner.

Do not print banner or titles (–qq) Suppress both the informational banner and titles. This option can only be used with the –l and the –cl options.

Print connection parameters in tabular form (–v) Format the output of the list over several lines, as a table.

Delete or overwrite data source without confirmation (–y) Automatically delete or overwrite each file without prompting you for confirmation.

Details options

Connection parameters (–c) Specify connection parameters as a connection string.

For more information, see Chapter 4, "Connection and Communication Parameters" in *Sybase IQ System Administration Guide*.

Encrypt network packets (–ec) Encrypt packets sent between the client application and the server.

For more information, see "Encryption connection parameter [ENC]" on page 146 in the *Sybase IQ System Administration Guide*.

Log output messages to file (–o) Write output messages to the named file. By default, messages are written to the console.

For more information, see "LogFile connection parameter [LOG]" on page 152 in the *Sybase IQ System Administration Guide*.

Operate quietly (–q) Do not display output messages. This option is available only from the command-line utility.

Set maximum network packet size (–p) The maximum packet size for network communications, in bytes. The value must be greater than 300, and less than 16000. The default setting is 1492.

For more information, see "CommBufferSize connection parameter [CBSize]" on page 138 in the *Sybase IQ System Administration Guide*.

Disable multiple-record fetching (–r) By default, when the database server gets a simple fetch request, the application asks for extra rows. You can disable this behavior by using this option.

For more information, see "DisableMultiRowFetch connection parameter [DMRF]" on page 144 in the *Sybase IQ System Administration Guide*.

Set client liveness timeout (–tl) Terminates connections when they are no longer intact. The value is in seconds.

The default is server setting, which in turn has a default of 120 seconds.

For more information, see "LivenessTimeout connection parameter [LTO]" on page 151 in the *Sybase IQ System Administration Guide*.

Set communications links (–x) A comma separated list of network drivers to run.

For more information, see "CommLinks connection parameter [Links]" on page 139 in the *Sybase IQ System Administration Guide*.

Display debugging information (–z) Provide diagnostic information on communications links on startup.

Server name Connect to the named server. Only the first 40 characters are used.

For more information, see Chapter 1, "Running the Database Server."

Examples

Write a definition of the data source newdsn. Do not prompt for confirmation if the data source already exists.

```
iqdsn -y -x tcpip -w newdsn -c "uid=DBA;pwd=SQL" -v
```

You can also change the order of options:

```
iqdsn -w newdsn -c "uid=dba;pwd=sql" -x tcpip -y
```

List all known user data sources, one data source name per line:

```
iqdsn -l
```

List all known system data sources, one data source name per line:

```
iqdsn -ls
```

List all data sources along with their associated connection string:

```
iqdsn -l -b
```

Report the connection string for user data source MyDSN:

```
iqdsn -g MyDSN
```

Report the connection string for system data source MyDSN:

```
iqdsn -gs MyDSN
```

Delete the data source BadDSN, but first list the connection parameters for BadDSN and prompt for confirmation:

```
iqdsn -d BadDSN -v
```

Delete the data source BadDSN without prompting for confirmation.

```
iqdsn -d BadDSN -y
```

Create a data source named NewDSN for the database server MyServer:

```
iqdsn -w NewDSN -c "uid=DBA;pwd=SQL;eng=bar"
```

If a NewDSN already exists, the previous definition is overwritten.

The following example connects to the sample database server. The server name sample overrides the previous specified value of MyServer:

```
iqdsn -w NewDSN-c "uid=DBA;pwd=SQL;eng=MyServer" sample
```

List all connection parameter names and their aliases:

```
iqdsn -cl
```

See also

"Working with ODBC data sources" and "Using ODBC data sources on UNIX" in Chapter 3, "Sybase IQ Connections" of the *Sybase IQ System Administration Guide*.

The Information utility (dbinfo)

The command-line Information utility dbinfo displays information about a database Catalog Store. The utility indicates when the database was created, the name of any transaction log file or log mirror that is maintained, the Catalog Store page size, the version of installed Java classes, and other information. Optionally, it can also provide Catalog table usage statistics and details. Note that if your database does not have a SYSCOLLATION table, the collation name is not returned.

