SYBASE®
An SAP® Company

Using Sybase IQ Multiplex

Sybase IQ 15.4

DOCUMENT ID: DC00802-01-1540-04
LAST REVISED: August 2013

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

Contents

Multiplex Overview

Administering Sybase® IQ multiplex servers and transactions requires basic knowledge of data storage and communication in the multiplex architecture.

Audience

This guide is for users who need Sybase® IQ multiplex capability to manage large query loads across multiple nodes.

Use the appendix in combination with the other Sybase IQ reference manuals to understand syntax, system objects, and utilities that support multiplex operations.

Introduction to Sybase IQ Multiplex

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 benefits:

- 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.

Sybase IQ Multiplex Architecture

Sybase IQ multiplexes have a hybrid cluster architecture that involves both shared and local storage.

Shared storage is used for permanent IQ data and shared temporary data for distributed query processing. Local storage is used for catalog metadata, temporary data, and transaction logs.

Each server has its own temporary store and catalog store.

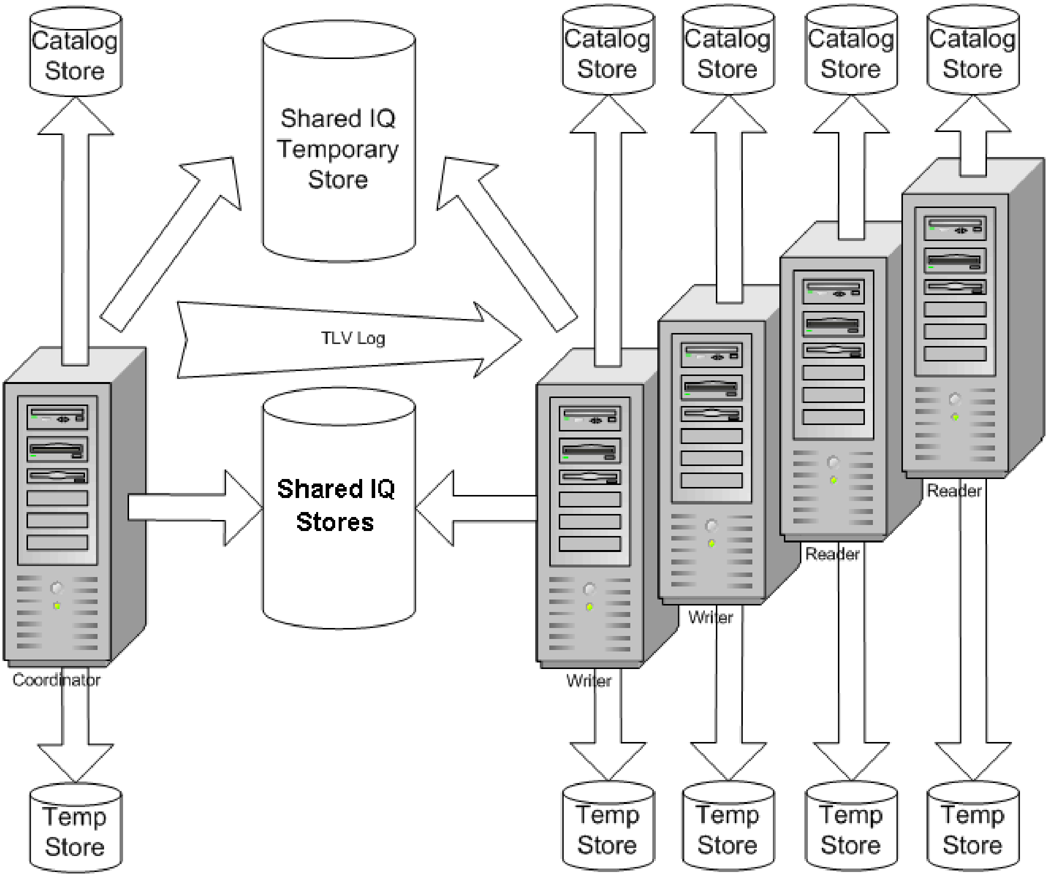
The shared IQ store and shared temporary store are common to all servers.

