SYBASE®

Using Sybase IQ Multiplex

Sybase IQ

15.2

DOCUMENT ID: DC00802-01-1520-01

LAST REVISED: April 2010

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

Contents

About This Book

Audience

This guide is for users who need Sybase® IQ multiplex capability to manage large query loads across multiple nodes. This guide describes concepts and procedures for administering multiplex servers and transactions, including procedures for monitoring and backing up servers. Use the appendix in combination with the other Sybase IQ reference manuals to understand syntax, system objects, and utilities that support multiplex operations.

Related Sybase IQ documents

The Sybase IQ 15.2 documentation set includes:

- *Release Bulletin* for your platform – contains last-minute information that was too late to be included in the books.

  A more recent version of the release bulletin may be available. To check for critical product or document information that was added after the release of the product CD, use the Sybase Product Manuals Web site.

- *Installation and Configuration Guide* for your platform – describes installation, upgrading, and some configuration procedures for Sybase IQ.

- *New Features Summary Sybase IQ 15.2* – summarizes new features and behavior changes for the current version.

- *Advanced Security in Sybase IQ* – covers the use of user-encrypted columns within the Sybase IQ data repository. You need a separate license to install this product option.

- *Error Messages* lists Sybase IQ – error messages referenced by Sybase error code, SQLCode, and SQLState, and SQL preprocessor errors and warnings.

- *IMSL Numerical Library User's Guide: Volume 2 of 2 C Stat Library* – contains a concise description of the IMSL C Stat Library time series C functions. This book is available only to RAP – The Trading Edition™ Enterprise users.

- *Introduction to Sybase IQ* – includes exercises for those unfamiliar with Sybase IQ or with the Sybase Central™ database management tool.
- *Performance and Tuning Guide* – describes query optimization, design, and tuning issues for very large databases.
- *Quick Start* – discusses how to build and query the demo database provided with Sybase IQ for validating the Sybase IQ software installation. Includes information on converting the demo database to multiplex.
- *Reference Manual* – reference guides to Sybase IQ:
  - *Reference: Building Blocks, Tables, and Procedures* – describes SQL, stored procedures, data types, and system tables that Sybase IQ supports.
  - *Reference: Statements and Options* – describes the SQL statements and options that Sybase IQ supports.
- *System Administration Guide* – includes:
  - *System Administration Guide: Volume 1* – describes start-up, connections, database creation, population and indexing, versioning, collations, system backup and recovery, troubleshooting, and database repair.
  - *System Administration Guide: Volume 2* – describes how to write and run procedures and batches, program with OLAP, access remote data, and set up IQ as an Open Server. This book also discusses scheduling and event handling, XML programming, and debugging.
- *Time Series Guide* – describes SQL functions used for time series forecasting and analysis. You need RAP – The Trading Edition™ Enterprise to use this product option.
- *Unstructured Data Analytics in Sybase IQ* – explains how to store and retrieve unstructured data in Sybase IQ databases. You need a separate license to install this product option.
- *User-Defined Functions Guide* – provides information about user-defined functions, their parameters, and possible usage scenarios.
- *Using Sybase IQ Multiplex* – tells how to use multiplex capability, which manages large query loads across multiple nodes.
- *Utility Guide* – provides Sybase IQ utility program reference material, such as available syntax, parameters, and options.

The Sybase IQ 15.2 documentation set is available online at Product Manuals at http://sybooks.sybase.com.

CHAPTER 1

Multiplex Overview

Introduction

Sybase IQ multiplex is a highly scalable shared disk grid technology that allows concurrent data loads and queries via independent data processing nodes connected to a shared data source.

IQ multiplex technology provides the following advantages:

- Performance

  IQ multiplex uses the standard IQ server engine, providing proven query and load performance.

- Low total cost of ownership

  IQ multiplex uses standard, low cost hardware and operating systems.

- Easy, flexible setup and configuration

  IQ multiplex can be configured easily using an SQL interface or through the user-friendly Sybase Central GUI.

- Scalability

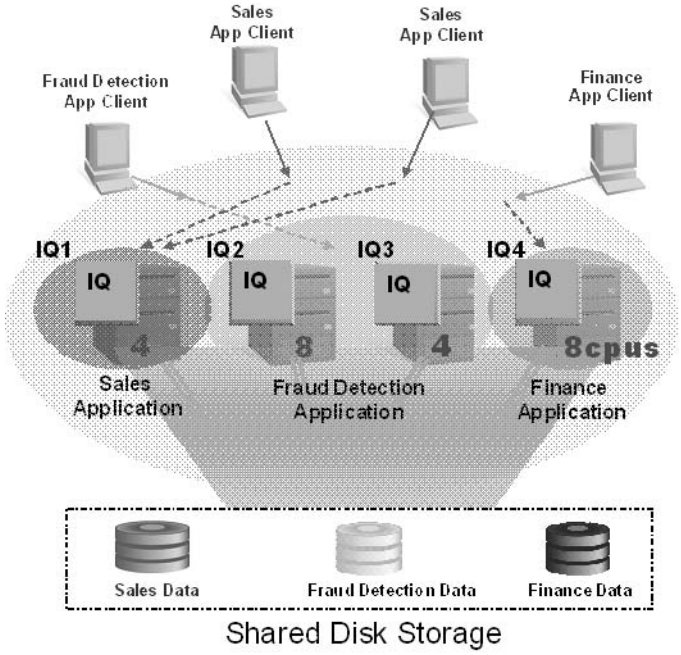
  IQ multiplex can scale to large number of nodes to support tens of thousands of users, with little or no data load and query performance degradation as nodes are added to the multiplex.

- High availability

  Failure of any node leaves query jobs unaffected on other nodes.

IQ multiplex provides an ideal platform for enterprise-wide deployment of critical applications.

Figure 1-1: IQ multiplex for enterprise-wide applications



Architecture

Sybase IQ multiplexes have a hybrid cluster architecture that involves shared storage for permanent IQ data and independent node storage for catalog metadata, temporary data, and transaction logs.

Sybase IQ multiplex:

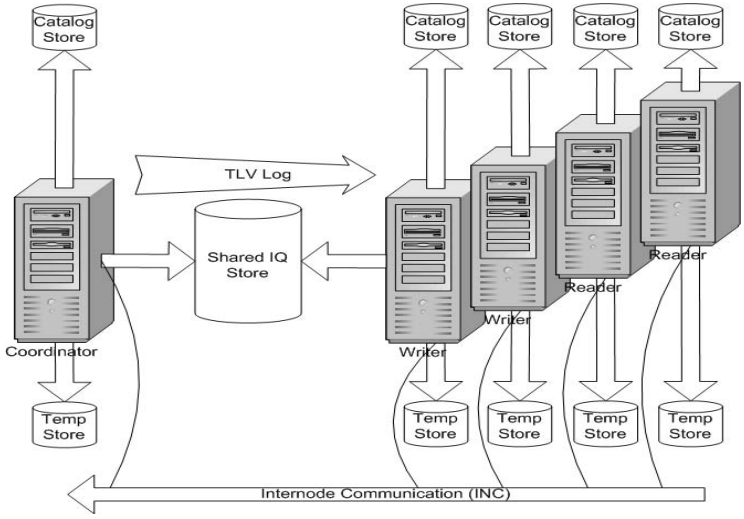
- Lets you manage large workloads across multiple nodes.
- Allows data loads from multiple writer nodes.
- Lets shared database objects be written by one user and queried by multiple users simultaneously. Many objects of this type may exist and be in use at the same time in the database.
- Offers high availability. Failures of a single node leave others up and running.
- Reduces the cost of data by sharing it. Adding systems to the array provides more CPU power and memory space for processing queries if the workload is balanced effectively across the multiplex nodes.

Sybase IQ's column stores reduce I/O requirements, enabling many systems to share the disk array(s) without degrading performance due to I/O contention.

Sybase IQ supports read-write transactions from multiple servers in the multiplex. The primary server, or **coordinator**, manages all global read-write transactions and maintains the global catalog. The **table version log** (TLV log) stores information about DDL operations and communicates information about new table versions to the secondary servers.

The configuration can be described as an “asymmetrical cluster,” because node capabilities may differ, unlike other database cluster architectures, which usually follow either a “shared everything” or “shared nothing” architecture.

Figure 1-2: IQ multiplex architecture



Each server has its own temporary store and catalog store.

The shared IQ store is common to all servers.

Data storage

For details about how Sybase IQ 15.2 stores data, see "Configurable tablespaces" in Chapter 1, "Overview of Sybase IQ System Administration" in the *System Administration Guide: Volume 1*.

Sybase IQ multiplex users need to be aware of the distinction between data managed solely by Sybase IQ and data that is managed by underlying SQL Anywhere software.

Table 1-1: DBspace administration

Managed by IQ	Managed by SA
IQ_SYSTEM_MAIN	System catalog
IQ_SYSTEM_MSG	SA temporary dbspace
IQ_SYSTEM_TEMP	SA catalog dbspaces
IQ user main dbspaces	

IQ_SYSTEM_MAIN and IQ user main dbspaces are shared by all multiplex servers, and all servers need access to the same physical file. See "Accessing the shared disk array" on page 5 for address formats.

Data managed by SQL Anywhere is not shared. Each node needs its own copy of such data.

Each server has its own set of files in IQ_SYSTEM_TEMP and IQ_SYSTEM_MSG.

Note If you have a small test multiplex with servers on the same physical host, follow this rule of thumb: To address dbspaces managed by IQ, use absolute, not relative paths. To address catalog dbspaces managed by SA, use relative paths, not absolute paths.

Accessing the shared disk array

In Sybase IQ 12.7, every node in the multiplex had its own database path, called an **alias**, to use when opening files in the shared IQ main store, temp store and *.iqmsg* file.

Sybase IQ 15.2 no longer supports aliases for the shared IQ main store. DBAs must ensure that all files in all dbspaces in the shared store, namely all files in IQ_SYSTEM_MAIN and shared user dbspaces, are accessible in exactly the same way using the same file paths from all nodes. For IQ_SYSTEM_TEMP and IQ_SYSTEM_MSG, independent files are maintained for each node. The paths for these files must be accessible on the node that owns them.

❖ **Accessing shared disks on UNIX or Linux**

On UNIX or Linux, there are two ways to access shared disks.

1 Use absolute paths to shared stores. For example:

```
/dev/rdsk/c4t600A0B80005A7F5D0000024B49757E55d0s0
/dev/rdsk/c4t600A0B80005A7F5D0000024B49757E55d0s1
/dev/rdsk/c4t600A0B80005A7F5D0000024B49757E55d0s2
```

2 Use soft links to shared stores. For example:

```
store/main

store/userdb1

store/userdb2
```

where each of these soft links point to raw devices as shown in step 1.

❖ **Accessing shared disks on Windows**

On Windows, there are two ways to access shared disks.

1 Use the Disk Physical number. For example:

```
\\\\.\\PhysicalDrive1
\\\\.\\PhysicalDrive2
\\\\.\\PhysicalDrive3
```

2 Use absolute paths using drive letters. For example:

```
x:\main
y:\userdb1
z:\userdb2
```

Using the drive letters limits the number of disks to fewer than 26, so the first method is better.

Multiplex configuration

Sybase IQ multiplex requires some setup data. The configuration data consists of static and dynamic components:

- **Static configuration** includes which nodes are part of the multiplex and the properties of those nodes.
- **Dynamic configuration** includes connectivity status and list of versions held on these nodes.

Coordinator node

Each multiplex configuration requires a coordinator node.

The coordinator node:

- Runs read-only and read-write operations against shared IQ objects.
- Manages IQ main dbspaces.
- Manipulates local data in SQL Anywhere system and user tables.
- Coordinates all read-write operations on shared IQ objects, including:
  - Shared IQ table locking
  - Shared IQ storage management
  - Providing global transaction IDs for read-write transactions involving shared IQ objects
  - Maintaining the global catalog

- Controls catalog synchronization for secondary servers
- Performs schema changes on shared IQ store objects
- Performs schema changes on SQL Anywhere store objects
- Maintains and cleans up object versions

Secondary node

One or more secondary nodes may participate in a Sybase IQ multiplex configuration. One secondary node acts as a designated failover node, the first choice node to assume the coordinator role if the current coordinator is unable to continue.

The number of secondary nodes supported depends on the license purchased, as follows:

- Demo/Trial Edition: Unlimited secondary nodes
- Small Business Edition: None (multiplex not allowed)
- Single Application Server Edition: One secondary node
- Enterprise Edition: Unlimited secondary nodes (license needed for each)

Secondary nodes:

- Can be either read-only nodes (reader nodes) or read-write nodes (writer nodes).

Writer nodes:

- Can run read-only and read-write operations against shared IQ objects.
- Can manipulate local data in temporary and SA base tables.

Reader nodes:

- Can run read-only operations against shared IQ objects.
- Can manipulate local data in temporary and SA base tables.

Inter-node communication (INC)

The servers that participate in the multiplex share messages and data via Inter-node Communication (INC). INC provides a TCPIP-based communication link between multiplex nodes. The link consists of heartbeat and pooled connections.

The topology view tab in Sybase Central shows the relationships between the nodes of a multiplex. INC links exist between the coordinator to the secondary nodes and from secondary nodes to the coordinator, but not between secondary nodes.

Secondary servers communicate with the coordinator via INC for DDL and global transactions. The coordinator communicates with the secondary servers via inter-node communication when certain stored procedures are executed on the coordinator. Secondary servers never communicate with each other.

Heartbeat connections

Every secondary node has a heartbeat connection, a dedicated connection to the coordinator node. This connection is established at the time of secondary node startup and remains active as long as the secondary node is active.

Both the coordinator and secondary node monitor the heartbeat connection. If this connection is broken, the node is declared offline. If the heartbeat is not reestablished within a given time period, the coordinator can automatically exclude the secondary server depending on setting of the option MPX_AUTOEXCLUDE_TIMEOUT.

Pooled connections

Each secondary node maintains an INC connection pool. The connection pool manages connections from the secondary node to the coordinator node used by INC for transactional communication. The INC connection pool reduces communication overhead by re-using existing connections rather than opening new ones, and controls how many INC connections may be established by a secondary node to the coordinator at the same time.

Two database options govern the characteristics of the INC connections from the secondary servers:

- The maximum number of connections to be allowed to be open from a secondary node to the coordinator, MPX_MAX_CONNECTION_POOL_SIZE.
- The maximum number of unused connections to be kept alive when the connection is returned to the unused pool, MPX_MAX_UNUSED_POOL_SIZE.

For more information about these options, see “Database options” on page 79.

Changes in Sybase IQ 15.2 multiplex administration

Sybase IQ 15.2 provides granularity in database administration tasks using new authorities reserved for well-defined tasks. This enables users to need fewer privileges and limits the number of users with DBA authority. MULTIPLEX ADMIN authority allows users without DBA authority to administer multiplex servers.

CHAPTER 2

Managing Multiplex Servers

Multiplex workflow

This topic shows the workflow to implement a multiplex.

❖ **Implementing a multiplex**

1. Using Sybase Central – The graphical user interface simplifies most multiplex operations, although Interactive SQL and command line utilities may be used instead. If you plan to use Sybase Central, start both Sybase Central and the Sybase IQ Agent.
2. Multiplex prerequisites – Make sure that you have the necessary hardware. The main store, including the user dbspaces, is required to be on shared devices that are mounted on all hosts where the multiplex servers are running.

3 Converting simplex databases to multiplex – You can use Sybase Central or scripts to create multiplex servers.

4 Updating dbspaces in multiplex – You can add temporary store files to secondary servers. IQ main store (dbspaces for user data) can only be added on the coordinator. Sybase recommends shutting down all secondary servers when adding more files to IQ_SYSTEM_MAIN.

Using Sybase Central

Sybase Central is a graphical tool to administer and monitor many Sybase server products, including Sybase IQ. You may use either Sybase Central or a combination of command line utilities and Interactive SQL to manage and monitor your multiplex. If you prefer to use Interactive SQL and command line utilities, skip to "Multiplex prerequisites" on page 18.

❖ **Starting Sybase Central on UNIX**

The Sybase IQ Agent must be started before you can use Sybase Central to do many of the multiplex operations. If you have not started the Sybase IQ Agent, see "Running the Sybase IQ Agent" on page 12.