Note The information returned by the dbinfo utility is only about the Catalog Store and does not reflect the IQ Store.

Syntax

dbinfo [*options*]

Parameters

The following table lists the available options for the dbinfo utility.

Table 3-5: dbinfo options

Option	Description
-c *"keyword=value; ..."*	Database connection parameters
-o *filename*	Log output messages to a file
-q	Operate quietly
-u	Output page usage statistics

Usage

Connection parameters (-c) Specify connection parameters. See Chapter 4, "Connection and Communication Parameters" in the *Sybase IQ System Administration Guide* for a description of the connection parameters.

Any valid user ID can run the Information utility, but to obtain page usage statistics you need DBA authority.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages.

Page usage statistics (-u) Display information about the usage and size of all Catalog Store tables, including system and user-defined tables.

You can only request page usage statistics if no other users are connected to the database.

The Log Translation utility (dbtran)

With the Log Translation utility, you can translate a transaction log into a SQL command file.

You can access the Log Translation utility in the following ways:

- From Sybase Central, using the Translate Log File wizard.
- At the command prompt, using the *dbtran* command. This is useful for incorporating into batch or command files.

Translating a transaction log using the Translate Log File wizard

❖ **To translate a transaction log into a command file**

1. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
2. In the right pane, click the Utilities tab.
3. In the right pane, double-click Translate Log File.

   The Translate Log File wizard appears.
4. Follow the instructions in the wizard.

Tip
You can also access the Translate Log File wizard by choosing Tools > Adaptive Server Anywhere 9 > Translate Log File.

You can also access the Translate Log File wizard from within Sybase Central using any of the following methods:

- Choosing Tools > Adaptive Server Anywhere 9 > Translate Log File.
- Selecting a database in the left pane, and choosing Translate Log File from the File menu.
- Right-clicking a database, and choosing Translate Log File from the popup menu.

Translating a transaction log using the dbtran command-line utility

Syntax

Running against a transaction log.

dbtran [*options*] *transaction-log* [*SQL-file*]

Running against a database server.

dbtran [*options*]

Option	Description
–a	Include uncommitted transactions
–c *"keyword=value*; …*"*	Supply database connection parameters—cannot be used with a transaction log name
–d	Display output in chronological order
–ek *key*	Specify encryption key
–ep	Prompt for encryption key
–f	Output only since the last checkpoint
–g	Include audit records in output
–ir *offset1*,*offset2*	Include only the portion of the log between the two specified offsets
–is *source*,…	Include only rows originating from the specified sources
–it *user.table*,…	Include only operations on specified tables by specifying a comma-delimited list of user.table
–j *date/time*	Output from the last checkpoint prior to the given time

Option	Description
–m	Specify transaction logs directory (requires -n option)
–n *filename*	Output SQL file, when used against a database server
–o *filename*	Log output messages to a file
–q	Run quietly, do not print messages
–r	Remove uncommitted transactions (default)
–rsu *username*,…	Override default Replication Server user names
–s	Produce ANSI standard SQL UPDATE transactions
–sr	Generate SQL Remote comments
–t	Include trigger-generated transactions in output
–u *userid*,…	Translate transactions for listed users only
–x *userid*,…	Exclude transactions for listed users
–y	Replace file without confirmation
–z	Include trigger-generated transactions as comments only
Transaction-log	Log file to be translated—cannot be used together with –c or -n
SQL-file	Output file containing the translated information—for use with *transaction-log* only

Description

The *dbtran* utility takes the information in a transaction log and places it as a set of SQL statements and comments into an output file. The utility can be run in the following ways:

- **Against a database server** Run in this way, the utility is a standard client application. It connects to the database server using the connection string specified following the -c option, and places output in a file specified with the -n option. DBA authority is required to run in this way.