Multiplex Architecture Diagram

Sybase IQ multiplex nodes may have different roles with different capabilities, unlike other database cluster architectures, which usually follow either a “shared everything” or “shared nothing” architecture.

The configuration can be described as an "asymmetrical cluster."

Figure 1: IQ Multiplex Architecture



Data Storage

Understand the distinction between data managed solely by Sybase IQ and data that is managed by underlying SQL Anywhere® software.

See *System Administration Guide: Volume 1 > Overview of Sybase IQ System Administration > Data Storage in Sybase IQ* and *System Administration Guide: Volume 1 > Overview of Sybase IQ System Administration > SQL Anywhere and Sybase IQ*.

Table 1. Dbspace administration

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

IQ_SYSTEM_MAIN, IQ_SHARED_TEMP, and IQ user main dbspaces are shared by all multiplex servers, and all servers need access to the same physical file. See *Files on Shared Disk Arrays* for address formats.

Data managed by SQL Anywhere is not shared. Each node requires a separate 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 guideline: To address dbspaces managed by IQ, use absolute, not relative paths. To address catalog dbspaces managed by SA, use relative paths, not absolute paths.

See also

- *UNIX or Linux Shared Array Path Definitions* on page 6
- *Windows Shared Disk Array Path Definitions* on page 7

Multiplex Configuration

Sybase IQ multiplex configuration requires static and dynamic components.

- *Static configuration* includes multiplex member nodes and their properties.
- *Dynamic configuration* includes runtime process and connection status and table version usage.

  Sybase IQ stores IQ table metadata in the table's dbspace and the table version (TLV) log, which is a shared entity on the main store. The multiplex synchronizes stored table versions when an unavailable node comes back up.

Coordinator Node

Each multiplex configuration requires a *coordinator node*.

When you convert an existing simplex server to multiplex, it becomes the 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.

Multiple Writers for Parallel Loads

Multiplexes include a single coordinator and permit multiple writers and readers.

This architecture:

- Supports read-write transactions from coordinator and write servers in the multiplex
- Allows multiple servers to perform loads to different tables at the same time

This speeds up loads by employing the local CPU, memory, and disk resources of multiple computers.

Parallel loads must be on different tables. You cannot perform loads on the same table at the same time.

Multiple writer configurations are expected to increase the performance of parallel table loads nearly linearly if the table loads are run on separate write servers. For a sizable load, the overhead involved in inter-node communication is negligible. For small read-write operations, such as single row inserts or updates, however, the inter-node communication is significant compared to the DML operation itself, so Sybase recommends that you avoid running heavy workloads of small read-write updates on writer nodes.

Small batches mean loads that take less than a minute. Due to constant overhead, small transactions take longer on the writer. For long running transactions, the fixed overhead cost is a smaller percentage.

See also

- *MPX_MAX_CONNECTION_POOL_SIZE Option* on page 104
- *MPX_MAX_UNUSED_POOL_SIZE Option* on page 105

Files on Shared Disk Arrays

Sybase IQ multiplex functionality requires using the same file paths to access files in dbspaces in the shared stores from all nodes.

All files in all dbspaces in the shared store, namely all files in IQ_SYSTEM_MAIN and shared dbspaces, must be accessible in exactly the same way from all nodes. Files in the shared temporary store should be accessible from all nodes in the multiplex.

In Sybase IQ12.7, 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 no longer supports aliases for shared IQ stores. 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.

See also

- *UNIX or Linux Shared Array Path Definitions* on page 6
- *Windows Shared Disk Array Path Definitions* on page 7

UNIX or Linux Shared Array Path Definitions

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

Use absolute paths to shared stores. For example:

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

Use soft links to shared stores. For example:

```
store/mainstore/userdb1store/userdb2
```

where each of these soft links point to raw devices as shown in the absolute path example.

See also

- *Data Storage* on page 3
- *Files on Shared Disk Arrays* on page 6
- *Main Store Requirements* on page 11

Windows Shared Disk Array Path Definitions

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

Use the Disk Physical number. For example:

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

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.