- To start Sybase Central on UNIX, at the command prompt, type:

```
% scjview
```

❖ **Starting Sybase Central on Windows**

- To start Sybase Central on Windows, run Sybase > Sybase IQ 15.2 > Sybase Central Java Edition from the Programs menu.

Running the Sybase IQ Agent

An **agent** is a process (often running on a remote machine) that acts on behalf of a client. The Sybase IQ Agent enables Sybase Central to:

- Start/stop servers
- Access log files
- Perform system functions

These functions are required by all IQ database administrators. If you have a multiplex database, the Sybase IQ Agent must be running *on each machine in your multiplex* in order to fully administer a remote IQ server. Sybase Central can be running while you start and stop agents.

This section gives instructions on running the agent on both UNIX and Windows platforms.

Specifying the host for the Sybase IQ Agent on UNIX

When you use scripts to start the Sybase IQ Agent, use the optional -host parameter to specify the host name explicitly.

To start the agent using the host name returned by uname -n use the optional -host parameter without an argument, as follows:

```
S99SybaseIQAgent15 -host
```

This is the also the default behavior if you omit the -host parameter entirely.

To start the agent using the host's alias enter:

```
S99SybaseIQAgent15 -host <foo>
```

where *foo* is an alias present in the */etc/hosts* file.

You can supply a list of alternate host names or IP addresses, separated by commas, instead of a single argument. Blank spaces are disallowed in the list. IP aliases may be used on a single host that supports multiple network cards with unique addresses. For example:

```
S99SybaseIQAgent15 -host
"host1.sybase.com,10.20.30.40,
host2,12.20.30.41"
```

To avoid problems, start the Sybase IQ Agent using either the short host name only or all three structures. For example:

```
S99SybaseIQAgent15 -host host2 -port 2099
```

or

```
S99SybaseIQAgent15 -host "ciaran,ciaran.sybase.com,
10.50.7.70"  -port 2199
```

The agent binds in the following order:

1 The local host name

2 The host IP address

3 The item(s) specified in the -host command line parameter

Configuring the Sybase IQ Agent to Start Automatically

Configuring the IQ Agent on UNIX

To enable automatic startup for the Sybase IQ Agent, place the following file in the UNIX startup directory (usually */etc/rc**):

```
$IQDIR15/bin64/S99SybaseIQAgent15
```

On a 32-bit system use “bin32” instead of “bin64”.

After you install Sybase IQ and move the file, the agent starts automatically whenever you reboot your system.

Configuring the IQ Agent on Windows

On Windows systems, the Sybase IQ 15.2 installation installs the Windows service Sybase IQ Agent 15, but you must reboot the machine to make the agent service function correctly.

Disabling remote file browsing on the Sybase IQ Agent

The Sybase IQ Agent supports remote file browsing by default. If you have security concerns, disable browsing

Disabling browsing on UNIX

There are two ways to disable browsing on UNIX:

- Start the agent with the parameter -nobrowse. For example, on UNIX:

  ```
  $IQDIR15/bin64/S99SybaseIQAgent15 -nobrowse
  ```

- Set the environment variable IQAGENTNOBROWSE to any value. For example, in csh:

  ```
  setenv IQAGENTNOBROWSE 1
  ```

Disabling browsing on Windows

In a DOS shell, set the environment variable IQAGENTNOBROWSE to any value. For example:

```
SET IQAGENTNOBROWSE=1
SET IQPORT=2525
```

Start the agent in the window where you started the DOS shell. For example:

```
SybaseIQAgent15.exe -r "SybaseIQAgent15"
```

Uninstalling and reinstalling Sybase IQ Agent Windows Service

To uninstall the Sybase IQ Agent 15 Windows Service:

```
SybaseIQagent15.exe -u "SybaseIQAgent15"
```

To reinstall the Sybase IQ Agent 15 Windows Service:

```
SybaseIQagent15.exe -i "SybaseIQAgent15"
```

Overriding the Sybase IQ Agent Port Number

Overriding the IQ Agent Port Number on UNIX

The Sybase IQ Agent port number defaults to 1099. You can override the default value, provided that you do so before the plug-in starts. Changing the default port number lets you to run any number of version 15.2 Sybase IQ Agents on a given host, or run agents for Sybase IQ 12.7, 15.0, 15.1 and 15.2 on the same host. Use the IQPORT environment variable or the -port command line parameter to override the default agent port number. You can also specify the port per server within Sybase Central, or override the value on the Sybase Central startup command, for example:

```
scjview -DIQPORT=3356
```

On UNIX, you can override the agent port number on the startup command line; for example:

```
$IQDIR15/bin64/S99SybaseIQAgent15 -port nnnn
```

Overriding the IQ Agent Port Number on Windows

On Windows, use the Service Manager, as follows:

1. Select the name Sybase IQ Agent 15.
2. Click the Properties icon or choose Action > Properties from the menu bar.
3. Click Stop under Service status to stop the agent.
4. In the Start Parameters text box, type `-port nnnn` where nnnn is the port number.
5. Click Start to restart the agent.

If the agent fails to start on Windows, check the event log for diagnostic information.

Setting permissions for the Sybase IQ Agent

Setting the permissions on UNIX

The S99SybaseIQAgent15 script should be owned by the same UID that is used to run all the servers. Do not use the root user account.

Setting the permissions on Windows

On Windows, you must change the owner of the Sybase IQ Agent. The user who starts the agent will be the creator and owner of multiplex databases and server directories, and must have write privileges on all of the servers in the multiplex. By default, the System account starts the agent.

❖ **Changing the user**

1. Choose Administrative Tools > Services.
2. Right-click “Sybase IQ Agent” and choose Start.
3. On the General tab, choose Automatic for Startup Type.
4. On the Log On tab, change the Log on as: option from the System Account to This Account.
5. Type *domain\username* for an account with the appropriate privileges in the account text box.

   or

   Browse to select an account, then click OK.
6. Type and reconfirm that account’s password, then click OK.

Troubleshooting Sybase IQ Agent startup

Agent startup on UNIX

To check if the agent is running, run the stop_iq utility:

```
stop_iq -agent
Checking system ...

The following 1 agent(s) are owned by 'ciaran'
## Owner     PID     Started  CPU Time Additional Information
-- --------- ------- -------- -------- -----------------------
- 1: ciaran 6669 Sep.01 5:11 PORT:1100 java
-Diq.agent=/work/sybase15/
IQ-15_2/java/IQAgent15.jar
-Di q.agent_log=/

-- Do you want to stop the agent displayed above <Y/N>?
```

```
Y
```

This output shows that user *ciaran* is running the agent.

Should you ever receive an error that the agent is not running, change directory to *$IQDIR15/bin64* (or *$IQDIR15/bin32*, depending on your platform) and type `S99SybaseIQAgent15` to restart the agent.

Agent startup on Windows

If you receive an error message that the Sybase IQ Agent is not running, open the Services utility under Control Panel. If Sybase IQ Agent does not have the status “Started,” click Start. Restart Sybase Central after starting the agent.

IQ Agent Log

If you experience any problems with Sybase Central, check the agent log file.

❖ **Viewing the Sybase IQ Agent log in Sybase Central**

- Right-click the server name and choose Agent > View Agent Log

❖ **Locating the Sybase IQ Agent log file**

- On UNIX systems, if the default Sybase IQ Agent port (1099) is used, the log name is: *$IQDIR15/logfiles/SybaseIQAgent1099.nnn.log* where *nnn* is the number of times you have started the agent since the directory was last cleaned out. For example, if you have started the agent three times since the last cleanup: *$IQDIR15/logfiles/SybaseIQAgent1099.003.log*

  On Windows systems, if the user explicitly set the Sybase IQ Agent port, the log name defaults to: *%ALLUSERSPROFILE%\SybaseIQ\logfiles\SybaseIQAgent_pppp.nnn.log* where *pppp* is the port number and *nnn* is the number of times you have started the agent since the directory was last cleaned out. For example: *C:\Documents and Settings\All Users\SybaseIQ\logfiles\SybaseIQAgent_7899.001.log*.

❖ **Changing the Sybase IQ Agent log file location**

To generate the agent log files in a different directory, set the log directory environment variable.

- On UNIX systems, set the $IQLOGDIR15 environment variable to the full path of the desired directory. For example, for the C shell (csh):

  ```
  setenv IQLOGDIR15 /work/mydir/agentlogs
  ```

  On Windows systems:

  1. Use Control Panel > System > Advanced Properties > Environment Variables.
  2. Click New under System variables.
  3. Type IQLOGDIR15 for **Variable** and the full path of the directory for **Value**.
  4. Click OK.

Multiplex prerequisites

In a multiplex environment, all main store file paths must be absolute or relative such that they are accessible to all servers in the multiplex. (The exception is Sybase IQ 12.7, when all servers in a multiplex had independent paths to the IQ main store files.)

A multiplex requires a database that has a shared disk array for the main store.

- See Chapter 5, "Working with Database Objects," in *Sybase IQ System Administration Guide* to learn how to create IQ databases.
- See Chapter 6, "Migrating Data," in *Installation and Configuration Guide* if you are migrating your database from an earlier release of Sybase IQ.

In order to run multiplex operations with Sybase Central, the Sybase IQ GUI, the Sybase IQ Agent must be running. The agent starts automatically on Windows systems, but not on UNIX. For details, see "Running the Sybase IQ Agent" on page 12.

Hardware

Sybase recommends that you locate the shared main IQ store on a Shared Disk Array. See "Accessing the shared disk array" on page 5.

All machines running servers participating in the multiplex must have Sybase IQ 15.2 installed. Sybase IQ does not support multiplexes of Sybase IQ servers at different release levels. For upgrade and migration instructions, see the *Installation and Configuration Guide*.

Note Sybase IQ does not support heterogeneous multiplexes (UNIX and Windows servers in a mixed multiplex). Coordinator and secondary servers must be on the same operating system and hardware platform.

For secondary servers, you must:

- Set up a computer with access to the shared IQ store using paths identical to those used by the existing server.
- Provide a directory on a local disk to contain the files for this server.

Multiplex planning worksheet

Sybase Central dialogs and SQL statements for creating a multiplex require some or all of the values in Table 2-1. Database paths on all platforms, whether raw device or operating system files, are limited to128 bytes.

Table 2-1: Multiplex database requirements

Dialog item	Type/length	Notes	Value
Host name	CHAR 128	Name of the machine where the database engine will run.	
Server name	CHAR 128	Server name for the coordinator. (The server name must be unique across the local area network.)	
Database path	CHAR 128	Create the database files on a local disk, not a remote location. The Create Database wizard asks for the path to the dbfile. Users cannot specify where the server will be started.	
IQ store paths (temp and main)	CHAR 128	All files in IQ_SYSTEM_MAIN and shared user dbspaces must be accessible in exactly the same way using the same file paths from all nodes. IQ main dbspace paths are shared and temporary and *.iqmsg* dbspace paths are only valid on the owning node.	
Database name	CHAR 70	Database name, limited to 70 characters. Included in the path.	

Converting simplex databases to multiplex

To change a simplex database to a multiplex, you simply connect to the simplex database and add a multiplex server. The simplex database server becomes the coordinator and the servers you add are called the secondary servers. You may use Sybase Central or Interactive SQL statements to create multiplex servers.

❖ **Converting databases to multiplex (Sybase Central)**

If you are new to running Sybase Central, see "Using Sybase Central" on page 12.

DBA authority is required to perform the following steps.

1. Make sure that your system meets hardware prerequisites. The main store dbfiles must be visible to secondary nodes. See "Multiplex prerequisites" on page 18. and "Accessing the shared disk array" on page 5.
2. Start Sybase Central using the method appropriate for your platform.
3. Use the Start Server Wizard to start a Sybase IQ server.
4. Connect to the IQ server.
5. Right-click the server name and choose Convert to Multiplex (Alt+M). If you do not have DBA authority, the Convert to Multiplex menu item is disabled.
6. Follow the instructions on the Create Server Wizard screens.

❖ **Converting databases to multiplex (Interactive SQL)**

In a multiplex environment, all IQ main store file paths must be absolute or relative in a way accessible to all servers in the multiplex.

1. Make sure that your system meets hardware prerequisites. The main store dbfiles must be visible to secondary nodes. See "Multiplex prerequisites" on page 18. and "Accessing the shared disk array" on page 5.
2. Start a Sybase IQ server and connect to a database with a shareable main store. This server becomes the coordinator of the multiplex.

   For example:

   ```
   start_iq @params.cfg -n mpxnode_c -x
   "tcpip{host=host1;port=2763}" mpxtest.db
   ```

3. Connect using Interactive SQL, for example:

   ```
   dbisql
   ```

4. In Interactive SQL, run the CREATE MULTIPLEX SERVER command, for example:

   ```
   CREATE MULTIPLEX SERVER mpxnode_w1 DATABASE
   'host2/mpx/mpxtest.db' HOST 'host2' PORT 2957 ROLE
   WRITER STATUS INCLUDED
   ```

   For command reference details, see "CREATE MULTIPLEX SERVER statement" on page 77.

   This command makes the connected server mpxnode_c into the coordinator and creates the definition for a secondary writer server mpxnode_w1.

5 The server shuts down to re-initialize the server as a multiplex coordinator. (As a normal part of this shutdown, when the first secondary node is created, Interactive SQL may return a disconnected (-308) error that requires no corrective action.)

6 Restart the coordinator. For example:

```
start_iq @params.cfg -n mpxnode_c -x
"tcpip{host=host1;port=2763}" mpxtest.db
```

7 Back up the catalog store to the directory where the secondary server will run. From the secondary node machine, run this backup command from the computer where the secondary node will run, because the *.DB* file should be on a local disk. For example, run the following command from the directory where the secondary node's *.DB* file resides:

```
dbbackup -y -d -c
"uid=DBA;pwd=sql;links=tcpip{port=2763};eng=mpxnode
_c" .
```

If you have changed to the directory where the secondary node's *.DB* file will reside, you can specify a dot (.) instead of the path on the dbbackup command.

8 Make sure that you remove the transaction log files in the directory where the secondary server will run:

```
rm -rf mpxtest.log
```

9 Run dblog to reset the transaction log:

```
dblog -r -t mpxtest.log mpxtest.db
```

10 Start the secondary server.

```
start_iq -STARTDIR /host2/mpx @params.cfg -n
mpxnode_w1 -x "tcpip{port=2957}" mpxtest.db
```

11 Start Interactive SQL and connect to the secondary multiplex node.

```
dbisql -c
"uid=DBA;pwd=sql;eng=mpxnode_w1;links=tcpip{port=29
57}"
```

12 Add the temporary dbfile in Interactive SQL using the ALTER DBSPACE statement. (Secondary servers do not allow CREATE DBSPACE.) For example:

```
ALTER DBSPACE IQ_SYSTEM_TEMP ADD FILE
mpxnode_w1_temp 'w1_temp1.iqtmp' SIZE 500
```

Starting the server creates a message log for you, but you must create the temporary store file yourself. You must connect to the secondary server to add a temp store file for that server. Adding temporary store files in multiplex always adds a file specifically to the server to which you are connected.You can start the server, but cannot run IQ queries until a temporary store is added.

For command reference details, see the ALTER DBSPACE statement in Chapter 1, "SQL Statements," in *Reference: Statements and Options*.

Multiplex database files

Each server in the multiplex uses several files.

Note Some of these files are only created automatically by Sybase Central.

Table 2-2: Contents of multiplex database directories

Folder, Directory, or Filename	Purpose
dbname.db	File that contains the catalog store. This cannot be a raw device.
dbname.iqmsg	File that contains messages from Sybase IQ
dbname.iqtmp	This is the IQ temporary store. This file exists only if the temporary file is added as an OS file. Raw devices are recommended for all files in the IQ temporary store.
dbname.log	File that contains the database transaction log. This cannot be a raw device.
dbname.lmp	License management property file. Creating a database generates this file automatically. Dropping a database deletes all database files except this one.
params.cfg	Contains startup parameters for this particular server. May be customized for performance or specific requirements. This file is only created automatically if you use Sybase Central. (If created by Sybase Central, *params.cfg* does not contain server name or tcpip parameters.)
start_server	Sybase Central generates these scripts. If you use Sybase Central to create the multiplex, they are generated automatically. You can also generate them using Sybase Central after you create the multiplex with Interactive SQL.
stop_server	
sync_server	

For more information about the administrative scripts, see "Using administrative shell scripts" on page 25.