  The following command translates log information from the server asiqdemo and places the output in a file named *asiqdemo.SQL*.

  ```
  dbtran -c "eng=asiqdemo;dbn=asiqdemo;uid=DBA;pwd=SQL"
  -n asiqdemo.sql
  ```

- **Against a transaction log file** Run in this way, the utility acts directly against a transaction log file. You should protect your transaction log file from general access if you wish to prevent users from having the capability of running this statement.

  ```
  dbtran asiqdemo.log asiqdemo.sql
  ```

When the *dbtran* utility runs, it displays the earliest log offset in the transaction log. This can be an effective method for determining the order in which multiple log files were generated.

Exit codes are 0 (success) or non-zero (failure).

For more information about the Log translation utility options, see "Log translation utility options" on page 95.

This utility accepts @filename parameters. For more information, see "Server command-line options" on page 7.

Log translation utility options

Include uncommitted transactions (–a) The transaction log contains any changes made before the most recent COMMIT by any transaction. Changes made after the most recent commit are not present in the transaction log.

Connection string (–c) When running the utility against a database server, this parameter specifies the connection string.

DBA authority is required to run *dbtran*.

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

Display output in chronological order (–d) Transactions are displayed in order from earliest to latest. This feature is provided primarily for use when auditing database activity: the output of this command should not be applied against a database.

Specify encryption key (–ek) This option allows you to specify the encryption key for strongly encrypted databases directly in the command. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way.

For strongly encrypted databases, you must specify either `-ek` or `-ep`, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

If you are running against a database server (using the `-c` option), make sure you specify the key using a connection parameter, not using the `-ek` option. For example, the following command gets the transaction log information about database *enc.db* from engine sample, and saves its output in *log.sql*.

```
dbtran -n log.sql -c
eng=sample;dbf=enc.db;uid=dba;pwd=sql;dbkey=mykey
```

Prompt for encryption key (–ep) This option allows you to specify in the command that you want to be prompted for the encryption key. This option causes a dialog box to appear, in which you enter the encryption key. It provides an extra measure of security by never allowing the encryption key to be seen in clear text.

For strongly encrypted databases, you must specify either `-ek` or `-ep`, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

If you are running against a database server (using the -c option), make sure you specify the key using a connection parameter, not using the `-ep` option. For example, the following command gets the transaction log information about database *enc.db* from engine sample, and saves its output in *log.sql*.

```
dbtran -n log.sql -c
eng=sample;dbf=enc.db;uid=dba;pwd=sql;dbkey=mykey
```

Output from last checkpoint only (–f) Only transactions that were completed since the last checkpoint are output.

Include audit information (–g) If the AUDITING database option is turned on, auditing information is added to the transaction log. You can include this information as comments in the output file using this option.

For more information, see "AUDITING option [database]" in Chapter 2, "Database Options" of the *Sybase IQ Reference Manual*.

Include rows from specified sources (–is) Isolate operations on rows that have been modified by operations from one or more of the following sources, specified as a comma-separated list:

- **All** All rows. This is the default setting.
- **SQLRemote** Include only rows that were modified using SQL Remote. You can also use the short form SR.
- **RepServer** Include only rows that were modified using the Replication Agent (LTM) and Replication Server. You can also use the short form RS.
- **Local** Include only rows that are not replicated.

Include offset range (–ir) Isolate a portion of the transaction log between two specified offsets.

Include specified tables (–it) Isolate those operations on the specified, comma-separated list of tables. Each table should be specified as *owner*.*table*.

Output from the last checkpoint prior to a given date (–j) Only transactions from the most recent checkpoint prior to the given date and/or time are translated. The user-provided argument can be a date, time or date and time enclosed in quotes. If the time is omitted, the time is assumed to be the beginning of the day. If the date is omitted, the current day is assumed. The following is an acceptable format for the date and time: *"YY/MMM/DD HH*:*NN"*.

Transaction logs directory (–m) Use this option to specify a directory that contains transaction logs. This option must be used in conjunction with the `-n` option.

Output file (–n) When you run the *dbtran* utility against a database server, use this option to specify the output file that holds the SQL statements.

Log output messages to file (–o) Write output messages to the named file.

Operate quietly (–q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Do not include uncommitted transactions (–r) Remove any transactions that were not committed. This is the default behavior.