See also

- *Data Storage* on page 3
- *Files on Shared Disk Arrays* on page 6
- *Main Store Requirements* on page 11

Communication Infrastructure

The servers that participate in the multiplex share messages and data using two frameworks.

Inter-node Communication (INC)

INC provides transactional communication between coordinator and secondary nodes.

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 executing DDL and read-write DML statements. 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 using INC.

The INC link consists of heartbeat and pooled connections.

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`.

See also

- *Internode Communication State* on page 73

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`.

See also

- *MPX_MAX_CONNECTION_POOL_SIZE Option* on page 104
- *MPX_MAX_UNUSED_POOL_SIZE Option* on page 105
- *Internode Communication State* on page 73

Multiplex Interprocess Communication (MIPC)

MIPC connects all multiplex nodes to support distributed query processing and high availability.

As multiplex servers start, they establish MIPC connections.

MIPC is a fully meshed communication framework that runs on both public and private interconnection configurations. Public interconnection configuration is mandatory while private configuration is optional.

Private high-speed interconnection configurations are for distributed query processing. Currently, private interconnection configurations are restricted to physical networks supporting the TCP/IP protocol.

If no private interconnection configuration is provided, MIPC uses the legacy public interconnection configuration specified in the system table column SYSIQMPXSERVER.conn_info. This set of host/port addresses is shared between external user connections, INC connections, and internal MIPC connections.

Redundant networks provide a more robust cluster. If both private and public interconnection configurations exist, MIPC uses the private connections if available, and reroutes traffic to the public connections if the private network fails. The multiplex monitors the messages between nodes to detect failures.

Possible interconnection link failures include:

- A physical failure, such as a cable that is disconnected or broken
- A power supply failure, such as a piece of network infrastructure equipment
- A software failure within the networking stack

For recommendations on planning network configurations for distributed query processing, see the *Installation and Configuration Guide*.

Changes in Sybase IQ Multiplex Functionality

Sybase IQ 15.4 extends the power of external user defined table functions.

Sybase IQ 15.4 includes these features:

- External Environment support for UDFs – Developers can use the Java environment to create stored procedures. Developers will be able to use standard Java APIs for processing arguments and returning values (result set) back to the server. Sybase IQ will extend the multiplex facilities for exploiting large clusters of computers from within the umbrella of SQL while making it possible for end users to write computational logic in languages other than SQL.
- Support for table UDFs – Table functions can facilitate set-based customized computing by allowing the end user to implement logic that processes input data and produces relational sets as output which can then be consumed in a SQL query as if it was a table expression.
- Table parameterized functions – Sybase IQ will extend the traditional table function approach to handle arbitrary rowsets as input and to allow user-specified partitioning and ordering requirements as arguments to table function invocations. Sybase IQ will extend the multiplex facilities for exploiting large clusters of computers from within the umbrella of SQL while making it possible for end users to write computational logic in languages other than SQL.

Multiplex Creation

Convert a single Sybase IQ server into a multiplex.

1. Set up the main store and hardware to meet requirements.
2. Convert databases to multiplex.

Multiplex Storage Requirements

Create multiplex stores on the appropriate device.

Stores	Supported Devices
IQ shared stores (IQ_SYSTEM_MAIN, IQ_SHARED_TEMP, user dbspaces)	Raw device required
IQ non-shared store (IQ_SYSTEM_TEMP)	Both raw devices and O/S files supported
Catalog and transaction log	O/S files required

Main Store Requirements

Make sure that your main store meets prerequisites for multiplex operation.

Main store file paths must be absolute or relative in such a way that they are accessible to all servers in the multiplex.

- See *Sybase IQ System Administration Guide: Volume 1 > Database Object Management > Database Definition* to create IQ databases.
- See *Installation and Configuration Guide > Database Upgrades* to migrate your database from an earlier release of Sybase IQ.