Checking server status

After you add the first secondary server, the database is enabled for multiplex. You can check the status of all the multiplex nodes now, if desired.

❖ **Checking server status (Sybase Central)**

1 Open the multiplex folder.

2 Select the Servers tab to view server status. The Servers tab lists details shown in Table 2-3.

Table 2-3: Servers tab content

Column	Description
Name	Server name
Role	Coordinator, Reader, or Writer.
IsConnected	Yes if server is connected. Otherwise No. Indicates whether Sybase Central has a connection to the server. Sybase Central only needs a connection to one of the servers in the multiplex, but the user may connect Sybase Central to more than one.
Failover node	Yes if this is the designated failover node, otherwise No.
Sync State	Synchronized, Unsynchronized (not a valid state), Excluded, Coordinator, or Not Running
As Of	When the displayed information was last updated. For example: “8/5/08 7:01:18 PM”

3 Status displays are not instantaneous and depend on network latency. To refresh the status display, select View > Refresh Folder from the main menu bar.

Note In a multiplex environment, users with RESOURCE authority see coordinators in the Servers folder, rather than the Multiplex folder.

❖ **Checking server status (Interactive SQL)**

1. Connect to the coordinator as a user with DBA authority, or as a user who has been assigned execute permission on the sp_iqmpxinfo procedure.
2. Run sp_iqmpxinfo.

   For more information, see "sp_iqmpxinfo procedure" on page 91.

Determining the number of user connections

INC connections from secondary nodes also show up as connections in the DB_PROPERTY function output.

For example, from the coordinator, run the system function DB_PROPERTY as follows:

```
SELECT db_property('conncount')
```

This function returns the number of client connections and the number of INC connections made by the secondary nodes to the coordinator. The INC connections do not affect the number of connections (set by startup parameter -gm) that can be made to the coordinator node.

Using administration tools

Sybase Central and Interactive SQL are the main tools for multiplex administration. For SQL syntax, see the Appendix, "Multiplex Reference." For Sybase Central, see Managing servers (Sybase Central). You can also generate administrative scripts with Sybase Central, as described in "Using administrative shell scripts" on page 25.

❖ **Managing servers (Sybase Central)**

1. Right-click the desired server.
2. Select the desired operation from the menu.

   You can perform the following operations from the popup menus:

   - View IQ Agent properties (version, port number, timeout setting) or Agent log.
   - Disconnect or drop the selected server.

- Open Interactive SQL

  This option connects without requiring the user to enter login information.

- Rename, stop, or synchronize the selected server. If the server is a coordinator, ping the host, rename, or stop the server.
- Generate Administrative Scripts

  Use this option to create or update the optional scripts that start or stop the server. You should generate the scripts after you install a new release or update of Sybase IQ. See "Using administrative shell scripts" on page 25.

- View the selected server's properties
- View the console log.

Note Wizards on the coordinator (Start, Stop, and Synchronize) allow you to affect all servers in the multiplex.

Using administrative shell scripts

Administrative scripts may be used to automate operations that you might normally perform periodically with Sybase Central. The scripts can:

- Start, stop, or synchronize servers
- Start dbisql

For example, you could use the scripts to synchronize secondary servers during the night.

The scripts *start_server* and *sync_server* are located in the database directory. On a Windows host, these scripts are *.bat* files. On UNIX, they are shell scripts. You can run these scripts from the command line in place of start_iq and dbbackup, respectively.

Open each script with a text editor to see comments about how to use it.

Scripts that require a password define it as an input parameter.

❖ **Creating administrative scripts**

1 Right click the server

2 Choose Generate Admin Scripts from the popup menu.

❖ **Scheduling administrative scripts (UNIX)**

- You can use the crontab system utility to schedule administrative tasks.

❖ **Scheduling administrative scripts (Windows)**

- Use Start > Settings > Control Panel > Scheduled Tasks to schedule the administrative tasks.

Using authorities for administration

Sybase IQ allows users with certain authorities to perform well-defined sets of database administration tasks. For an overview of authorities that affect both simplex and multiplex servers, see "Database permissions and authorities overview" in "Managing User IDs and Permissions," in *System Administration Guide: Volume 1*.

MULTIPLEX ADMIN authority overview

MULTIPLEX ADMIN authority allows a user without DBA authority to perform multiplex administration tasks.

MULTIPLEX ADMIN or DBA authority is required to:

- Rename the multiplex and store the multiplex name in SYS.ISYSIQINFO system table.
- Change multiplex server settings.
- Create multiplex servers.
- Delete servers from the multiplex.

MULTIPLEX ADMIN authority can be granted only by the DBA or PERMS ADMIN to other users.

Using multiplex procedures for tailored security

For strict security, you can disallow all access to underlying tables, and grant permissions to users or groups of users to execute certain stored procedures. This approach strictly defines who can define data in the database.

To allow users with the particular authorities to administer certain tasks using IQ system procedures:

1 Create a group for each desired authority.

2 Grant the authority to the designated group.

3 Grant EXECUTE permissions on the IQ procedure for performing the authority tasks to the group.

When you create a new user who is to be granted the authority, grant membership for this user to the group created for that authority. Since most authorities are inherited through group membership, the user inherits the authority and also the execute permissions for the IQ procedures from the group.

❖ **Granting users the permissions to run related stored procedures**

Except as noted, the following steps require DBA or PERMS ADMIN authority.

To grant MULTIPLEX ADMIN authority and permissions to execute procedures related to multiplex administration to a user named user1:

1 Connect to the database as a user with DBA authority or a user with USER ADMIN and PERMS ADMIN authority.

2 Create a group MPX ADMIN_GRP. You could write this as:

```
CREATE USER MPXADMIN_GRP
GRANT GROUP TO MPXADMIN_GRP
```

or:

```
call sp_addgroup('MPXADMIN_GRP')
```

3 Grant MULTIPLEX ADMIN authority to MPXADMIN_GRP:

```
GRANT MULTIPLEX ADMIN TO MPXADMIN_GRP
```

4 Grant EXECUTE permission on Sybase IQ stored procedures for user administration to MPXADMIN_GRP:

```
GRANT EXECUTE on sp_iqmpxgetconnversion
to MPXADMIN_GRP
GRANT EXECUTE on sp_iqmpxinfo
```

```
to MPXADMIN_GRP
GRANT EXECUTE on sp_iqmpxincconnpoolinfo
to MPXADMIN_GRP
GRANT EXECUTE on sp_iqmpxincheartbeatinfo
to MPXADMIN_GRP
GRANT EXECUTE on sp_iqmpxvalidate
to MPXADMIN_GRP
GRANT EXECUTE on sp_iqmpxversioninfo
to MPXADMIN_GRP
```

5 Grant membership in group MPXADMIN_GRP to user1. user1 inherits the MULTIPLEX ADMIN authority and the ability to execute the assigned IQ procedures through membership in MPXADMIN_GRP group.

```
GRANT MEMBERSHIP IN GROUP MPXADMIN_GRP TO user1
```

Running multiplex servers

To start and stop IQ multiplex servers interactively, use Sybase Central or the start server utility, start_iq. To start IQ multiplex servers in scripts, you may use command line parameters.

❖ **Editing the params.cfg file**

The *params.cfg* file holds values for switches to control server startup features such as main and temp cache settings and error logging. Before you start the server, review and edit the *params.cfg* file located in the database directory for the server.

Table 1-1 in the *Utility Guide* describes startup switches. For more about *params.cfg*, see Table 2-1 in the *System Administration Guide*.

If you use Sybase Central to start the database:

1 Remove any -n switch in a *params.cfg* file used to start a multiplex database. Remove -x tcpip{port = } switch because Sybase Central prompts for the port number.

The configuration file must not contain -n to name the server, or the database name or path.

2 The configuration file must be named *params.cfg*.

3 The configuration file must be located in the same folder or directory as the database file (*.db*).

❖ **Starting servers (Sybase Central)**

Use Sybase Central to start multiplex servers interactively.

1 Log in using an account with DBA privileges and start Sybase Central as appropriate for your platform.

2 In the left pane of Sybase Central, select Sybase IQ 15.

3 Select Tools > Sybase IQ 15 > Start Server.

By default, the wizard starts a single server. To start all servers, select the "All servers in multiplex" option button.

4 Follow the instructions in the wizard.

When starting all of the servers in a multiplex, supply information for the coordinator. Always change the default port number for each server to a different number that is not in use.

❖ **Starting servers (Command line)**

The start_iq utility starts simplex or multiplex servers. The -n <engine name> switch is required. The engine name must match the server name used when creating the multiplex server. The -x (connection string) value must match tcpip connection parameters specified when creating the multiplex server. The database file path must match the database file path as specified when creating the multiplex server. For applications, the -gn value must be set higher than than total number of expected simultaneous requests plus 5 (for internal events and connections.) For a complete list of parameters, see the *Utility Guide*.

- The following command starts a server and names it host1_test1_writer:

```
start_iq @/host1/mpxdevices/params.cfg -n
host1_test1_writer -x "tcpip{host=host1;port=2763}"
/host1/mpxdevices/test1.db
```

Note If you checked "Generate Admin Scripts" when creating the multiplex server in Sybase Central, *start_server* and *sync_server* scripts are created. You can run these scripts from the command line in place of start_iq and dbbackup, respectively.

Synchronizing secondary servers

The process that updates a secondary server with respect to the coordinator is called **synchronization**.

Synchronizing copies the coordinator's version of the database catalog to a single secondary server. The multiplex automatically makes committed changes, including global schema changes, changes to data in IQ tables visible on all servers in the multiplex. In general, Sybase IQ propagates DDL and DML changes to secondary servers, so you only need to synchronize a secondary server for these server management actions:

- While creating a new secondary server
- Restoring the coordinator from backup
- Restarting a secondary server that has been excluded or shut down for an extended time period
- After running the coordinator in single-node mode
- After adding a file to the IQ_SYSTEM_MAIN dbspace

❖ **Synchronizing individual secondary servers (Sybase Central)**

1. Make sure that the coordinator is running, and connect to it with Sybase Central.
2. To synchronize a secondary server, right-click that server and choose Control > Synchronize. You can keep running queries on the other servers.

   Sybase Central stops the secondary server that you right-clicked, replaces its catalog store, then restarts the server. The coordinator continues running throughout the operation.

❖ **Synchronizing multiple servers (Sybase Central)**

To synchronize multiple servers:

1. Right-click the Multiplex folder to launch the synchronization wizard.
2. Select Control > Synchronize context menu item.
3. In the wizard page, select the servers to synchronize.

❖ **Synchronizing servers (command line)**

Perform the following steps on the computer where the secondary server runs.

Note If you checked "Generate Admin Scripts" when creating the multiplex server in Sybase Central, you can run the *sync_server* script instead of dbbackup.

1. Remove the *.LOG* transaction log file in the secondary server directory.
2. Back up the catalog store to the directory where the secondary server runs:

```
dbbackup -y -d -c
"uid=dba;pwd=sql;links=tcpip{port=2763};eng=mpxnode
_c" /host1/mpx/
```

3. Set the log file path:

```
dblog -r -t mpxtest.log mpxtest.db
```

4. Start the secondary node using the start_iq command.
5. Repeat the previous steps for each secondary server in the multiplex.

Adding multiplex servers

The coordinator must be running in order to add secondary servers. See "Checking server status" on page 23.

❖ **Adding multiplex servers (Sybase Central)**

1. Right-click the server name in the Multiplexes folder.
2. Choose Add Server from the popup.
3. Follow the instructions in the Create Server Wizard.

❖ **Adding multiplex servers (Interactive SQL)**

1. In Interactive SQL, run the CREATE MULTIPLEX SERVER command, for example:

```
CREATE MULTIPLEX SERVER mpxnode_w2 DATABASE
'host1/mpx/mpxtest.db' HOST 'host1' PORT 2957 ROLE
WRITER STATUS INCLUDED
```

For command reference details, see "CREATE MULTIPLEX SERVER statement" on page 77.

This command creates the definition for a secondary writer server mpxnode_w2.

2 Follow steps 7 through 12 in "Converting databases to multiplex (Interactive SQL)" on page 20.

The first time you start the multiplex after adding a secondary server, the coordinator may return an error similar to:

```
I. 12/11 12:50:08. Trying to start TCPIP link ...
I. 12/11 12:50:08. Multiplex environment incorrect for
this server
I. 12/11 12:50:08. Please connect and run procedure
sp_iqmpxvalidate for help
```

This error occurs because the secondary server has no temporary file in IQ_SYSTEM_TEMP. You must add a temporary dbfile to IQ_SYSTEM_TEMP on the newly added server. See "Converting databases to multiplex (Interactive SQL)" on page 20, steps 11 and 12. After adding this file, run sp_iqmpxvalidate to make sure that no errors are reported on this server.

Altering multiplex servers

You can perform various operations to alter servers from the server property sheet in Sybase Central or using Interactive SQL. The coordinator must be running. Although you can run the ALTER MULTIPLEX SERVER command from any server in the multiplex, it is recommended that, like all DDL, commands be run on the coordinator.In all cases except when altering role from reader to writer, the named server is automatically shut down once the change is committed.

❖ **Renaming servers (Sybase Central)**

1 Open the server property sheet.

2 Change the server name on the Configuration tab.

❖ **Renaming servers (Interactive SQL)**

1 Start the server, connect to it, and issue a command in the following format:

```
ALTER MULTIPLEX SERVER oldname RENAME newname
```

2 This command renames and shuts down the named server. For complete syntax, see "ALTER MULTIPLEX SERVER statement" on page 76.

❖ **Changing database file path (Sybase Central)**

1 Open the server's property sheet.

2 On the Configuration tab, change the database file path.

❖ **Changing database file path (Interactive SQL)**

1 Start the server, connect to it, and issue a command in the following format:

```
ALTER MULTIPLEX SERVER server name DATABASE 'new db
file path'
```

2 This command also shuts down the named server.

❖ **Changing server role (Sybase Central)**

1 Open the server property sheet.

2 On the General tab, change the server role (reader/writer).

❖ **Changing server role (Interactive SQL)**

1 Start the server, connect to it, and issue a command in the following format:

```
ALTER MULTIPLEX SERVER server name ROLE
{WRITER|READER}
```

You cannot change the role of coordinator. Changing a write server to a reader automatically shuts down the server.

❖ **Changing host/port (Sybase Central)**

1 Open the server's property sheet.

2 On the Configuration tab, change the host/port information.

❖ **Changing host/port (Interactive SQL)**

1 Start the server, connect to it, and issue a command in the following format:

```
ALTER MULTIPLEX SERVER server name HOST 'hostname'
PORT portnumber
```

This command also shuts down the named server.

❖ **Including or excluding servers (Sybase Central)**

If a multiplex secondary server will be shut down for an extended period of time, that server should be excluded. Excluding the server allows the coordinator to ignore this server when performing version cleanup. Otherwise, the coordinator will need to reserve all old versions of IQ objects since the secondary node was shut down. This takes up unnecessary disk space. A designated failover server cannot be excluded unless it is the last secondary node to be excluded in the multiplex. Include/exclude does not apply to the coordinator.

1 Open the server property sheet.

2 On the General tab, include or exclude the server.

An alternate method is to right-click the server and select context menu Control > Include/Exclude.

❖ **Including or excluding servers (Interactive SQL)**

1 Start the server, connect to it, and issue a command in the following format:

```
ALTER MULTIPLEX SERVER server name STATUS
{INCLUDED|EXCLUDED}
```

2 If the target server is running, it is strongly recommended that you shut it down before excluding it. In case you do not, it will eventually shut itself down, but it is better that you plan for and shut it down prior to excluding it. Excluding a server shuts it down. After including a server, the server must be synchronized and then started.

❖ **Designating failover node (Sybase Central)**

1 Right-click the Multiplex Name in the Sybase Central tree view.

2 Choose Designate Failover.

❖ **Designating failover node (Interactive SQL)**

1 Start the server, connect to it, and issue a command in the following format:

```
ALTER MULTIPLEX SERVER server name ASSIGN AS
FAILOVER SERVER
```

The designated failover node defaults to the first multiplex server added to the multiplex.