Override Replication Server user names (–rsu) By default, the `-is` option assumes the default Replication Server user names of dbmaint and sa. You can override this assumption using the `-rsu` option with a comma-separated list of user names.

Generate ANSI standard SQL UPDATE (–s) If the option is not used, and there is no primary key or unique index on a table, the Translation utility generates UPDATE statements with a non-standard FIRST keyword in case of duplicate rows. If the option is used, the FIRST keyword is omitted for compatibility with the SQL standard.

Generate SQL Remote comments (–sr) Place generated comments in the output file describing how SQL Remote distributes operations to remote sites.

Include transactions generated by triggers (–t) By default, actions carried out by triggers are not included in the command file. If the matching trigger is in place in the database, when the command file is run against the database, the trigger will carry out the actions automatically. Trigger actions should be included if the matching trigger does not exist in the database against which the command file is to be run.

Output transactions for listed users only (–u) This option allows you to limit the output from the transaction log to include only specified users.

Output transactions except for listed users (–x) This option allows you to limit the output from the transaction log to exclude specified users.

Operate without confirming actions (–y) Choosing this option automatically replaces existing command file(s) without prompting you for confirmation.

Include transactions generated by triggers as comments only (–z) Transactions that were generated by triggers will be included only as comments in the output file.

The Ping utility (dbping)

The command-line Ping utility dbping assists in diagnosing connection problems.

Syntax

dbping [*options*]

Parameters

The following table lists the available options for the dbping utility.

Table 3-6: dbping options

Option	Description
–c *"keyword=value; ..."*	Supply database connection parameters
–d	Make a database connection if the server is found
–l *library*	Use the specified ODBC driver or driver manager library
–m	Use the ODBC Driver Manager. Otherwise, connect using embedded SQL.
–o *filename*	Log output messages to a file
–pc *property*,…	Report specified connection properties
–pd *property*,…	Report specified database properties
–ps *property*,…	Report specified database server properties
–q	Operate quietly—do not print messages
–z	Display debugging information

Usage

The dbping utility is a tool to help debug connection problems. It takes a full or partial connection string and returns a message indicating whether the attempt to locate a server or database, or to connect, was successful.

The utility can be used for embedded SQL or ODBC connections. It cannot be used for jConnect (TDS) connections.

Exit codes are 0 (success) or non-zero (failure).

This utility accepts @filename parameters.

Options

Connection parameters (-c) For a description of the connection parameters, see Chapter 4, "Connection and Communication Parameters" in the *Sybase IQ System Administration Guide*. If no connection parameters are specified, connection parameters from the SQLCONNECT environment variable are used, if set.

Make database connection (-d) Ping the database, not just the server.

If you do not supply the -d option, then dbping reports success if it finds the server specified by the -c option. If you do supply the -d option, then dbping reports success only if it connects to both server and database.

For example, if you have a server named blair running the database sample, the following succeeds:

```
dbping -c "eng=blair;dbn=sample"
```

The following command fails, with the message `Ping database failed -- specified database not found`

```
dbping -d -c "eng=blair;dbn=sample"
```

Load specified library (–l) Specify the library to use (without its file extension). This option avoids the use of the ODBC driver manager, and so is particularly useful on UNIX operating systems.

For example, the following command loads the ODBC driver directly:

```
dbping –m –c "dsn=ASIQ12 Sample" –l dbodbc9
```

On UNIX, if you wish to use a threaded connection library, you must use the threaded version of the Ping utility, *dbping_r*.

Use ODBC to connect (-m) Establish a connection using ODBC. By default, dbping attempts a connection using the embedded SQL interface.

Report connection properties (–pc) Upon connection, display the specified connection properties. Supply the properties in a comma-separated list. You must specify enough connection information to establish a database connection if you use this option.

For a list of connection properties, see "Connection-level properties" in *Adaptive Server Anywhere Database Administration Guide*.

For example, the following command displays the DIVIDE_BY_ZERO_ERROR option setting, which is available as a connection property.

```
dbping –c ... –pc Divide_by_zero_error
```

Report database properties (–pd) Upon connection, display the specified database properties. Supply the properties in a comma-separated list. You must specify enough connection information to establish a database connection if you use this option.