See also

- *UNIX or Linux Shared Array Path Definitions* on page 6
- *Windows Shared Disk Array Path Definitions* on page 7

Hardware Requirements

A Sybase IQ multiplex requires shared storage for data on IQ_SYSTEM_MAIN, IQ_SHARED_TEMP, and user dbspaces.

All machines running servers participating in the multiplex must have the current version of Sybase IQ installed. For upgrade instructions, see *Installation and Configuration Guide > Planning Your Installation > Upgrading From an Earlier Version*.

When you convert an existing simplex server to multiplex, it becomes the coordinator. For secondary servers, set up a computer with access to the shared IQ store using paths identical to those used by the coordinator.

Sybase IQ does not support:

- Multiplexes of Sybase IQ servers at different release levels.
- Heterogeneous multiplexes (UNIX and Windows servers in a mixed multiplex). Coordinator and secondary servers must be on the same operating system and hardware platform.
- Multiplex server instances on virtual machines.

Java External Environment in a Multiplex

Before you can use Java external environment UDFs in a multiplex configuration, install the Java class file or JAR files on each node of the multiplex that requires the UDF.

Use Sybase Control Center, Sybase Central, or the Interactive SQL **INSTALL JAVA** statement to install the Java class file and JAR.

Multiplex Planning Worksheet

Before creating a multiplex, check that the paths of proposed objects meet requirements.

Database paths on all platforms, whether raw device or operating system files, are limited to 128 bytes. Sybase IQ supports:

- Raw devices for IQ shared stores (IQ_SYSTEM_MAIN, IQ_SHARED_TEMP, and user dbspaces)
- Both raw devices and operating system files for non-shared IQ stores (IQ_SYSTEM_TEMP)
- Operating system files only for the catalog and transaction log

Sybase Central dialogs and SQL statements for creating a multiplex require some or all of the following values.

Table 2. Multiplex database requirements

Dialog item	Type/length	Notes	Value
Host name	CHAR 128	Name of the machine where the database engine will run.	

Dialog item	Type/length	Notes	Value
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 Databases to Multiplex

Use Sybase Central, Sybase® Control Center, or Interactive SQL to create multiplex servers.

To change a simplex database to a multiplex, 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.

Converting Databases to Multiplex with Interactive SQL

To change a simplex database to a multiplex, connect to the simplex database and use CREATE MULTIPLEX SERVER.

Prerequisites

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.

Task

1. Make sure that your system meets hardware prerequisites. The main store dbfiles must be visible to secondary nodes.
2. Start a Sybase IQ server and connect to a database with a shareable main store. This server becomes the coordinator of the multiplex:

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

3. Connect using Interactive SQL:

   ```
   dbisql
   ```

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

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

   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.

   ```
   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. 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=2957}"
    ```

12. Add the temporary dbfile in Interactive SQL using the **ALTER DBSPACE** statement.

    (Secondary servers do not allow **CREATE DBSPACE**.)

    ```
    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 *Reference: Statements and Options > SQL Statements > ALTER DBSPACE Statement*.

Converting Databases to Multiplex with Sybase Central

Use the Create Server Wizard to enable a simplex server for multiplex.

Prerequisites

DBA authority is required to perform the following steps.

Task

1. Make sure that your system meets hardware prerequisites. The main store dbfiles must be visible to secondary nodes.
2. Start Sybase Central using the method appropriate for your platform.
3. If on UNIX, check that the IQ Agent is running:
   ```
   stop_iq -agent
   ```
4. Use the Start Server Wizard to start a Sybase IQ server.
5. Connect to the IQ server.
6. Right-click the server name and choose Convert to Multiplex (Alt+M). If you lack DBA authority, the Convert to Multiplex menu item is disabled.
7. Follow the instructions on the Create Server Wizard screens.

Converting Databases to Multiplex with Sybase Control Center

If you prefer a web-based tool for converting databases to multiplex, use Sybase Control Center.

See the Sybase Control Center for Sybase IQ online help in SCC or at *http://sybooks.sybase.com/nav/summary.do?prod=10680*.

Multiplex Database Files

Each server in the multiplex uses several files to store data, metadata, and configuration information.

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

Table 3. 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. The IQ_SYSTEM_TEMP store can be on either a raw device or a system file.
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	

See also

- *Administrative Shell Scripts* on page 30

Multiplex Server Administration

Use Sybase Central, Sybase® Control Center, or a combination of command line utilities and Interactive SQL to manage multiplex servers.

Multiplex Administration with Interactive SQL

If you prefer the flexibility of scripts or are familiar with Interactive SQL from other products, use Interactive SQL.

Interactive SQL is a command line interface that lets you enter SQL statements individually or run them in command scripts to query, modify, or load Sybase servers, including Sybase IQ.

See also

- *CREATE MULTIPLEX SERVER Statement* on page 99
- *Adding Multiplex Servers with Interactive SQL* on page 18

Starting Multiplex Servers from the Command Line

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.

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 the total number of expected simultaneous requests plus 5 (for internal events and connections.) For a complete list of parameters, see *Utility Guide > start_iq Database Server Startup Utility > start_iq Server Options*.

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
```