See also "Failover" on page 36.

Dropping multiplex servers

Dropping a secondary server removes it from the multiplex configuration. If the target server is running, it is strongly recommended that you shut it down before dropping it. In case you do not, it will eventually shut itself down, but it is better that you plan for and shut it down prior to dropping it. You cannot drop the coordinator node and the designated failover node unless it is the last secondary node. When the last secondary server is removed, the multiplex is converted back to simplex and the coordinator shuts down. A write server that is holding free list cannot be dropped.

A normal restart of the write server will give up its free list and then you can shut it down and then drop it. If, for some reason, the writer cannot start, and you can guarantee that its process is dead, you can restart the coordinator with the -iqmpx_reclaimwriterfreelist switch. This forces the coordinator to reclaim the writer's free list and it can then be dropped. This startup flag should be used only when you can guarantee that the writer process is dead and cannot be started. If the writer process is still writing to the database when the coordinator reclaims its free list, database corruption may result.

❖ **Dropping servers (Sybase Central)**

1. To delete a secondary server, right-click that server and choose Drop Server from the popup menu.
2. The wizard prompts you before dropping the server.
3. Click the option button to delete associated files, if desired, and choose Finish.

If, for some reason, you did not shut down the server you are dropping, Sybase Central may fail to delete all files for that secondary server because they are in use.

❖ **Dropping servers (Interactive SQL)**

1. Connect to the coordinator.
2. Issue a command in the following format:

```
DROP MULTIPLEX SERVER server name
```

Replacing the coordinator

If the current coordinator node fails, or must be shut down for maintenance, the entire multiplex is placed in a read-only state. In this state, you can query the existing IQ data but not modify it.

To re-establish read-write capabilities, you must promote another server to be the coordinator. This operation is called **manual failover**.

Designated failover node

A multiplex requires a designated failover node to take over as coordinator if the current coordinator is not running. This node must be used when performing manual failover. During multiplex creation, the first secondary server created becomes the designated failover node, but you can later designate any other secondary server as failover node.

Use the sp_iqmpxinfo procedure to display the designated failover node. To change the failover node, see Designating failover node (Sybase Central) and "Designating failover node (Interactive SQL)" on page 34.

Failover

It is the user's responsibility to ensure that the former coordinator process is no longer running before attempting failover. In a worst case scenario, the former coordinator computer might be running but disconnected from the network, or in a hardware hibernation state. In this situation, you cannot log into the coordinator computer, which would be unreachable by tools such as Sybase Central, but the coordinator computer could start functioning normally without warning. Ideally, the computer on which the coordinator was running should be shut down during the manual failover process.

Warning! Initiating manual failover while the former coordinator process is alive may cause database corruption.

❖ **Replacing the coordinator (Sybase Central)**

Make sure that the coordinator is really down before you replace it.

1 Right-click the multiplex set node in the Sybase Central tree view. The Failover wizard is only enabled when the coordinator is down and the designated failover node is running.

2 Specify the action to take against the current coordinator by choosing one of three options from the drop-down list: Drop it (the default), Keep it as Reader, or Keep it as Writer.

If you choose to drop the server, the Delete Server Files check box appears (deselected by default).

If you choose to keep the server as reader or writer, two radio buttons display: Included and Excluded (the default).

Choose Included or Excluded (the default). If you choose Included, the Synchronize After Failover check box appears. This check box is deselected by default.

3 Specify the new failover node by choosing a node from the Identify the New Failover Node dropdown.

4 Click Finish to start the failover process.

Two dialog boxes display.

5 Click Yes if you are certain that the coordinator is down and you are ready to fail over. Several progress messages display at the base of the wizard screen.

❖ **Replacing the coordinator (Command line)**

The coordinator process must be dead before you initiate replacement. The designated coordinator node must be included and part of the multiplex. Sybase recommends that you have a reader be the designated failover node. Readers have no pending writeable transactions, which makes failover easier.

1 Ensure that coordinator process is dead.

If there were any read-write transactions running on secondary nodes when the original coordinator was shut down, these transactions will be rolled back. Ideally if the coordinator is running on dedicated server hardware, that computer should be shut down during the failover process.

- On UNIX, log into the coordinator machine and make sure that the environment variables are set, then issue the following command:

```
stop_iq
```

and stop the appropriate iqsrv15 process.

- On Windows, log into the coordinator machine. Start Task Manager and look for the process name *iqsrv15.exe*. Stop the *iqsrv15.exe* process.

2 To identify the designated failover node, connect to any running multiplex server and execute the stored procedure sp_iqmpxinfo. The column coordinator_failover shows the designated failover node.

3 Connect to the designated failover node and run COMMIT, then BEGIN TRANSACTION to ensure that this node is up to date with the latest TLV log.

   Shut down the designated failover node cleanly, using Sybase Central (Right-click > Control > Stop) or the dbstop utility.

4 At the command line, restart the intended coordinator using the failover switch (-iqmpx_failover 1) on the server startup utility:

```
start_iq -STARTDIR/host1/mpx
@params.cfg -iqmpx_failover 1
-n mpxnode_w1 -x "tcpip{port=2764}"
mpxtest.db
```

Once the server startup is complete, the failover process is complete and the designated failover node becomes the new coordinator node. After failover, on the next transactions, other secondary servers recognize the new coordinator and connect to it for all read-write transactions. The former coordinator becomes a reader and can be started as a regular secondary node.

To start the former coordinator, you must synchronize it against the new coordinator. Follow steps 1 through 4 in “Synchronizing servers (command line)” on page 31 but in step 2 (dbbackup), the connection string specified with the -c parameter must contain the new coordinator's connection parameters.

Multiplex login management

For an overview of Sybase IQ login policies, see “Login management” in Chapter 8, “Managing User IDs and Permissions,” in *System Administration Guide: Volume 1*.

Multiplex extensions to login policy DDL

To enforce login policies on a per-server basis, the ALTER LOGIN POLICY has an optional clause, MULTIPLEX SERVER *server-name*.

Example

The following statement creates a login policy called *Reader1OnlyPolicy*. By default, this policy prohibits access to servers:

```
CREATE LOGIN POLICY Reader1OnlyPolicy
LOCKED=ON
```

This statement allows holders of *Reader1OnlyPolicy* to access only server *Reader1*:

```
ALTER LOGIN POLICY Reader1OnlyPolicy
LOCKED=OFF MULTIPLEX SERVER Reader1
```

UserA is allowed to log in only on *Reader1* and not on any other server in the multiplex:

```
ALTER USER UserA LOGIN POLICY
Reader1OnlyPolicy
```

CHAPTER 3

Running Multiplex Transactions

Multiplex transactions overview

A **local transaction** is any transaction that does not modify a shared object. Note that a local transaction may be read-only or read-write but modify data in local objects only (SA tables or temp tables). Local transactions may be performed on any multiplex node, and the effects of the transaction are only visible to that node.

A **global transaction** is any transaction that modifies data in shared objects or changes the schema of any persistent object. Global transactions may only be performed on the coordinator node or a writer node. The effects of a global transaction are visible on all nodes of the multiplex.

All transactions begin as local transactions. A transaction only becomes global when the first read-write operation (such as an insert) is initiated on a shared IQ object. When a global transaction is initiated from a secondary writer node, the coordinator node must be running, because the writer node must communicate with the coordinator node in order to coordinate the global aspects of the transaction.

In a writer-initiated global transaction, the writer node CPU and local disks are used to do the work of the read-write operation, while the coordinator node handles the global bookkeeping for the transaction, such as the assignment of global transaction IDs, global lock management, and writing the TLV log.

If the coordinator fails or is shut down during a writer-initiated global transaction, a “Coordinator node not responding” error occurs. Depending on the current state of the global transaction, one of the following will happen:

- If this error occurs during the initiation of the transaction, only the current command is rolled back and the transaction can continue.
- If a global transaction is already initiated, and this error occurs before committing the global transaction, that transaction cannot be committed and must be rolled back.
- If this error occurs during the commit of a global transaction, the user's connection is terminated.

Note If a global transaction initiated from a writer node modifies both global and local persistent objects (for example, an SA base table and an IQ base table), and the coordinator fails during commit, global object changes may be committed while local object changes are lost. This is consistent with the same scenario updating both local and proxy tables in the same transaction, where “best effort” is used to commit both local and global components of a transaction.

Running DML commands

In IQ 15.0, global transactions can be run from the coordinator server and any writer server. DML commands which modify tables in the shared IQ store are Global DML commands. Any transaction that contains a Global DML command becomes a Global Transaction.

Global DML

Global DML commands behave as if they were executed on the coordinator, and obey the same locking rules as on a single server. For example, if one transaction on any server has modified a shared IQ table, no other transaction may modify that table until the original transaction commits or rolls back. Whenever a global transaction commits, whether it runs on a writer node or the coordinator, the metadata for that global transaction is communicated to all servers in the multiplex through the TLV log.

Table data scope

When running DML commands in multiplex, the visibility of the table rows differs for different table types. There are three types of row visibility in multiplex:

- Global scope – All connections on all servers can see the rows.
- Server scope – All connections on a particular multiplex server can see the rows.
- Connection scope – Only a single connection on a particular multiplex server can see the rows.

Table 3-1 describes the scope pertaining to different table types:

Table 3-1: Multiplex table data scope

Table type	Data scope
IQ base	Global
IQ temporary	Connection
Global temporary table	Connection
SA catalog (table created IN SYSTEM)	Server
SA temporary (table created IN SYSTEM)	Connection

Sybase IQ copies data in SA catalog tables on the coordinator node to the secondary node catalog store whenever a secondary node is synchronized. The data in catalog store tables is otherwise not related between secondary server and coordinator server instances. After synchronization, local SA table data on the secondary server is lost, because table data from the coordinator overwrites it.

Note Because CIS proxy tables point to data on remote servers, the data scope is external. Data management for such tables is done by the remote server.

Any schema change for tables participating in a join index is not allowed in a 15.0 or higher multiplex. Any DML operation permitted on a table participating in a join index is allowed only on the coordinator, which must be in single node mode. See “Server startup switches” on page 98.

Running DDL commands

Statement type and object type affect the scope of Data Definition Language (DDL) statements in a multiplex.

Statement scope

DDL statements may be propagated to all nodes or be local to the executing node, as follows:

- Local scope – execute on the local server and affect the local catalog store or local temporary store only.
- Global scope – execute on the coordinator and affect the shared IQ store and global catalog store. The coordinator writes statements with global scope to the TLV log on commit.

Determining scope

DDL commands that affect temporary objects, which lack a permanent object ID, are local. This includes these object types:

- Local temporary tables
- Local procedures
- Temporary options

DDL commands that create, alter, or drop a permanent object ID in the ISYSOBJECT table are global. This includes these object types:

- Table – includes SA base tables, IQ base tables, and global temporary tables
- View
- Materialized view (SA tables only)
- Column
- Index
- Procedure
- Event
- User

- Publication
- Remote type
- Login mapping
- JAR
- Java class
- Service
- Database options (minus locally scoped option exceptions)

You cannot run global DDL commands if the coordinator node is not running. Attempting to do so results in the error `SQLCODE: -1004011, SQLSTATE QIA11: Coordinator node not responding`.

When creating objects that are global, make sure that they do not depend on objects that are local. For example, if you created this temporary function or stored procedure:

```
CREATE TEMPORARY FUNCTION f1() RETURNS INT
BEGIN
RETURN 1;
END
```

Next, if you tried to create a view that is dependent on the temporary function:

```
CREATE VIEW v1 AS SELECT * FROM f1()
```

You would receive the error `Procedure 'f1' not found` since it is not a permanent procedure. Sybase IQ does not allow such operations in a multiplex environment.

Consider a second example where you create a global object with a dependency on a local object. Assume you create the lineitem temporary table on a secondary node:

```
DECLARE LOCAL TEMPORARY TABLE #lineitem (
  l_orderkey       integer,
  l_partkey        integer     iq unique(20000000),
  l_suppkey        integer     iq unique(20000000),
  l_linenumber     integer,
  l_quantity       integer     iq unique(50),
  l_extendedprice  double,
  l_discount       double      iq unique(11),
  l_tax            double      iq unique(9),
  l_returnflag     char(1)     iq unique(3),
  l_linestatus     char(1)     iq unique(2),
  l_shipdate       date        iq unique(270),
```

```
  l_commitdate     date        iq unique(256),
  l_receiptdate    date        iq unique(300),
  l_shipinstruct   char(25),
  l_shipmode       char(10)    iq unique(7),
  l_comment        char(44)
)
```

Next, you create indexes—which are global objects—on the columns of the lineitem temporary table using the BEGIN PARALLEL IQ command:

```
BEGIN PARALLEL IQ
CREATE LF INDEX LFIQ_IDX_TXXX_CXX_L_PK on #lineitem
(l_partkey);
CREATE LF INDEX LFIQ_IDX_TXXX_CXX_L_OK on #lineitem
(l_orderkey);
END PARALLEL IQ
```

Sybase IQ returns the error `Table 'lineitem' not found` because the BEGIN PARALLEL IQ command is a global command sent to the coordinator node, but the lineitem table is a local temporary table on the secondary node.

Role restriction

Some statements are restricted to nodes of certain roles:

- Coordinator – Statements with a coordinator role restriction only run on a coordinator node.
- Writer – Statements with a writer role restriction run on a writer node or a coordinator.
- None – Statements with no role restriction run on any node in the multiplex.

Table 3-2 lists statements with role restrictions. Unlisted statements are unrestricted.

Table 3-2: Role restricted commands

Coordinator role restriction	Writer role restriction
• All CREATE/ALTER/DROP DBSPACE commands operating on IQ main store dbspaces • BACKUP DATABASE • LOCK TABLE • sp_iqemptyfile	All DDL commands that affect objects in the IQ main store dbspaces. This includes ALTER/DROP of: • Tables • Single and multicolumn indexes • Table constraints

Preserving rows

In Sybase IQ 15.2, a global temporary table created with ON COMMIT PRESERVE ROWS cannot be altered or dropped from a secondary node if the instance of the global temporary table on the connection of the secondary node executing the drop table has data. You need to truncate the table and try again or alter or drop the table from the coordinator node. If the global temporary table is created with ON COMMIT DELETE ROWS, you may alter or drop the table even if it has rows.

For example, connect to a secondary node:

```
CREATE GLOBAL TEMPORARY TABLE
foo_gtt(c1 int)
ON COMMIT PRESERVE ROWS;
INSERT INTO foo_gtt VALUES(200);
COMMIT;
DROP TABLE foo_gtt;
```

The drop statement fails with the following error:

```
Operation not allowed on global temporary
table foo_gtt as it is in use. Please reconnect and
try again. SQLCODE=1287. ODBC 3
State="HY000"
```

Updating dbspaces in multiplex

Before updating dbspaces, see the overview of dbspaces and dbfiles in “Configurable tablespaces” in Chapter 1, “Overview of System Administration,” in *System Administration Guide: Volume 1*.

In multiplex, CREATE DBSPACE, ALTER DBSPACE and DROP DBSPACE commands for main and catalog dbspaces are only permitted on the coordinator. Due to delays in TLV propagation and version maintenance, there may be a timing window of up to a few minutes before you can drop an empty dbfile or dbspace. A "Command not replayed" error will occur during this timing window. In order to perform the DROP DBSPACE or ALTER DBSPACE DROP FILE commands, the OkToDrop column reported by the sp_iqdbspace and sp_iqfile procedures must indicate "Y."

The following rules affect multiplex dbspace updates:

- Only the coordinator node is allowed to manipulate shared IQ dbspaces.
- Secondary servers must be in synch before you run ALTER DBSPACE ALTER FILE RENAME PATH. Synchronization happens automatically through TLV log replay, but there may be a time delay of up to two minutes between the previous dbspace operation on this dbspace (create or alter) and the time the rename path can be run. If secondary servers are out of synch, you may see a "Command not replayed" error.
- Temporary dbspace files must be created/altered/dropped from the node where they reside. Syntax is identical to simplex temporary dbspace syntax.
- If any shared IQ main dbspace files have paths that are not accessible from a secondary node, that secondary node cannot access the file or any contents of that file until the path is corrected. See "Accessing the shared disk array" on page 5.