For a list of database properties, see "Database-level properties" in *Adaptive Server Anywhere Database Administration Guide*.

For example, the following command displays the Java version in use by the database:

```
dbping –c ... –pd JDKVersion
```

Report database server properties (–ps) Upon connection, display the specified database server properties. Supply the properties in a comma-separated list.

For a list of database server properties, see "Server-level properties" in *Adaptive Server Anywhere Database Administration Guide*.

For example, the following command displays the command line that was used to start the server:

```
dbping –c ... –ps CommandLine
```

Log output messages to file (–o) Write output messages to the named file.

Operate quietly (–q) If dbping fails, a message is always displayed. If succeeds, no message appears if –q is specified.

Display debugging information (–z) This option is available only when an embedded SQL connection is being attempted. That is, it cannot be combined with –m or –l. It displays the network communication protocols used to attempt connection, and other diagnostic messages.

The Server Location utility (dblocate)

The Server Location command-line utility dblocate is provided to assist in diagnosing connection problems by locating databases on the immediate TCP/IP network.

Syntax

dblocate [*options*]

Parameters

The following table lists the options available for the dblocate utility.

Table 3-7: dblocate options

Option	Description
–o *filename*	Log output messages to a file
–q	Operate quietly—do not print messages

Usage

The dblocate utility locates any Adaptive Server Anywhere or Sybase IQ database servers running over TCP/IP on the immediate network. It prints a list of database servers and their addresses.

Depending on your network, the utility may take several seconds before printing its results.

Options

Log output messages to file (–o) Write output messages to the named file.

Operate quietly (–q) Do not display output messages.

The Transaction Log utility (dblog)

Sybase IQ automatically handles the creation and deletion of the transaction log for a database. The command-line Transaction Log utility dblog displays or changes the name of the transaction log or transaction log mirror associated with your database. You can also use dblog to stop a database from maintaining a transaction log mirror, or start maintaining a transaction log mirror.

The database *must* run with a transaction log. The Sybase IQ server will not start without a transaction log. A transaction log mirror is a duplicate copy of a transaction log, maintained by the database in tandem. While a transaction log mirror is not required, it is always recommended, especially for sites that do not back up their IQ database frequently.

Syntax

dblog *[options] database-file*

Parameters

The following table lists the options available for the dblog utility.

Table 3-8: dblog options

Option	Description
-m *mirror-name*	Set transaction log mirror name.
-o *filename*	Log output messages to a file
-q	Quiet mode—do not print messages
-r	No longer use a transaction log mirror
-t *log-name*	Set the transaction log name

Usage

The dblog command line utility allows you to display or change the name of the transaction log or transaction log mirror. You can also stop or start maintaining a transaction log mirror.

The name of the transaction log is first set when the database is created. The database must not be running when you change its transaction log filename.

When you use the RESTORE statement to move and/or rename a database, you can rename all of the files except the transaction log. Transactions continue to be written to the old log file name, in the location where the Catalog Store file (the .db file) is located after the database is restored.

When you rename or move all other files in the database, it is preferable to do the same for the log file. To move or rename the log file, you use the Transaction Log utility (dblog). You should run this utility:

- After using RESTORE with a new database name
- After using RESTORE with the RENAME option

You can use dblog to rename the transaction log even if you have not restored the database, given these restrictions:

- The IQ server must be stopped.
- The databases must not be participating in SQL Remote replication. For IQ, this means that this procedure cannot be used in multiplex environments.
- After the log is renamed, retain the old log until the next database backup, in case the old log is needed for recovery from a media failure.

The dblog utility displays additional information about the transaction log, including the following:

- Version number
- Starting offset, for use in replication
- Ending offset, for use in replication

- Page size
- Total number of pages
- Number of empty pages
- Percentage of the log file in use

For information on truncating the transaction log file, see "The transaction log file" in Chapter 4, "Managing System Resources" of the *Sybase IQ Performance and Tuning Guide*.

Options

Set the name of the transaction log mirror file (-m) This option sets a filename for a new transaction log mirror. If the database is not currently using a transaction log mirror, it starts using one. If the database is already using a transaction log mirror, it changes to using the new file as its transaction log mirror.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only from the command-line utility.