Checking Server Status in Interactive SQL

Check the server status to determine if any servers are not responding or not included in the multiplex, and to see which server will be the new coordinator in the event of a failover.

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**.

See also

- *sp_iqmpxinfo Procedure* on page 125

Adding Multiplex Servers with Interactive SQL

Follow these steps to add multiplex servers from Interactive SQL.

1. Check server status. The coordinator must be running in order to add secondary servers.
2. 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 *Multiplex Reference > SQL Statements > CREATE MULTIPLEX SERVER Statement*.

   This command creates the definition for a secondary writer server `mpxnode_w2`.
3. Follow steps 7 through 12 in *Multiplex Creation > Converting Databases to Multiplex > Converting Databases to Multiplex with Interactive SQL*.

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 *Multiplex Server Administration > Multiplex Server Administration with Interactive SQL > Converting databases to multiplex with Interactive SQL*, steps 11 and 12. After adding this file, run **sp_iqmpxvalidate** to make sure that no errors are reported on this server.

See also

- *Multiplex Administration with Interactive SQL* on page 17
- *CREATE MULTIPLEX SERVER Statement* on page 99

Renaming Multiplex Servers with Interactive SQL

You can use Interactive SQL to change the name of a multiplex server.

Prerequisites

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. Except when altering role from reader to writer, the named server automatically shuts down once the change is committed.

Task

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

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

This command renames and shuts down the named server.

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

See also

- *ALTER MULTIPLEX SERVER Statement* on page 96

Requirements for Altering Multiplex Servers

Familiarize yourself with requirements and results before you alter a multiplex server.

The coordinator must be running.

Run the **ALTER MULTIPLEX SERVER** command from the coordinator.

The named server automatically shuts down once the change is committed, except when altering role from reader to writer.

Changing the Database File Path with Interactive SQL

If you have to move your database, for example to a disk with more space, you can change the file path with Interactive SQL.
Start the server, connect to it, and issue a command in the following format:

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

This command also shuts down the named server.

Changing Server Roles with Interactive SQL

You can use Interactive SQL to change a reader to a write server.
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 and Port with Interactive SQL

You can use Interactive SQL to change the host and port information for a multiplex server. 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 with Interactive SQL

If a multiplex secondary server will be shut down for an extended period of time, exclude that server from the multiplex.

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 with Interactive SQL

You must designate a failover node to continue the coordinator role in the event that the current coordinator is unable to continue.

You must designate a failover node to continue the coordinator role in the event that the current coordinator is unable to continue.

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

   ```
   ALTER MULTIPLEX SERVER server-name ASSIGN AS FAILOVER SERVER
   ```
2. The designated failover node defaults to the first multiplex server added to the multiplex.

See also

- *Designated Failover Node* on page 37
- *sp_iqmpxinfo Procedure* on page 125

Requirements for Dropping Multiplex Servers

Familiarize yourself with requirements and results before you drop a multiplex server.

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.