Updating the IQ_SYSTEM_MAIN dbspace

The IQ_SYSTEM_MAIN dbspace manages important database structures including the freelist, which tracks which blocks are in use. Sybase recommends shutting down secondary nodes before adding space to IQ_SYSTEM_MAIN. In a 15.0 or higher multiplex, if a dbfile is added to IQ_SYSTEM_MAIN, all running secondary nodes shut down automatically and the IQ message file for the secondary node reports:

```
Multiplex secondary node shutting down
due to a file added to the IQ_SYSTEM_MAIN dbspace.
This node must be synchronized and restarted.
```

This behavior is an exception only for IQ_SYSTEM_MAIN and does not apply to any other dbspace. Similarly, for other dbspace operations there is no disruption and all nodes in the multiplex continue to run. See "Adding space to IQ_SYSTEM_MAIN on a multiplex coordinator" on page 50.

After adding new files to IQ_SYSTEM_MAIN, synchronize and restart secondary nodes. The path of all dbfiles in IQ_SYSTEM_MAIN must be visible to the secondary nodes. If a file path for a dbfile in IQ_SYSTEM_MAIN is not visible to the secondary node, it fails to start, and reports `Error opening DBFILE 'filepath'`.

❖ **Changing an IQ_SYSTEM_MAIN file path**

If you need to change an IQ_SYSTEM_MAIN file path that cannot be resolved by creating links, follow these steps on the coordinator:

1 Shut down all the servers in the multiplex.

2 Start the coordinator in single node mode using the -iqmpx_sn 1 switch.

3 Make the dbfile read-only. For example:

```
ALTER DBSPACE IQ_SYSTEM_MAIN ALTER FILE
mainfile READONLY
```

4 Run the sp_iqemptyfile procedure against *mainfile*. For example:

```
sp_iqemptyfile mainfile
```

5 Drop the dbfile. For example:

```
ALTER DBSPACE IQ_SYSTEM_MAIN DROP FILE
mainfile
```

6 Add the dbfile with the path visible to all secondary nodes. For example, on UNIX:

```
ALTER DBSPACE IQ_SYSTEM_MAIN ADD FILE
mainfile '/dev/rdsk/c4t600A0B80005A7F5D0000024'
```

For example, on Windows:

```
ALTER DBSPACE IQ_SYSTEM_MAIN ADD FILE
mainfile '\\\\.\\PhysicalDrive1'
```

7 Restart the coordinator normally, without the -iqmpx_sn 1 switch.

8 Synchronize secondary nodes. For details, see "Synchronizing secondary servers" on page 30.

❖ **Adding space to IQ_SYSTEM_MAIN on a multiplex coordinator**

If the coordinator runs out of space in IQ_SYSTEM_MAIN, including reserve space, it may abort to prevent database corruption and you may be unable to start the coordinator normally, because the coordinator may require space during recovery. In that case, start the coordinator in a single node mode using the -iqmpx_sn startup switch, and add more space by adding file(s) to IQ_SYSTEM_MAIN. Restart the coordinator normally and synchronize all secondary nodes, as in the following procedure.

When the coordinator needs space in IQ_SYSTEM_MAIN, proceed as follows:

1 Shut down all servers in the multiplex.

2 Start the coordinator in single node mode using -iqmpx_sn 1. For example:

```
start_iq @params.cfg mpxtest.db -iqmpx_sn 1 -n
mpxnode_c
```

3 To add more space to IQ_SYSTEM_MAIN with a new file on a raw device, use syntax like the following:

Windows:

```
ALTER DBSPACE IQ_SYSTEM_MAIN ADD FILE
mymainfile '\\.\PhysicalDrive3'
```

Unix:

```
ALTER DBSPACE IQ_SYSTEM_MAIN ADD FILE mymainfile
'/dev/rdsk/c4t600A0B80005A7F5D0000024'
```

where *mymainfile* is the logical or chosen logical name of the new dbfile.

4 Restart the coordinator normally, without the -iqmpx_sn 1 switch.

5 Synchronize secondary nodes. For details, see "Synchronizing secondary servers" on page 30.

Updating user dbspaces

❖ **Adding a dbfile to a user dbspace**

1 Connect to a coordinator.

2 Add a file to a user dbspace or add a new user dbspace. On the ADD FILE clause, specify either a full path to a raw device, or a soft link.

For example, this statement specifies a full path on a UNIX system:

```
ALTER DBSPACE mydbspace ADD FILE mydbfilename
'/dev/rdsk/c4t600A0B80005A7F5D0000024'
```

For example, this statement specifies a full path on a Windows system:

```
ALTER DBSPACE mydbspace ADD FILE mydbfilename
'\\\\.\\PhysicalDrive2'
```

For example, this statement specifies a soft link on a UNIX system:

```
ALTER DBSPACE mydbspace ADD FILE mydbfilename
'store/userdb1'
```

You do not need to synchronize or restart any secondary servers.

Updating catalog dbspaces

Under normal circumstances, you will not need to add or remove catalog dbspaces, as catalog dbspace files automatically grow as needed. However, if your catalog dbspace file is on constrained storage, you may need to add a new catalog dbspace to accommodate catalog data.

❖ **Adding a catalog dbspace**

1 Shut down all servers in the multiplex.

2 Start the coordinator in single node mode using -iqmpx_sn 1. For example:

```
start_iq @params.cfg mpxtest.db -iqmpx_sn 1
-n mpxnode_c
```

3 Create the dbspace.

```
CREATE DBSPACE DspCat2 AS 'sadb2.db'
CATALOG STORE
```

4 Restart the coordinator normally, without the -iqmpx_sn1 switch. For example:

```
start_iq @params.cfg mpxtest.db -n mpxnode_c
```

5 Synchronize all secondary servers in the multiplex. For details, see "Synchronizing secondary servers" on page 30.

❖ **Dropping a catalog dbspace**

Follow this procedure to drop a dbspace from the catalog store.

1 Shut down all the servers in the multiplex.

2 Start the coordinator with the -gm and -iqmpx_sn switches.

```
start_iq @params.cfg mpxtest.db -gm 1
```

```
-iqmpx_sn 1 -n mpxnode_c
```

3 Drop the catalog dbspace in Interactive SQL.

```
DROP DBSPACE DspCat2
```

4 Restart the coordinator normally, without the -iqmpx_sn 1 switch.

5 Synchronize all secondary servers in the multiplex. See "Synchronizing secondary servers" on page 30.

Handling dynamic collisions

Dynamic collisions only occur on secondary nodes. A dynamic collision occurs when a schema change is committed on a table, procedure, function, or view while a secondary server is querying the same object. The collision results from the TLV replay of shared IQ object schema updates.

In an interactive setting, a dynamic collision results in a forced disconnect of a user connection to a secondary node. To resolve a dynamic collision during DDL execution, the secondary server finds the connections responsible and disconnects them. The disconnect is logged in the *.iqmsg* file. Here is an example of a message that appears in the *.iqmsg* file when a dynamic collision occurs:

```
Shared IQ Store update DDL statement:
drop table DBA.gtt44
Disposition: SQLSTATE:42W21 --
dropped 1 connection(s) for table:
DBA.gtt44 Retry successful
```

To avoid dynamic collisions that can disrupt queries and disconnect users, schedule schema changes when the multiplex is not heavily loaded. This will result in the least disruption to query operations on secondary servers.

CHAPTER 4

Monitoring with Sybase Central

Monitoring multiplex state and status

The graphical detail pane is available from the Multiplexes container in the Sybase IQ plug-in for Sybase Central. The graphical, topology view visually represents the entire multiplex environment, and complements the existing list view, making it easy to visually monitor the state of the multiplex environment.

The view consists of nodes and links. A node represents a multiplex server, while a link represents a connection between two multiplex servers. Links exist only between the coordinator node, of which there can be only one, and a secondary node. Links cannot exist between two secondary nodes.

You can rearrange the layout by moving nodes in the view; the next time you display the topology view, the most-recently used graphical layout is restored as nearly as possible. (If the multiplex configuration has changed, the layout can be only partially restored.)

You can use the topology view to designate a failover node and to perform failover. See "Designated failover node" on page 36 and "Failover" on page 36.

❖ **Viewing multiplex topology**

The topology view in Sybase Central shows the types of servers in the multiplex and their current states.

- In the Sybase Central tree view, select the multiplex name, then click the Topology tab.

You can also print and save the topology view.

❖ **Printing the multiplex topology**

1. Select the multiplex name in the Sybase Central tree view, and switch to the Topology tab.
2. Right-click in the Topology view, and select Print Graph.
3. Review your print options and print the page to the desired printer.

❖ **Saving the graph**

1. In the topology view, right-click and select Save Graph As.
2. Specify a file name for the .JPEG file.
3. Click Save.

Updating the topology view

The top of the topology pane displays a status message that logs the last updated timestamp, a summary of server status, and, if raised, a status change event message.

Sybase Central monitors the status of the multiplex and automatically refreshes the topology view and status message when servers are deleted or excluded. You can also specify a refresh interval.

❖ **Changing the refresh rate of multiplex status monitor**

1. Right-click in the topology view and select Change Monitor Rate.
2. Specify the number of seconds between status checks.
3. Click OK.

Viewing node information

In addition to the graphic display, the topology view also displays some text information about servers.

❖ **Viewing node information**

1. View labels beside each icon for the server name and mode.
2. Move the mouse over a server icon to display a tooltip that shows the host, port, state, role, and status.

internode communication state

Each secondary node has a communication link to the coordinator, displayed as a line. The label, style and color of the link show internode communication (INC) state:

- Active – solid green line.
- Not Responding – dashed orange line.
- Timed Out – dotted red line.
- Excluded – link contains label "(Excluded)."

Right-click on any link to see a popup menu with a property sheet that provides the state and statistics about the heartbeat and connection pool. See "Heartbeat connections" on page 8 and "Pooled connections" on page 8.

Server mode

Node labels display the server name and mode:

- Coordinator – R/W server that allocates work and maintains configuration data for the multiplex. There can be only one coordinator per multiplex.
- Reader – R/O secondary server.
- Writer – R/W secondary server.

Server state

The server icon indicates the server state:

- Running – if currently connected, green. If not connected, gray with an adaptor (electrical plug) in the upper-right corner.
- Stopped – red with down arrow.

Server status

The server icon indicates the server state:

- Included.
- Excluded – "(Excluded)" label.

Monitoring performance statistics

The Performance Monitor displays a collection of statistics for one or more participating nodes. Statistics display in a dynamic chart in real time.

You can access the performance monitor at two different levels:

- Multiplex level – monitor only one statistic, across multiple servers.
- Server level – on a single server or a multiplex server, monitor up to ten statistics at a time.

In the Sybase Central tree view, select the server or multiplex, then click the Performance Monitor tab. You can change the type or contents of the performance monitor graphs.

❖ **Configuring data collection rates**

The Configure Collection Rates dialog is available only for the server-level performance monitor. Use this dialog to change the rates at which statistics are collected. When the performance monitor displays, a multi-threaded data collection engine collects monitored statistics at two different rates. Less-expensive statistics are collected more frequently, while more-expensive statistics are collected less frequently. To change this default collection rate, configure the rates yourself.

1. In the Sybase Central tree view, select the server, then click the Performance Monitor tab.
2. Right-click in the chart area and select Configure Collection Rates.
3. Select the rates, in seconds, for statistics collection.
4. To reduce the impact on server performance, increase the rate values so that statistics are collected less frequently.
5. Click OK.

❖ **Changing the statistics that are monitored**

The Configure Collection Rates dialog behaves differently for server-level monitoring and multiplex-level monitoring:

- When you launch this dialog from the server-level performance monitor (the server can be a single server or a multiplex server), you can select up to ten statistics to monitor at a time.
- When you launch it from the multiplex-level performance monitor, you can only monitor one statistic at a time.

1. Click the multiplex server, then click the Performance Monitor tab.
2. Right-click in the chart area and select Change Statistics.
3. Choose the statistics to monitor. "Monitoring performance statistics" on page 56 lists all available statistics.
4. To save any changes, click OK.

❖ **Changing the servers that are monitored**

The Change Servers to Monitor dialog is available only for the multiplex-level performance monitor.

1. Click the multiplex and switch to the Performance Monitor tab.
2. Right-click in the Performance Monitor chart and select Change Servers.
3. Select each server to monitor. Deselect any servers not to be monitored.
4. To save any changes, click OK.

❖ **Saving the chart**

You can save the chart as a .JPEG image file.

1. Right-click in the Performance Monitor chart and select Save Chart As.
2. Specify a file name for the .JPEG file.
3. Click Save.

❖ **Printing the chart**

You can print the performance monitor chart.

1. Right-click in the Performance Monitor chart and select Print Chart.
2. Review your print options and send the page to the desired printer.

❖ **Switching chart view**

You can choose between time-series, 2-D bar, and 3-D bar chart views.

1. Right-click the Performance Monitor chart and select Chart View.
2. Select Time Series Chart, Bar Chart 2-D Vertical, or Bar Chart 3-D Vertical.

❖ **Customizing the chart**

To change chart settings and chart refresh (monitor GUI heartbeat) rate:

1 Right-click in the performance monitor chart area and select Customize Chart.

Change any of these components:

- Time Window – appears only if the chart is a Time Series chart. Specify the period of time during which the data tracked to produce chart statistics. The minimum value is 1 minute. The maximum value is 240 minutes (4 hours).
- Chart Refresh Rate – specify how often, in seconds, the data in the chart is refreshed. The chart refresh rate also shows at the bottom of the Performance Monitor panel.
- Real vs. Normalized Value – Real Value (the default) reflects actual data. Normalized Value scales chart data into a fixed range. This option is for display purposes only, so that statistics with different ranges display better in one chart.
- Legend – select to display or hide the legend.

2 To save any changes, click OK.

Categories of statistics

Statistics are grouped into these categories:

- CPU usage statistics
- Memory usage statistics
- Cache statistics
- Thread statistics
- Connection statistics
- Request statistics
- Transaction statistics
- Store I/O statistics
- Dbspace usage
- Network statistics

CPU usage statistics

Table 4-1: CPU usage

Name	Description	Monitored by default?
CPU Usage	IQ process CPU usage percentage, including both system and user usage	Yes
CPU System Usage	IQ process CPU system usage percentage	No
CPU User Usage	IQ process CPU user usage percentage	No

Memory usage statistics

Table 4-2: Memory usage

Name	Description	Monitored by default?
Memory Allocated	Memory allocated by the IQ server, in megabytes	Yes
Maximum Memory Allocated	Maximum memory allocated by the IQ server, in megabytes	No

Cache statistics

Table 4-3: Cache statistics

Name	Description	Monitored by default?
Catalog Cache Hits	Number of catalog cache hits per second	No
Temporary Cache Hits	Number of temporary cache hits per second	No
Main Cache Hits	Number of main cache hits per second	No
Catalog Cache Reads	Number of catalog cache page lookups per second	Yes
Temporary Cache Reads	Number of temporary cache page lookups per second	No

Name	Description	Monitored by default?
Main Cache Reads	Number of main cache page lookups per second	No
Catalog Cache Current Size	Current catalog cache size, in megabytes	No
Temporary Cache Current Size	Current temporary cache size, in megabytes	No
Main Cache Current Size	Current main cache size, in megabytes	No
Catalog Cache in Use Percentage	Percentage of catalog cache in use	No
Temporary Cache in Use Percentage	Percentage of temporary cache in use	No
Main Cache in Use Percentage	Percentage of main cache size in use	No
Catalog Cache Pinned	Number of pinned catalog cache pages	No
Temporary Cache Pinned	Number of pinned temporary cache pages	No
Main Cache Pinned	Number of pinned main cache pages	No
Catalog Cache Pinned Percentage	Percentage of catalog cache pinned	No
Temporary Cache Pinned Percentage	Percentage of temporary cache pinned	No
Main Cache Pinned Percentage	Percentage of main cache pinned	No
Catalog Cache Dirty Pages Percentage	Percentage of catalog cache dirty pages	No
Temporary Cache Dirty Pages Percentage	Percentage of temporary cache dirty pages	No
Main Cache Dirty Pages Percentage	Percentage of main cache dirty pages	No