No longer use a transaction log mirror (-r) For databases that maintain a mirrored transaction log, this option changes their behavior to maintain only a single transaction log.

Set the name of the transaction log file (-t) This option sets a filename, including an optional directory path, for a new transaction log. The database changes to using the new file as its transaction log.

Do not use the dblog -n option. This option is *not* supported for IQ databases.

The Validation utility (dbvalid)

The command-line Validation utility dbvalid validates the indexes and keys on some or all of the Adaptive Server Anywhere tables in the Catalog Store. The Validation utility scans the entire table and looks up each record in every index and key defined on the table. By default, the Validation utility uses the express check option.

Note The dbvalid utility lets you easily validate Adaptive Server Anywhere Catalog Store tables, but does *not* validate IQ tables. Use the IQ stored procedure sp_iqcheckdb to validate IQ tables.

You can access the dbvalid utility at the system command-line level, which is useful for incorporating dbvalid into batch or command files.

Syntax

dbvalid [*options*] [*object-name*,...]

Parameters

Table 3-9 lists the options available for the dbvalid utility.

Table 3-9: dbvalid options

Option	Description
object-name	The name of a table or (if -i is used) an index to validate
-c "*keyword=value*; ..."	Supply database connection parameters
-o *filename*	Log output messages to a file
-f	Validate tables with full check
-fd	Validate tables with data check
-fi	Validate tables with index check
-fx	Validate tables with express check
-i	Each *object-name* is an index
-q	Operate quietly—do not print messages
-s	Validate database pages using checksums
-t	Each *object-name* is a table

Usage

With the dbvalid command-line utility, you can validate the indexes and keys on some or all of the Adaptive Server Anywhere tables in the Catalog Store. This utility scans the entire table and confirms that each row exists in the appropriate indexes. It is equivalent to running the Adaptive Server Anywhere VALIDATE TABLE statement on each Catalog Store table.

Note VALIDATE TABLE is not supported in Sybase IQ. The procedure sp_iqcheckdb provides a similar function for IQ Store tables.

By default, the Validation utility uses the express check option. However, the express check option is *not* used if you specify –f, –fd, –fi, –fn, or –i.

If the Catalog Store table is inconsistent, dbvalid reports an error. If errors are reported, you can drop all of the indexes and keys on a table and recreate them. You must also recreate any foreign keys to the table.

Warning! Validating a table or an entire Catalog Store should be performed while no connections are making changes to the database; otherwise, spurious errors may be reported indicating some form of database corruption even though no corruption actually exists.

Exit codes are:

Program Exit Code	Description
0	Database validated successfully
1	General failure in utility
2	Error validating database
7	Unable to find database to connect to (database name is wrong)
8	Unable to connect to database (user ID/password is wrong)
11	Unable to find server to connect to (server name is wrong)
12	Incorrect encryption key for starting database

For information on specific checks made during validation, see the option descriptions that follow.

Options

Connection parameters (-c) Supply database connection parameters. For a description, see "Connection parameters" on page 133 in the *Sybase IQ System Administration Guide*. The user ID must have DBA authority or REMOTE DBA authority.

For example, the following validates the sample database, connecting as user DBA with password SQL:

```
dbvalid -c "uid=DBA;pwd=SQL;dbf-c:\sybase\ASIQ-12_6\demo\asiqdemo.db"
```

Full check for each table (-f) In addition to the default validation checks, carry out both data checks (-fd) and index checks (-fi). This option corresponds to the WITH FULL CHECK option on the Adaptive Server Anywhere VALIDATE TABLE statement. Depending on the contents of your Catalog Store, this option may significantly extend the time required to validate.

Data check for each table (-fd) In addition to the default validation checks, check that all of each LONG BINARY, LONG VARCHAR, TEXT or IMAGE data type can be read. Entries with these data types may span more than one page. In the IQ Catalog Store:

- Domain — user-defined data type
- IMAGE — a domain to LONG BINARY
- TEXT — a domain to LONG VARCHAR

This option instructs the database server to check all pages used by each entry. This corresponds to the WITH DATA CHECK option on the Adaptive Server Anywhere VALIDATE TABLE statement. Depending on the contents of your Catalog Store, this option may significantly extend the time required to validate.