See also

- *DROP MULTIPLEX SERVER Statement* on page 101
- *Dropping Multiplex Servers with Interactive SQL* on page 21

Dropping Multiplex Servers with Interactive SQL

Dropping a secondary server removes it from the multiplex configuration.

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

```
DROP MULTIPLEX SERVER server-name
```

See also

- *Requirements for Dropping Multiplex Servers* on page 20
- *Coordinator Failure* on page 37
- *DROP MULTIPLEX SERVER Statement* on page 101

Synchronizing Servers from the Command Line

Synchronizing copies the coordinator's version of the database catalog to secondary servers.

Prerequisites

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**.

Task

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.

See also

- *Requirements for Failover* on page 38
- *Updates on IQ_SYSTEM_MAIN* on page 47

Counting User Connections

To avoid exceeding the maximum number of connections, periodically check the current number of user connections.

The **DB_PROPERTY** 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.

From the coordinator, run the system function **DB_PROPERTY**.

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

Multiplex Administration with Sybase Central

If you are a new user, or more comfortable with graphical interfaces, use Sybase Central.

Sybase Central is a graphical tool to administer and monitor many Sybase server products, including Sybase IQ.

Starting Multiplex Servers with Sybase Central

To start IQ multiplex servers in scripts, you may use command line parameters.

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.

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.

Checking Server Status in Sybase Central

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

1. Open the multiplex folder.
2. Select the Servers tab to view server status.

Table 4. 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.

See also

- *sp_iqmpxinfo Procedure* on page 125

The Sybase IQ Agent

The Sybase IQ Agent enables Sybase Central to perform functions required to administer Sybase IQ servers.

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

See also

- *Starting Sybase Central on UNIX* on page 29
- *Running the Sybase IQ Agent on UNIX* on page 28
- *Starting Sybase Central on Windows* on page 29

The Sybase IQ Agent Log Location

The path to the Sybase IQ Agent log is platform specific.

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`.

Moving the Sybase IQ Agent Log on UNIX

To generate the IQ 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
```

Moving the Sybase IQ Agent Log on Windows

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

a. Use Control Panel > System > Advanced Properties > Environment Variables.
b. Click New under System variables.
c. Type IQLOGDIR15 for *Variable* and the full path of the directory for *Value*.
d. Click OK.

Viewing the Sybase IQ Agent Log in Sybase Central

If you experience any problems with Sybase Central, check the agent log file for errors and messages.
Right-click the server name and choose Agent > View Agent Log

Troubleshooting Sybase IQ Agent Startup on Windows

If you receive an error message that Sybase IQ Agent is not running, start the agent.

1. Open the Services utility under Control Panel.
2. If Sybase IQ Agent does not have the status “Started,” click Start.
3. Restart Sybase Central after starting the agent.

Troubleshooting Sybase IQ Agent Startup on UNIX

To check if the agent is running on UNIX, run the **stop_iq** utility.

1. Enter the **stop_iq** command with the **-agent** argument:

   ```
   stop_iq -agent
   ```

   ```
   Checking system ...

   The following 1 agent(s) are owned by 'fiona'
   ## Owner     PID     Started  CPU Time Additional Information
   -- --------- ------- -------- -------- -----------------------
   - 1: fiona 6669 Sep.01 5:11 PORT:1100 java
   -Diq.agent=/work/sybase15/
   IQ-15_3/java/IQAgent15.jar
   -Di q.agent_log=/

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

   This output shows that user `fiona` is running the agent.
2. To stop the agent, type “Y” for yes at the prompt:`Y`
3. Should you 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.

Setting permissions for the Sybase IQ Agent on Windows

On Windows, you must change the owner of the Sybase IQ Agent because the System account starts the agent by default.

The user who starts the agent becomes the creator and owner of multiplex databases and server directories, and must have write privileges on all of the servers in the multiplex.

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

Permissions for the Sybase IQ Agent on UNIX

You must start the Sybase IQ Agent from an account with appropriate privileges.

Make sure that the `S99SybaseIQAgent15` script is owned by the same UID that is used to run all the servers. Do not use the root user account.