Thread statistics

Table 4-4: Thread statistics

Name	Description	Monitored by default?
IQ Threads in Use	Number of threads used by the IQ server	No
IQ Threads Available	Number of threads available in the IQ server	No
SA Threads in Use	Number of threads used by the SQL Anywhere engine.	No

Connection statistics

Table 4-5: Connection Statistics

Name	Description	Monitored by default?
Total Connections	Total number of connections including user and INC connections.	Yes
User Connections	Number of user connections.	No
INC Incoming Connections	Number of INC incoming connections	No
INC Outgoing Connections	Number of INC outgoing connections	No
User Connections Per Minute	Number of user connections per minute	No
User Disconnections Per Minute	Number of user disconnections per minute	No

Request statistics

Table 4-6: Request statistics

Name	Description	Monitored by default?
Requests	Number of times per second the server has been entered to allow it to handle a new request or continue processing an existing request	No

Name	Description	Monitored by default?
Unscheduled Requests	Number of requests that are currently queued up waiting for an available server thread	No
IQ Waiting Operations	Number of IQ operations waiting for the resource governor	No
IQ Active Operations	Number of active IQ operations	No

Transaction statistics

Table 4-7: Transaction statistics

Name	Description	Monitored by default?
Total Transaction Count	Total number of active transactions including user and INC transactions	No
User Transaction Count	Number of active user transactions	No
INC Transaction Count	Number of active INC transactions	No
Active Load Table Statements	Number of active load table statements	No

Store I/O statistics

Table 4-8: Store I/O statistics

Name	Description	Monitored by default?
Catalog Store Disk Reads	Number of kilobytes per second that have been read from the catalog store	No
Temporary Store Disk Reads	Number of kilobytes per second that have been read from the temporary store	No
Main Store Disk Reads	Number of kilobytes per second that have been read from the main store	No
Catalog Store Disk Writes	Number of kilobytes per second that have been written to the catalog store	No
Temporary Store Disk Writes	Number of kilobytes per second that have been written to the temporary store	No
Main Store Disk Writes	Number of kilobytes per second that have been written to the main store	No

Dbspace usage

Table 4-9: Dbspace Usage

Name	Description	Monitored by default?
Per DBSpace Free Space Percentage	Percentage of free space available for every dbspace. There is one such statistic per dbspace.	No
Per DBSpace Size in Use	Dbspace size in use. There is one such statistic per dbspace.	No
Per DBSpace Per File Free Space Percentage	Percentage of free space available for every dbspace file. There is one such statistic per dbspace per file.	No

Name	Description	Monitored by default?
Per DBSpace Per File Size in Use	Dbspace file size in use. There is one such statistic per dbspace per file.	No

Network statistics

Table 4-10: Network statistics

Name	Description	Monitored by default?
Bytes Received	Number of bytes per second received during client/server communications	Yes
Bytes Received Uncompressed	Number of bytes per second received during client/server communications if compression is disabled	No
Bytes Sent	Number of bytes per second sent during client/server communications	Yes
Bytes Sent Uncompressed	Number of bytes per second sent during client/server communications if compression is disabled	No
Free Communication Buffers	Number of available network communication buffers	No
Total Communication Buffers	Total number of network communication buffers	No

CHAPTER 5

Backing Up and Restoring Data in a Multiplex Environment

Backing up and restoring data in a multiplex environment

This section describes:

- Notes on backing up and restoring multiplex servers and databases.
- Special restrictions that apply to backup and restore operations in a multiplex environment

You can also use the restore operation to re-create a multiplex on a different system when no problems have occurred.

You can execute the BACKUP and RESTORE SQL commands only on the coordinator node. For complete syntax, see BACKUP statement and RESTORE statements in Chapter 1, "SQL Statements," in *Reference: Statements and Options*. To back up the IQ store and catalog store on a multiplex database, log in to the coordinator using an account with DBA or backup authority. During restore operations, the database can be running only if you restore a backup of read-only files. When restoring files in a read-only dbspace, the dbspace must be offline.

Multiplex backup list

Back up the IQ store as described in "Types of backups" in Chapter 12, "Data Backup, Recovery, and Archiving," in *System Administration Guide: Volume 1*. The last step of both IQ-level and system-level restore operations is to propagate changes by synchronizing the secondary servers.

You may want to preserve the *server.dbrlog.NNN* files (stored in the write server's directory under */repDirs/logfiles* on UNIX or *\repDirs\logfiles* on Windows).

Updating your virtual backup list

If you are using virtual backup, you must add to your system backup specification all the main store dbfiles that are specified in the backup. Use the stored procedure sp_iqfile to create the system backup list.

Use the stored procedures sp_iqbackupsummary, sp_iqbackupdetails, and sp_iqrestoreaction, the system views SYSIQBACKUPHISTORY and SYSIQBACKUPHISTORYDETAIL, and the db_backupheader utility to track backups and plan restore actions.

Raw devices and symbolic links

If you use symbolic links for raw device names, as Sybase recommends, make sure the system backup utility follows the symbolic link and backs up the device.

Checking database consistency

The sp_iqcheckdb procedure can perform several different functions, depending on the parameters specified. The four modes for checking and repairing database consistency are described in "sp_iqcheckdb procedure" in Chapter 7, "System Procedures," in *Reference: Building Blocks, Tables, and Procedures*.

You may run sp_iqcheckdb on any multiplex node, but on a secondary server the procedure cannot access the freelist, so no freelist checks are performed.

Start a multiplex coordinator single-node mode (-iqmpx_sn 1) to run sp_iqcheckb dropleaks. You cannot run dropleaks mode on multiplex secondary nodes.

Restoring in the multiplex environment

Verify with Sybase Technical Support that a restore operation is necessary. If you cannot open your database on a secondary server, synchronize the server, following the instructions in "Synchronizing secondary servers" on page 30. It is never necessary to restore a coordinator node due to secondary node problems.

You can perform read-write restore operations only against a server running the utility database. You cannot run restore operations against a secondary server.

❖ **Restoring IQ store data when restoring database in same location**

1 Confirm that database home directories for each server still exist. If not, create them or restore them from file system backups.

2 Shut down every server in the multiplex (coordinator and all secondary servers). In Sybase Central, right-click the multiplex icon and use the Stop Server wizard. The wizard lets you select servers to stop.

Note If automatic startup is enabled in your ODBC configuration, users on the same machine as the server may be set up to start the server automatically. Prevent this from happening while you are restoring the database.

3 After stopping all servers, confirm that the database shut down successfully. If you see an active iqsrv15 process with name of a multiplex, stop the process.

To confirm database shutdown on a UNIX system, use the ps command. For example:

```
% ps -ef | grep iqsrv15
fiona 434     1  1   May 19 ?? 0:05 start_iq
-n myhost_myserver
-c 32MB -x tcpip(port=1234) mpxdb.db
fiona 4751   442  1 16:42:14 pts/5
0:00 grep start_iq
```

To confirm database shutdown on a Windows system, use Task Manager. Look on the Processes tab for *iqsrv15.exe* or find the IQ Server icon in the system tray and stop it using right-click and Shutdown.

4 Move files required for debugging and reconfiguring the multiplex.

- Make a file system copy of the *.iqmsg* file. If you have message log archiving configured, see "Backing up the right files," in *System Administration Guide: Volume 1*.
- On each server, preserve any files that were added to IQ_SYSTEM_TEMP for that server. These files are of the form *dbname.iqtmp* if you used an OS file, or they may be raw devices. If the IQ temporary store is damaged, start the server with the -iqnotemp switch to drop and recreate the temporary store dbspaces. For more information, see the *Release Bulletin.*

  Either drop the database or delete the following files from the coordinator:

  ```
  <database_home>/<dbname>.db
  ```

  ```
  <database_home>/<dbname>.log
  ```

  If a query server is damaged, however, drop it and re-create it after RESTORE. Then follow the instructions in "Restoring IQ store data when restoring database in a different location."

5 Start the utility database from the coordinator server directory using the coordinator server's name:

```
% start_iq -n thoreau_Server01 -c 32MB
-x tcpip(port=1234)
```

6 Connect to the utility database (utility_db).

```
% dbisql -c "eng=thoreau_Server01;uid=DBA;pwd=SQL;
dbn=utility_db"
```

7 Run the RESTORE command. To restore certain dbspace files to a different path, specify a RENAME clause. For details, see the RESTORE statement in *Reference: Statements and Options*.

8 Shut down the utility database.

9 Make sure that the temporary dbspaces exist as before, on raw devices or as files of the correct length. See "Backing up the right files," in Chapter 12, "Data Backup, Recovery, and Archiving," in *System Administration Guide: Volume 1*. For information on starting the server without using the IQ temporary store, see the *Release Bulletin* for your platform.

10 Start the coordinator server and, *if restoring to the same location*, synchronize the secondary servers. For more information, see "Synchronizing secondary servers" on page 30.

11 Start the secondary servers.

❖ **Restoring IQ store data when restoring database in different location**

1 Confirm that database home directories for each server still exist. If not, create them or restore them from file system backups.

2 If this is not the first time you have restored to the new location, shut down all multiplex servers running at the destination location (coordinator and secondary servers). In Sybase Central, right-click the multiplex and use the Stop Server wizard. The wizard lets you stop one or more servers. The multiplex at the original location where the backup was taken may continue running.

Note If automatic startup is enabled in your ODBC configuration, users on the same machine as the server may be set up to automatically start the server. Prevent this from happening while you are restoring the database.

3 After stopping all servers, confirm that the database shut down successfully. If you see an active iqsrv15 process with the name of a multiplex server, stop the process.

To verify on a UNIX system, use the ps command. For example:

```
% ps -ef | grep iqsrv15
fiona 434     1  1   May 19 ?? 0:05 start_iq
-n myhost_myserver
-c 32MB -x tcpip(port=1234) mpxdb.db
fiona 4751   442  1 16:42:14 pts/5
0:00 grep start_iq
```

To verify on a Windows system, use Task Manager. Look on the Processes tab for *iqsrv15.exe* or find the IQ Server icon in the system tray and stop it using right-click and Shutdown.

4 Move files required for debugging and reconfiguring the multiplex.

- Make a file system copy of the *.iqmsg* file. If you have message log archiving configured, see "Backing up the right files," in *System Administration Guide: Volume 1*.

5 Start the utility database from the coordinator server directory using the new coordinator server's name:

```
% start_iq -n thoreau_Server01 -c 32MB
-x tcpip(port=1234)
```

6 Connect to the utility database (utility_db).

```
% dbisql -c "eng=thoreau_Server01;uid=DBA;pwd=SQL;
dbn=utility_db"
```

7 Run the RESTORE command with a new location path for the database. To restore certain dbspace files to a different path, specify a RENAME clause. For details, see the RESTORE statement in *Reference: Statements and Options*.

8 Start the restored database either by reconnecting to the utility_db server and specifying the restored database file name or by stopping the server and restarting it with the restored database. If you restart the server, use the single-node and override flags (-iqmpx_sn 1 -iqmpx_ov 1).

9 Use DROP MULTIPLEX SERVER statement to drop all the secondary nodes. For example, `DROP MULTIPLEX SERVER node_w3_skm`.

Once you drop the last secondary node, the coordinator shuts down automatically, signifying conversion to simplex.

10 Restart the coordinator without the single node or override switch.

11 Recreate all the secondary nodes with the correct location path. For example:

```
CREATE MULTIPLEX SERVER node_r2_skm DATABASE
'/sunx3005/mpx_simdb.db'
HOST 'localhost' PORT 8998
ROLE READER STATUS INCLUDED
```

After you create the first secondary node, the server automatically shuts down, signifying conversion to multiplex.

Note In the preceding command, you must include the database file extension *.DB*.

12 When you restart the coordinator, you see a warning in the server log about the multiplex environment being invalid. This warning is generated if IQ_SYSTEM_TEMP dbspace does not contain any files, and is the case for all the secondary nodes you created in step 12. Ignore this warning for now.

13 Synchronize the secondary servers, following the instructions in "Synchronizing secondary servers" on page 30.

14 Start the secondary servers.

15 Connect to each secondary server and add files in IQ_SYSTEM_TEMP.

16 Run sp_iqmpxvalidate on the coordinator. It should report `no error detected`.

To restore an exact copy of the multiplex to a different location, when copies of all of the server's temporary files exist at the new location, you replace steps 9 through 17 with:

Use ALTER MULTIPLEX SERVER to alter the server name, host, port, and database path of each server.

Selective restore operations in a multiplex environment

As of IQ version 15.0, you can restore the database either selectively (by restoring only read-write dbspaces, or a set of read-only dbspaces or read-only files), or completely. For details, see Chapter 12, "Data Backup, Recovery, and Archiving, in the *System Administration Guide: Volume 1* and RESTORE statement in *Reference: Statements and Options*.

APPENDIX A

Multiplex Reference

SQL statements

The following statements include syntax to support multiplex functionality.

ALTER LOGIN POLICY statement

Description

Modifies some or all option values for existing login policies in the database on the named multiplex server.

Syntax

ALTER LOGIN POLICY *policy-name policy-options*
MULTIPLEX SERVER *server-name*

Parameters

policy-options:

policy-option [*policy-option*...]

policy_option:

policy-option-name =*policy-option-value*

policy-option-value={ **UNLIMITED** | **ROOT** | *value* }

Example

The following example alters the *MpxUsers* login policy. This example changes the locked and max_connections options. The locked value indicates that users with the policy are prohibited from establishing new connections and the max_connections value indicates the number of concurrent connections allowed on the named multiplex server.

```
ALTER LOGIN POLICY MpxUsers locked=ON
max_connections=4 MULTIPLEX SERVER mpxsrv1;
```

Permissions

Must have DBA or USER ADMIN authority.

ALTER MULTIPLEX RENAME statement

Syntax

ALTER MULTIPLEX RENAME multiplex-name

Usage

Renames the multiplex and stores the multiplex name in SYS.ISYSIQINFO system table.When a multiplex is created, it is named after the coordinator. The multiplex name is only used in Sybase Central to identify a multiplex in the Multiplexes folder. This statement is automatically committed.

Permissions

Must have DBA or MULTIPLEX ADMIN authority.

ALTER MULTIPLEX SERVER statement

Syntax

ALTER MULTIPLEX SERVER *server-name server-option*

Parameters

server-option:

{ RENAME *new-server-name*
| DATABASE '*dbfile*'
| ROLE { WRITER | READER | COORDINATOR }
| STATUS { INCLUDED | EXCLUDED }
| ASSIGN AS FAILOVER SERVER
| *host-port-list* }

host-port-list: { HOST ' *hostname* ' PORT *port number* }

Usage

Changes the multiplex server, as follows:

RENAME Changes the name of the given server. The server automatically shuts down. The next restart requires the new name.

DATABASE Changes the catalog file path for the given server. The server will automatically shutdown and next time it should be started using new catalog path. Its user's responsibility (Sybase central might hide it) to relocate the catalog file itself.

ROLE Changes the role of the given server. Users are not allowed to change the role of coordinator or role to coordinator. If the writer node's role is changed to reader, the server shuts down.

STATUS Changes the status of the given server. A failover node cannot be excluded unless it is the last node to be excluded. The server automatically shuts down after exclusion. After including a node, you must be synchronize and restart it.

ASSIGN Designates the given server as the new failover server. The node should not be in the excluded state. The ASSIGN AS FAILOVER clause is a standalone clause that cannot be used with any other ALTER MULTIPLEX SERVER clause.

The coordinator must be running, but you can run the ALTER MULTIPLEX SERVER command from any server in the multiplex. (Sybase recommends that all DDL statements be run on the coordinator.) In all cases except when altering role from reader to writer, the named server is automatically shut down.

Example

Exclude secondary server mpx_writer1:

```
ALTER MULTIPLEX SERVER mpx_writer1 STATUS EXCLUDED
```

Note It is recommended that the target server be shutdown before you exclude it. If you do not, an excluded server will automatically shut down and requires `ALTER MULTIPLEX SERVER` ***`server-name`*** `STATUS INCLUDED` and a synchronize to rejoin the multiplex.

Permissions

Must have DBA or MULTIPLEX ADMIN authority.