Index check for each table (-fi) In addition to the default validation checks, validate each index on the table. This corresponds to the WITH INDEX CHECK option on the Adaptive Server Anywhere VALIDATE TABLE statement. Depending on the contents of your Catalog Store, this option may significantly extend the time required to validate.

Express check for each table (-fx) This option is only supported for databases created with Sybase IQ version 12.5 or later. In addition to the default and data checks, check that the number of rows in the table matches the number of entries in the index. This corresponds to the WITH EXPRESS CHECK on the Adaptive Server Anywhere VALIDATE TABLE statement. This option does not perform individual index lookups for each row.

Validate specified indexes (-i) Instead of validating tables, validate indexes. Ensure that every row referenced in the index actually exists in the table. For foreign key indexes, it also ensures that the corresponding row exists in the primary table. If you supply a *table-name* instead of an *index-name,* validates the primary key index. In this case, for dbvalid, each of the *object-name* values supplied represents an index instead of a table and has a name in the following format:

```
[ [ owner.]table-name.]index-name
```

Must be the owner of the table on which the index is created, have DBA authority or have REMOTE DBA authority.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages.

Validate database using page checksums (–s) Checksums are used to determine whether a database page has been modified on disk. If you created a database with checksums enabled, you can validate the Catalog Store using checksums. Checksum validation reads each page of the Catalog Store from disk and calculates its checksum. If the calculated checksum is different from the checksum stored on the page, the page has been modified on disk and an error is returned. The page numbers of any invalid Catalog Store pages appear in the server messages window. The –s option cannot be used in conjunction with –i, –t, or any of the –f options.

Validate tables (-t) The list of *object-name* values is a list of tables. This is also the default behavior.

Example

The following command validates the Catalog Store of the sample database, connecting as user DBA with password SQL:

```
dbvalid -c "uid=DBA;pwd=SQL;dbf-c:\sybase\ASIQ-12_5\demo\asiqdemo.db"
```

See also

- "sa_validate system procedure" in Chapter 9, "System Procedures" of the *Sybase IQ Reference Manual*
- "sp_iqcheckdb procedure" in Chapter 9, "System Procedures" of the *Sybase IQ Reference Manual*
- "Database verification" in Chapter 2, "System Recovery and Database Repair" of the *Sybase IQ Troubleshooting and Error Messages Guide*

CHAPTER 4

The SQL Preprocessor

About this chapter

This chapter provides reference information for the SQL preprocessor (sqlpp).

Introduction to the SQL preprocessor

Embedded SQL is a database programming interface for the C and C++ programming languages. Embedded SQL consists of SQL statements intermixed with (embedded in) C or C++ source code. These SQL statements are translated by a SQL preprocessor into C or C++ source code, which you then compile.

The Sybase IQ SQL preprocessor utility sqlpp translates the SQL statements in an input file (*.sqc*) into C language source that is put into an output file (*.c*).

Running the SQL preprocessor (sqlpp)

This section provides the syntax of the sqlpp command and describes the sqlpp command line options.

Syntax

sqlpp [*options*] *sql-filename* [*output-filename*]

Parameters

The following table lists the options available for the sqlpp utility.

Table 4-1: sqlpp options

Option	Description
-d	Favor data size
-e *level*	Flag non-conforming SQL syntax as an error
-f	Put the far keyword on generated static data
-h *line-width*	Limit the maximum line length of output
-k	Include user declaration of SQLCODE
-n	Line numbers
-o *operating-sys*	Target operating system specification (WINDOWS, WINNT or UNIX)
-q	Quiet mode—do not print banner
-r	Generate reentrant code
-s *string-len*	Maximum string constant length for the compiler
-w *level*	Flag non-conforming SQL syntax as a warning
-x	Change multibyte SQL strings to escape sequences
-z *sequence*	Specify collation sequence

Usage

The SQL preprocessor processes a C or C++ program containing Embedded SQL before the compiler is run. sqlpp translates the SQL statements in the input file *sql-filename* into C language source that is put into the *output-filename*. The normal extension for source programs with Embedded SQL is *.sqc*. The default output filename is the *sql-filename* with an extension of *.c*. If the *sql-filename* has a *.c* extension, the default output filename extension is *.CC*.

Options

Favor data size (-d) Generate code that reduces data space size. Data structures are reused and initialized at execution time before use. This increases code size.

Flag SQL/92 errors (-e *level*) This option flags any Embedded SQL that is not part of a specified set of SQL/92 as an error.

The allowed values of *level* and their meanings are as follows:

- **e** flag syntax that is not entry-level SQL/92 syntax
- **i** flag syntax that is not intermediate-level SQL/92 syntax
- **f** flag syntax that is not full-SQL/92 syntax
- **t** flag non-standard host variable types
- **w** allow all supported syntax

Add far keyword (-f) Put the far keyword in front of preprocessor-generated data. This may be required in conjunction with the Borland C++ compiler for the large memory model. By default, all static data is put in the same segment. Adding the far keyword forces static data into different segments. (By default, WATCOM C and Microsoft C place data objects bigger than a threshold size in their own segment.)

Limit maximum output line length (-h num) Limits the maximum length of lines output by SQLPP to *num*. The continuation character is a back slash (\), and the minimum value of *num* is ten.

User SQLCODE declaration (-k) Notifies the preprocessor that the program to be compiled includes a user declaration of SQLCODE.

Generate line number information (-n) Generate line number information in the C file. This consists of *#line* directives in the appropriate places in the generated C code. If the compiler you are using supports the *#line* directive, this option makes the compiler report errors on line numbers in the SQC file (the file with the Embedded SQL) as opposed to reporting errors on line numbers in the C file generated by the SQL preprocessor. Also, the *#line* directives are used indirectly by the source level debugger so that you can debug while viewing the SQC source file.

Target operating system (-o) Specify the target operating system. Note that this option must match the operating system where you run the program. A reference to a special symbol is generated in your program. This symbol is defined in the interface library. If you use the wrong operating system specification or the wrong library, an error is detected by the linker. The supported operating systems are:

- **WINDOWS** Microsoft Windows ME
- **WINNT** Microsoft Windows 2000/2003/XP
- **UNIX** UNIX for your platform

Operate quietly (-q) Operate quietly. Do not print the banner.

Generate reentrant code (-r) For more information on reentrant code, see the discussion of SQLCA management for multi-threaded or reentrant code in the *Adaptive Server Anywhere Programming Guide*.

Set maximum string size (-s *string-len*) Set the maximum size string that the preprocessor puts into the C file. Strings longer than this value are initialized using a list of characters (*'a','b','c'*, etc.). Most C compilers have a limit on the size of string literal they can handle. This option is used to set that upper limit. The default value is 500.

Flag SQL/92 warnings (-w *level*) This option flags any Embedded SQL that is not part of a specified set of SQL/92 as a warning.

The allowed values of *level* and their meanings are as follows:

- **e** flag syntax that is not entry-level SQL/92 syntax
- **i** flag syntax that is not intermediate-level SQL/92 syntax
- **f** flag syntax that is not full-SQL/92 syntax
- **t** flag non-standard host variable types
- **w** allow all supported syntax

Escape (-x) Change multibyte strings to escape sequences so that they can pass through compilers.

Specify collation sequence (-z *sequence*) This option specifies the collation sequence or filename. (For a listing of available collation sequences, see "The Collation utility (dbcollat)" on page 83).

The collation sequence is used to help the preprocessor understand the characters used in the source code of the program, for example, in identifying alphabetic characters suitable for use in identifiers. If -z is not specified, the preprocessor attempts to determine a reasonable collation to use, based on the operating system and SQLLOCALE environment variable.

See also

The chapter "Embedded SQL Programming" of the *Adaptive Server Anywhere Programming Guide*

Chapter 7, "SQL Preprocessor Error Messages" in the *Sybase IQ Troubleshooting and Error Messages Guide*

Index

E

F

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X