CREATE MULTIPLEX SERVER statement

Syntax

CREATE MULTIPLEX SERVER *server-name* **DATABASE** '*dbfile* ' *host-port list* [**ROLE** { **READER** | **WRITER** }] [**STATUS** | { **INCLUDED** | **EXCLUDED** }]

Parameters

host-port-list:

{ HOST ' *hostname*' PORT *port-number* }

Usage

Choose the name of the multiplex server (*server-name*) according to the rules for server startup option -n. See "Starting the database server" in Chapter 1, "Running the Database Server," in the *Utility Guide*.

When creating the initial multiplex server, both coordinator node and secondary node rows are added to SYS.ISYSIQMPXSERVER. The transaction log records this operation as two separate CREATE MULTIPLEX SERVER commands, one for the coordinator node and one for the secondary node.

The SYS.ISYSIQMPXSERVER system table stores the HOST *hostname* PORT *portname* pairs in its connection_info string as *host:port[;host:port…]*.

Note Use multiple host:port pairs if the computer the multiplex server is running on has multiple redundant network cards mapped to different network addresses.

You may specify the clauses DATABASE, *host-port list*, ROLE and STATUS in any order. The default ROLE is READER. The default STATUS is INCLUDED.

When you add a server, the coordinator must be running, but you can run the CREATE MULTIPLEX SERVER command from any server in the multiplex.

This statement is automatically committed.

Example

For examples, see "Adding multiplex servers" on page 31.

Permissions

Must have DBA or MULTIPLEX ADMIN authority.

DROP MULTIPLEX SERVER statement

Syntax

DROP MULTIPLEX SERVER *server-name*

Usage

Deletes a server from the multiplex. Sybase recommends that you shut down each multiplex server before dropping it. This statement is automatically committed.

If not already stopped as recommended, the dropped server automatically shuts down after executing this statement.

Dropping the last secondary server converts the multiplex back to simplex. After dropping the last secondary server within the multiplex, the coordinator automatically shuts down. If required, it needs to be restarted.

Example

```
DROP MULTIPLEX SERVER writer1
```

Permissions

Must have DBA or MULTIPLEX ADMIN authority.

Database options

The following options control multiplex inter-node communications.

MPX_AUTOEXCLUDE_TIMEOUT option

Function

Timeout for auto-excluding a secondary node on the coordinator node. 0 indicates that the nodes will not be auto excluded. This option does not apply to the designated failover node.

Allowed values

0 to 10080 minutes (1 week). Values must be exactly divisible by the heartbeat_frequency setting in minutes. For example, if the heartbeat_frequency setting is 120 (2 minutes), mpx_autoexclude_timeout must be divisible by 2.

Default

60 minutes

MPX_HEARTBEAT_FREQUENCY option

Function

Interval until the heartbeat thread wakes and cleans up the connection pool on the secondary node.

Allowed values

2 seconds to 3600 seconds

Default

60 seconds

MPX_IDLE_CONNECTION_TIMEOUT option

Function

Time after which an unused connection in the connection pool on a secondary node will be closed.

Allowed values	0 sec to no limit
Default	600 seconds

MPX_MAX_CONNECTION_POOL_SIZE option

Function	Maximum number of connections allowed in the connection pool on a secondary node
Allowed values	10 number
Default	1 to 1000

MPX_MAX_UNUSED_POOL_SIZE option

Function	Maximum number of unused connections in the connection pool on a secondary node.
Allowed values	0 sec to no limit
Default	0 to maximum pool size

System tables

The following tables are new or changed to support multiplex functionality.

ISYSIQINFO system table

```
CREATE TABLE SYS.SYSIQINFO (
    last_full_backup TIMESTAMP,
    last_incr_backup TIMESTAMP,
    create_time TIMESTAMP NOT NULL,
    update_time TIMESTAMP NOT NULL,
    file_format_version UNSIGNED INT NOT NULL,
    cat_format_version UNSIGNED INT NOT NULL
    sp_format_version UNSIGNED INT NOT NULL,
    block_size UNSIGNED INT NOT NULL
    chunk_size UNSIGNED INT NOT NULL,
```

```
    file_format_date CHAR(10) NOT NULL,
    dbsig BINARY(136) NOT NULL,
    multiplex_name CHAR(128) NULL,
    last_multiplex_mode TINYINT NULL,
PRIMARY KEY ( create_time ),
)
```

This table indicates the database characteristics as defined when the Sybase IQ database was created using CREATE DATABASE. It always contains only one row. The multiplex_name and last_multiplex_mode columns are new for Sybase IQ 15.2.

last_full_backup Completion time of the most recent full backup.

last_incr_backup Completion time of the most recent incremental backup.

create_time Date and time created.

update_time Date and time of the last update.

file_format_version File format number of files for this database.

cat_format_version Catalog format number for this database.

sp_format_version Stored procedure format number for this database.

block_size Block size specified for the database.

chunk_size Number of blocks per chunk as determined by the block size and page size specified for the database.

file_format_date Date when file format number was last changed.

dbsig Used internally by catalog.

multiplex_name Used internally by catalog.

last_multiplex_mode Mode of the server that last opened the catalog read-write. One of the following values.

- 0 – Single Node.
- 1 – Reader.
- 2 – Coordinator.
- 3 – Writer.

ISYSIQMPXLOGINPOLICYOPTION system table

The ISYSIQMPXLOGINPOLICYOPTION table stores information that allows a user to override the two login policy options (lock and max_connections) for a particular login policy on a per multiplex server basis. See "SYSIQMPXLOGINPOLICYOPTION system view" on page 82.

ISYSIQMPXSERVER system table

ISYSIQMPXSERVER stores membership properties and version status data for a given multiplex node. See "SYSIQMPXSERVER system view" on page 83.

System views

The following views support multiplex functionality.

SYSIQMPXLOGINPOLICYOPTION system view

Presents group information from ISYSIQMPXLOGINPOLICYOPTION in a readable format.

ISYSIQMPXSYSPOLICYLOGIN stores information that allows a user to override the two login policy options (lock and max_connections) for a particular login policy on a per multiplex server basis.

Column name	Column type	Description
login_policy	UNSIGNED BIGINT NOT NULL	The ID number of the login policy.
server_id	UNSIGNED INT NOT NULL	The server name. Must be case insensitive unique.
login_option_name	CHAR(128) NOT NULL	The option name to be overridden: locked or max_connections.

Column name	Column type	Description
login_option_value	LONG VARCHAR NOT NULL	The value of the login policy option. For values and their defaults, see *Reference: Statements and Options*.

Constraints on underlying system table login_policy_id foreign key(ISYSLOGINPOLICY)

server_id foreign key(ISYSMPXSERVER)

SYSIQMPXSERVER system view

Presents a readable version of the table ISYSIQMPXSERVER. The ISYSIQMPXSERVER system table stores membership properties and version status data for the given multiplex node.

Column name	Column type	Description
server_id	UNSIGNED INT NOT NULL	The ID number of the server.
server_name	CHAR(128) NOT NULL	The server name. Must be case insensitive unique.
role	TINYINT NOT NULL	Coordinator, reader, or writer.
status	TINYINT NOT NULL	Excluded or included.
current_version	UNSIGNED BIGINT NULL	Current version ID of the server.
active_version	LONG BINARY NULL	The list of active versions on the server (encoded).
connection_info	LONG VARCHAR NULL	String containing hostname and port pairs delimited by semicolons.
db_path	LONG VARCHAR NOT NULL	Full path to the database file for the server.

Constraints on underlying system table Primary key(server_id)

System procedures

The following system procedures have special syntax or usage notes to support multiplex functionality. Where syntax is not provided, syntax is common across simplex and multiplex servers and is documented in *Reference: Building Blocks, Tables, and Procedures*.

sp_iqcheckdb procedure

Usage

On a multiplex coordinator node, dropleaks mode also detects leaked blocks, duplicate blocks, or extra blocks across the multiplex.

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

sp_iqconnection procedure

Function

Shows information about connections and versions, including which users are using temporary dbspace, which users are keeping versions alive, what the connections are doing inside Sybase IQ, connection status, database version status, and so on.

Syntax

sp_iqconnection [*connhandle*]

Usage

The input parameter *connhandle* is equal to the Number connection property and is the ID number of the connection. The connection_property system function returns the connection ID:

```
SELECT connection_property ( 'Number' )
```

When called with an input parameter of a valid *connhandle*, sp_iqconnection returns the one row for that connection only.

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

Description

sp_iqconnection returns a row for each active connection. The columns ConnHandle, Name, Userid, LastReqTime, ReqType, CommLink, NodeAddr, and LastIdle are the connection properties Number, Name, Userid, LastReqTime, ReqType, CommLink, NodeAddr, and LastIdle respectively, and return the same values as the system function sa_conn_info. The additional columns return connection data from the Sybase IQ side of the Sybase IQ engine. Rows are ordered by ConnCreateTime.

The column MPXServerName stores information related to multiplex Inter-Node Communication (INC), as shown in Table A-1:

Table A-1: MPXServerName column values

Server where run	MPXServerName column content
Simplex server	NULL (All connections are local/user connections.)
Multiplex coordinator	• NULL for local/user connections • Contains value of secondary node's server name (source of connection) for every INC connection (either on-demand or dedicated heartbeat connection.
Multiplex secondary	• NULL for local/user connections • Contains value of coordinator's server name (source of connection).

In Java applications, specify Sybase IQ-specific connection properties from TDS clients in the RemotePWD field. This example, where myconnection becomes the IQ connection name, shows how to specify IQ specific connection parameters:

```
p.put("RemotePWD",",,CON=myconnection");
```

For more details about using the RemotePWD parameter, see *SQL Anywhere Server – Programming*.

Table A-2: sp_iqconnection columns

Column name	Description
ConnHandle	The ID number of the connection.
Name	The name of the server.
Userid	The user ID for the connection.
LastReqTime	The time at which the last request for the specified connection started.
ReqType	A string for the type of the last request.
IQCmdType	The current command executing on the Sybase IQ side, if any. The command type reflects commands defined at the implementation level of the engine. These commands consists of transaction commands, DDL and DML commands for data in the IQ store, internal IQ cursor commands, and special control commands such as OPEN and CLOSE DB, BACKUP, RESTORE, and others.
LastIQCmdTime	The time the last IQ command started or completed on the IQ side of the Sybase IQ engine on this connection.
IQCursors	The number of cursors open in the IQ store on this connection.

Column name	Description
LowestIQCursorState	The IQ cursor state, if any. If multiple cursors exist on the connection, the state displayed is the lowest cursor state of all the cursors; that is, the furthest from completion. Cursor state reflects internal Sybase IQ implementation detail and is subject to change in the future. For this version, cursor states are: NONE, INITIALIZED, PARSED, DESCRIBED, COSTED, PREPARED, EXECUTED, FETCHING, END_OF_DATA, CLOSED and COMPLETED. As suggested by the names, cursor state changes at the end of the operation. A state of PREPARED, for example, indicates that the cursor is executing.
IQthreads	The number of Sybase IQ threads currently assigned to the connection. Some threads may be assigned but idle. This column can help you determine which connections are using the most resources.
TxnID	The transaction ID of the current transaction on the connection. This is the same as the transaction ID displayed in the *.iqmsg* file by the BeginTxn, CmtTxn, and PostCmtTxn messages, as well as the Txn ID Seq logged when the database is opened.
ConnCreateTime	The time the connection was created.
TempTableSpaceKB	The number of kilobytes of IQ temporary store space in use by this connection for data stored in IQ temp tables.
TempWorkSpaceKB	The number of kilobytes of IQ temporary store space in use by this connection for working space such as sorts, hashes, and temporary bitmaps. Space used by bitmaps or other objects that are part of indexes on Sybase IQ temporary tables are reflected in TempTableSpaceKB.
IQConnID	The ten-digit connection ID displayed as part of all messages in the *.iqmsg* file. This is a monotonically increasing integer unique within a server session.
satoiq_count	An internal counter used to display the number of crossings from the SQL Anywhere side to the IQ side of the Sybase IQ engine. This might be occasionally useful in determining connection activity. Result sets are returned in buffers of rows and do not increment satoiq_count or iqtosa_count once per row.
iqtosa_count	An internal counter used to display the number of crossings from the IQ side to the SQL Anywhere side of the Sybase IQ engine. This might be occasionally useful in determining connection activity.
CommLink	The communication link for the connection. This is one of the network protocols supported by Sybase IQ, or is local for a same-machine connection.
NodeAddr	The node for the client in a client/server connection.
LastIdle	The number of ticks between requests.
MPXServerName	If an INC connection, the varchar(128) value contains the name of the multiplex server where the INC connection originates. NULL if not an INC connection.

Example

The following is an example of sp_iqconnection output:

```
ConnHandle      Name    Userid                  LastReqTime   ReqType
===  ===============    ======  ==========================   =======
 9  'IQ_MPX_SERVER_H'   'dbo'   '2008-11-18  13:15:00.035'  'EXEC'
11  'IQ_MPX_SERVER_H'   'dbo'   '2008-11-18  13:15:00.046'  'EXEC'
13  'IQ_MPX_SERVER_H'   'dbo'   '2008-11-18 14:52:55.003'   'EXEC'
15  'IQ_MPX_SERVER_H'   'dbo'   '2008-11-18 14:53:25.005'   'EXEC'
17  'SQL_DBC_49450e8'   'DBA'   '2008-11-18 14:59:45.680'   'OPEN'
44  'Sybase Central 1'  'DBA'   '2008-11-18 14:59:45.023'   'CLOSE'
```

```
            IQCmdType    LastIQCmdTime      IQCursors  LowestIQCursorState
====================    ==============     =========  ===================
   'NONE'               2008-11-18 13:15:00.0       0             'NONE'
   'NONE'               2008-11-18 13:15:00.0       0             'NONE'
   'NONE'               2008-11-18 14:52:55.0       0             'NONE'
   'NONE'               2008-11-18 14:53:25.0       0             'NONE'
'IQUTILITYOPENCURSOR'   2008-11-18 14:59:45.0       0             'NONE'
'NONE'                  2008-11-18 14:43:33.0       0             'NONE'
```

```
IQthreads   TxnID          ConnCreateTime TempTableSpaceKB TempWorkSpaceKB
=========   =====   =====================  =============== ===============
        1       0   2008-11-18 13:14:09.0               0               0
        1       0   2008-11-18 13:14:34.0               0               0
        1       0   2008-11-18 13:14:55.0               0               0
        1       0   2008-11-18 13:15:25.0               0               0
        1   50024   2008-11-18 13:28:08.0               0               0
        1   50545   2008-11-18 14:03:50.0               0               0
```

```
IQconnID satoiq_count iqtosa_count CommLink NodeAdd LastIdle MPXServerName
======== ============ ============ ======== ======= ======== ============
   23198           28           12  'local'      ''     2977  'mpx0631_r1'
   23202           28           12  'local'      ''     1503  'mpx0631_r2'
   23207          127           12  'local'      ''    10000  'mpx0631_w1'
   23212          127           12  'local'      ''    10000  'mpx0631_w2'
   23267          658           66  'TCPIP' '10.18.60.181' 9375
   23443          510           54  'local'      ''        1238
```

sp_iqdbsize procedure

Usage

If run on a multiplex database, the default parameter is main, which returns the size of the shared IQ store.

Permissions — DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

sp_iqdbspace procedure

The sp_iqdbspace procedure displays NA (not available) in the Usage column for the IQ dbspace only when run against a secondary node in a multiplex configuration.

Permissions — DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

sp_iqfile procedure

The sp_iqfile procedure displays NA (not available) in the Usage column for the file of the IQ dbspace only when run against a secondary node in a multiplex configuration.

Permissions — DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

sp_iqindexinfo procedure

Usage — By default in a multiplex database, sp_iqindexinfo displays information about the shared IQ store on a secondary server. If individual tables or indexes are specified, then the store to display is selected automatically.

Permissions — DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

sp_iqmpxincconnpoolinfo procedure

Function — If run on the coordinator node, displays INC connection pool status for every node. If executed on a secondary node, displays INC connection pool status for just the current node.

Syntax — **sp_iqmpxincconnpoolinfo**

Usage

If the procedure is run on the coordinator and a secondary node is not responding or timed out, the result set omits the row for that node, because this data cannot be accessed unless that node is running.

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

Description

The sp_iqmpxincconnpoolinfo procedure returns the following:

Table A-3: sp_iqiqmpxincconnpoolinfo columns

Column name	Data type	Description
server_id	unsigned int	Identifier for the server
server_name	char(128)	Name of the server
current_pool_size	unsigned int	Current size of connection pool
idle_connection_count	unsigned int	Number of idle connections in the pool
connections_in_use	unsigned int	Number of connections in use

Example

This example shows sample output of sp_iqmpxincconnectionpoolinfo:

```
server_id,server_name,current_pool_size,
idle_connection_count,connections_in_use

2,'r2_dbsrv90210',0,0,0

3,'w3_dbsrv90210',0,0,0
```

sp_iqmpxincheartbeatinfo procedure

Function

If run on the coordinator node, displays INC heartbeat status for every node. If executed on a secondary node, displays INC heartbeat status for just the current node.

Syntax

sp_iqmpxincheartbeatinfo

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

Description

The sp_iqmpxincheartbeatinfo procedure returns the following:

Table A-4: sp_iqmpxincheartbeatinfo columns

Column name	Data type	Description	Values
server_id	unsigned int	Identifier for the server	

Column name	Data type	Description	Values
server_name	char(128)	Name of the server	
last_positive_hb	TIMESTAMP	Date/time of last successful heartbeat ping	DD:MM:YYYY: HH:MM:SS
time_not_responding	TIME	Time since last successful heartbeat ping	HH:MM:SS
time_until_timeout	TIME	If a node is not responding, the time left until node is declared offline.	

Examples

This example shows sample output of sp_iqmpxincheartbeatinfo:

```
server_id,server_name,last_positive_hb,
time_not_responding,time_until_timeout
2,'r2_dbsrv90210',2008-11-17
15:48:42.0,00:00:00,00:00:00
3,'w3_dbsrv90210',2008-11-17
15:48:42.0,00:00:00,00:00:00
```

If the elapsed time exceeds 24 hours, Sybase IQ returns sp_iqmpxincheartbeatinfo output like the following:

```
server_id,server_name,last_positive_hb,
time_not_responding,time_until_timeout
2,'r2_mpx_cr_srv',Jan 14 2010
11:57AM,11:59PM,11:59PM
3,'w4_mpx_cr_srv',Jan 14 2010
11:57AM,11:59PM,11:59PM
(2 rows affected)
(return status = 0)
```

A value of 11:59PM in the time_not_responding and time_until_timeout columns means that the time has crossed the 24-hour limit.

sp_iqmpxinfo procedure

Function

If run on the coordinator node, displays complete multiplex configuration info for all nodes. If run on a secondary node, displays complete multiplex configuration info for only that node.

Note Users with RESOURCE authority are not permitted to execute this stored procedure unless granted EXECUTE permission by a user with DBA authority or by a user with PERMS ADMIN authority.

Syntax

sp_iqmpxinfo

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

Description

The sp_iqmpxinfo procedure returns the following:

Table A-5: sp_iqmpxinfo columns

Column name	Data type	Description
server_id	unsigned int	Identifier for the server for which information is displayed
server_name	char(128)	Name of the server
connection_info	long varchar	A formatted string containing the host/port portion of the connection string used for TCPIP connections between multiplex servers.
db_path	long varchar	Full database path
role	char(16)	'coordinator' | 'writer' | 'reader'
status	char(8)	'included' | 'excluded'
mpx_mode	char(16)	'single' | 'coordinator' | 'writer' | 'reader' | 'writer' | 'unknown'
inc_state	char(16)	'active' | 'not responding' | 'timed out'
coordinator_failover	char(128)	Name of the failover server
current_version	unsigned bigint	Decimal-formatted version ID
active_versions	long_varchar	Comma-separated list of decimal formatted version IDs.

Example

This example shows sample output of sp_iqmpxinfo:

```
server_id,server_name,connection_info,db_path,
role,status,mpx_mode,inc_state,
coordinator_failover,current_version,
active_versions

1,'dbsrv10261_c1','host=(fe80::214:4fff:fe45:
e6f2%2):11075,(fd77:55d:59d9:329:214:4fff:fe45:
e6f2%2):11075,10.18.41.194:11075',
'/sunx5prod/mpxsim/c1/mpx_simdb.db',
'coordinator','included','coordinator',
'N/A','r2_dbsrv10261',0,

2,'r2_dbsrv10261','host=localhost:7733',
'/sunx5prod/mpxsim/r2/mpx_simdb.db',
'reader','included','reader','active',
'r2_dbsrv10261',0,

3,'w3_dbsrv10261','host=localhost:5507',
'/sunx5prod/mpxsim/w3/mpx_simdb.db','writer',
'included','writer','active','r2_dbsrv10261',0,
```

sp_iqmpxvalidate procedure

Function

Checks multiplex configuration for inconsistencies.

Syntax

call dbo.sp_iqmpxvalidate('show_msgs')

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission to run the stored procedure.

Description

Executes multiple checks on tables SYS.SYSIQDBFILE and other multiplex events and stored procedures. May run on any server. Returns a severity result to the caller; values are:

Value	Description
0	No errors detected
1	Dynamic state is not as expected.
2	Nonfatal configuration error; for example, multiplex operation impaired
3	Fatal configuration problem; for example, one or more servers might not start

If called interactively, also returns a table of the errors found, if any, unless the calling parameter is not 'Y'.

Each error indicates its severity. If there are no errors, the procedure returns `No errors detected`.

sp_iqmpxversioninfo procedure

Function

Shows the current version information for this server. Information includes server type (write server, query server, single-node mode) and synchronization status.

Syntax

sp_iqmpxversioninfo( )

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission to run the stored procedure.

Description

Table 5-1: sp_iqmpxversioninfo columns returned

Column	Data type	Description
CatalogID	unsigned bigint	Catalog version on this server
VersionID	unsigned bigint	Latest version available on this server
OAVID	unsigned bigint	Oldest active version on this server
ServerType	char(1)	Type of server: “C” (Coordinator), “W” (Write Server) or “Q” (Query Server)
CatalogSync	char(1)	Catalog synchronization: “T” (synchronized) or “F” (not synchronized)
WCatalogID	unsigned bigint	Catalog version on the write server
WVersionID	unsigned bigint	Latest version available on the write server

sp_iqspaceinfo procedure

Usage

If run on a multiplex database, the default parameter is main, which returns the size of the shared IQ store.

If you do not supply a parameter, this procedure returns no results unless you have at least one user-created object, such as a table.

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

sp_iqspaceused procedure

Usage

If run on a multiplex database, this procedure applies to the server on which it runs.

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

sp_iqstatus procedure

Usage

On a secondary server in a multiplex, this procedure also lists information about the shared IQ store and IQ temporary store.

If sp_iqstatus shows a high percentage of main blocks in use on a multiplex server, run sp_iqversionuse to find out which versions are being used and the amount of space that can be recovered by releasing versions. See "sp_iqversionuse procedure" on page 97.

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

sp_iqtransaction procedure

Function

Shows information about transactions and versions.

Syntax

sp_iqtransaction

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

Description

sp_iqtransaction returns a row for each transaction control block in the Sybase IQ transaction manager. The columns Name, Userid, and ConnHandle are the connection properties Name, Userid, and Number, respectively. Rows are ordered by TxnID.

The sp_iqtransaction output does not contain rows for connections that do not have a transaction started. To see all connections, use sp_iqconnection.

Note Although you can use sp_iqtransaction to identify users who are blocking other users from writing to a table, sp_iqlocks is a better choice for this purpose.

Table A-6: sp_iqtransaction columns

Column name	Description
Name	The name of the server.
Userid	The user ID for the connection.
TxnID	The transaction ID of this transaction control block. The transaction ID is assigned during begin transaction. This is the same as the transaction ID displayed in the *.iqmsg* file by the BeginTxn, CmtTxn and PostCmtTxn messages as well as the Txn ID Seq logged when the database is opened.
CmtID	The ID assigned by the transaction manager when the transaction commits. It is zero for active transactions.
VersionID	In simplex databases, the VersionID is the same as the TxnID. For the multiplex coordinator, the VersionID is the same as the TxnID of the active transaction and VersionID is the same as the CmtID of a committed transaction. In multiplex secondary servers, the VersionID is the CmtID of the transaction that created the database version on the multiplex coordinator. It is used internally by the Sybase IQ in-memory catalog and the IQ transaction manager to uniquely identify a database version to all nodes within a multiplex database.
State	The state of the transaction control block. This variable reflects internal Sybase IQ implementation detail and is subject to change in the future. At the time of this writing, transaction states are NONE, ACTIVE, ROLLING_BACK, ROLLED_BACK, COMMITTING, COMMITTED, and APPLIED.
ConnHandle	The ID number of the connection.
IQConnID	The ten-digit connection ID displayed as part of all messages in the *.iqmsg* file. This is a monotonically increasing integer unique within a server session.
MainTableKBCr	The number of kilobytes of IQ store space created by this transaction.
MainTableKBDr	The number of kilobytes of IQ store space dropped by this transaction, but which persist on disk in the store because the space is visible in other database versions or other savepoints of this transaction.
TempTableKBCr	The number of kilobytes of IQ temporary store space created by this transaction for storage of IQ temporary table data.
TempTableKBDr	The number of kilobytes of IQ temporary table space dropped by this transaction, but which persist on disk in the IQ temporary store because the space is visible to IQ cursors or is owned by other savepoints of this transaction.
TempWorkSpaceKB	For ACTIVE transactions, this is a snapshot of the work space in use at this instant by this transaction, such as sorts, hashes, and temporary bitmaps. The number varies depending on when you run sp_iqtransaction. For example, the query engine might create 60MB in the temporary cache but release most of it quickly, even though query processing continues. If you run sp_iqtransaction after the query finishes, this column shows a much smaller number. When the transaction is no longer active, this column is zero. For ACTIVE transactions, this column is the same as the TempWorkSpaceKB column of sp_iqconnection.

Column name	Description
TxnCreateTime	The time the transaction began. All Sybase IQ transactions begin implicitly as soon as an active connection is established or when the previous transaction commits or rolls back.
CursorCount	The number of open Sybase IQ cursors that reference this transaction control block. If the transaction is ACTIVE, it indicates the number of open cursors created within the transaction. If the transaction is COMMITTED, it indicates the number of HOLD cursors that reference a database version owned by this transaction control block.
SpCount	The number of savepoint structures that exist within the transaction control block. Savepoints may be created and released implicitly. Therefore, this number does not indicate the number of user-created savepoints within the transaction.
SpNumber	The active savepoint number of the transaction. This is an implementation detail and might not reflect a user-created savepoint.
MPXServerName	The value indicates if an active transaction is from an inter-node communication (INC) connection. If from INC connection, the value is the name of the multiplex server where the transaction originates. NULL if not from an INC connection. Always NULL if the transaction is not active.
GlobalTxnID	The value indicates the global transaction ID associated with the current transaction. Zero if there is no associated global transaction.

Example

Here is an example of sp_iqtransaction output:

```
Name,Userid,TxnID,CmtID,VersionID,State,ConnHandle,IQConnID,
MainTableKBCr,MainTableKBDr,TempTableKBCr,TempTableKBDr,
TempWorkSpaceKB,TxnCreateTime,CursorCount,SpCount,SpNumber,
MPXServerName,GlobalTxnID

'IQ_MPX_SERVER_H','dbo',49878,49881,49881,'COMMITTED',9,23198,152,
152,0,0,0,'2008-11-18 13:15:00.015',0,0,0,,0

'IQ_MPX_SERVER_H','dbo',49880,49882,49882,'COMMITTED',13,23207,152,
152,0,0,0,'2008-11-18 13:15:00.016',0,0,0,,0

'IQ_MPX_SERVER_H','dbo',49884,49885,49885,'COMMITTED',11,23202,152,
152,0,0,0,'2008-11-18 13:15:00.038',0,0,0,,0

'IQ_MPX_SERVER_H','dbo',49909,49910,49910,'COMMITTED',15,23212,152,
152,0,0,0,'2008-11-18 13:16:00.016',0,0,0,,0

'SQL_DBC_49450e8','DBA',50024,0,50024,'ACTIVE',17,23267,0,0,0,
0,0,'2008-11-18 13:28:23.358',0,2,261,,0

'Sybase Central 1','DBA',50545,0,50545,'ACTIVE',44,23443,0,0,0,
0,0,'2008-11-18 14:04:53.256',0,1,0,,0
```

sp_iqversionuse procedure

Description

The procedure produces a row for each user of a version. Run sp_iqversionuse first on the coordinator to determine which versions should be released and the amount of space in KB to be released when the version is no longer in use. Connection IDs are displayed in the IQConn column for users connected to the coordinator. Version usage due to secondary servers is displayed as the secondary server name with connection ID 0.

Run sp_iqversionuse on multiplex secondary servers to determine individual connections to secondary servers. Users from other servers are not displayed on a secondary server.

Example

In this example, the oldest version 42648 is in use by connection 108 on the coordinator (*mpxw*). Committing or rolling back the transaction on connection 108 releases 7.9MB of space. Version 42686 is in use by secondary server (*mpxq*) according to output from the coordinator. Using the secondary server output, the actual connection is connection 31. The actual amount of space returned from releasing version 42686 depends on whether 42648 is released first.

WasReported is 0 for versions 42715 and 42728 on the coordinator because these are new versions that have not yet been replicated by SQL Remote. Since version 42728 does not appear on the secondary server output, it has not yet been used by the secondary server.

The following output is returned when sp_iqversionuse executes on the coordinator *mpxw*:

```
call dbo.sp_iqversionuse
```

VersionID	Server	IQConn	WasReported	MinKBRelease	MaxKBRelease
42648	'mpxw'	108	1	7920	7920
42686	'mpxq'	0	1	7920	304
42702	'mpxq'	0	1	0	688
42715	‘mpxq'	0	0	0	688
42728	'mpxq'	0	0	0	688

The following output is returned when sp_iqversionuse executes on the secondary server (*mpxq*):

```
call dbo.sp_iqversionuse
```

VersionID	Server	IQConn	WasReported	MinKBRelease	MaxKBRelease
42686	'mpxq'	31	1	0	0

VersionID	Server	IQConn	WasReported	MinKBRelease	MaxKBRelease
42715	'mpxq'	00	1	0	0

Permissions

DBA authority required. Users without DBA authority must be granted EXECUTE permission in order to run the stored procedure.

Startup and database administration utilities

The following utilities have multiplex syntax or restrictions. For syntax common across multiplex and simplex servers, see the *Utility Guide*.

Backup utility (dbbackup)

The dbbackup utility truncates the database name to 70 characters and creates a target file with a truncated name. Sybase IQ uses dbbackup when synchronizing secondary servers. Due to the dbbackup restrictions, database names must be less than 70 characters.

Server startup utility (start_iq)

The server startup switches in Table A-7 are used to configure Sybase IQ multiplex servers at startup.

Table A-7: Server startup switches

Startup switch	Values	Description
-iqmpx_failover	1	Initiates multiplex coordinator failover to establish the designated failover Secondary node as the new coordinator. Starting the coordinator with this option has no effect.
-iqmpx_ov	1	Performs multiplex configuration override for the current node. Used to change node properties during startup in the event that a node's location or other property has changed.

Startup switch	Values	Description
-iqmpx_sn	1	Runs the current node in multiplex in single node mode. This mode is used exclusively for fixing problems with the multiplex configuration and should be used with extreme caution. Requires all other nodes in the multiplex to be shut down. Recommended only for use on the coordinator node.
-iqmpx_reclaimwriterfreelist	server name	This option applies only while restarting a coordinator node. The coordinator will forcefully reclaim the free-list of the writer node identified by server-name. This switch is only needed in the event that a Writer fails and cannot be restarted.
-iqmsgnum *num*	0-64 (inclusive)	Specifies the number of archives of the old message log maintained by the server. Default value is 0, which means that messages are wrapped in the main message log file. Takes effect only if -iqmsgsz or the IQMsgMaxSize server property is non-zero. The IQMsgNumFiles server property corresponds to -iqmsgnum and takes precedence over the value of -iqmsgnum.
-iqmsgsz *size*	integers 0-2047 (inclusive) in megabytes.	Limits the maximum size of the message log. The default value is 0, which specifies no limit on the size of the message file.

Index

E

F

H

I

J

L

M

N

O

P

R

S

T

U

V

W