Overriding the Sybase IQ Agent Port Number on Windows

Sybase IQ Agents on a given host require unique port numbers. Use the Service Manager to override the default Sybase IQ Agent Port Number on Windows.

1. Select the name Sybase IQ Agent 15.
2. Choose Action > Properties from the menu bar.

   Click Stop under Service status to stop the agent.

   In the Start Parameters text box, type `-port nnnn` where nnnn is the port number.

   Click Start to restart the agent.

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

Overriding the Sybase IQ Agent Port Number on UNIX

Sybase IQ Agents on a given host require unique port numbers.

Specify the -DIQPORT parameter on the Sybase Central startup command, for example:

```
scjview -DIQPORT=3356
```

Changing the port number lets you run multiple current version Sybase IQ Agents on a given host, or to run agents for Sybase IQ 12.7, 15.0, 15.1, 15.2, 15.3, and 15.4 on the same host.

Uninstalling and Reinstalling Sybase IQ Agent Windows Service

If you experience problems with the Sybase IQ Agent on Windows, try uninstalling and reinstalling the Sybase IQ Agent Windows Service.

1. To uninstall the Sybase IQ Agent 15 Windows Service:

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

2. To reinstall the Sybase IQ Agent 15 Windows Service:

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

Disabling Remote File Browsing on Windows

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

1. In a DOS shell, set the environment variable **IQAGENTNOBROWSE** to any value:

   ```
   SET IQAGENTNOBROWSE=1
   ```

   ```
   SET IQPORT=2525
   ```

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

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

Disabling Remote File Browsing on the Sybase IQ Agent on UNIX

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

To disable browsing on UNIX:

Start the agent with the parameter **-nobrowse**:

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

Setting IQAGENTNOBROWSE

The Sybase IQ Agent supports remote file browsing by default.

Set the IQAGENTNOBROWSE environment variable to disable browsing if you have security concerns. This method is an alternative to using the **-nobrowse** startup parameter.

Configuring the IQ Agent on Windows

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

Configuring the Sybase IQ Agent to Start Automatically on UNIX

You can configure the Sybase IQ agent to start automatically whenever you reboot your system.

To avoid restarting the Sybase IQ Agent repeatedly, enable it for automatic startup.

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

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

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

Remote Sybase IQ Agents

If the Sybase IQ Agent is on a remote server, you need to specify the host name for that server to start the agent.

When you use scripts to start the Sybase IQ Agent, use the optional **-host** parameter on the **S99SybaseIQAgent15** command 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 IQAgent 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

Running the Sybase IQ Agent on UNIX

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.

Prerequisites

Sybase Central can be running while you start and stop agents.

Task

Enter the following command on the command line or in a script:

```
S99SybaseIQAgent15
```

See also

- *The Sybase IQ Agent* on page 24

Starting Sybase Central on Windows

Before you can administer your multiplex, you must start Sybase Central.
To start Sybase Central on Windows, run Sybase > Sybase IQ 15.4 > Sybase Central v6.1 from the Programs menu.

See also

- *The Sybase IQ Agent* on page 24

Starting Sybase Central on UNIX

Before you can administer your multiplex, you must start Sybase Central.

1. If you have not started the Sybase IQ Agent, see *Running the Sybase IQ Agent on UNIX*.
2. To start Sybase Central on UNIX, at the command prompt, type:

   ```
   % scjview
   ```

See also

- *The Sybase IQ Agent* on page 24

Managing Servers with Sybase Central

Sybase Central is a graphical interface for managing Sybase IQ multiplex servers.

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.
   - 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.

See also

- *Administrative Shell Scripts* on page 30
- *Multiplex Reference* on page 89

Administrative Shell Scripts

You can use administrative scripts 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.

See also

- *Multiplex Reference* on page 89
- *Managing Servers with Sybase Central* on page 29
- *Multiplex Database Files* on page 15

Creating Administrative Shell Scripts

Create administrative shell scripts for your multiplex servers from the Sybase Central navigation pane.

1. Right click the server
2. Choose Generate Admin Scripts from the popup menu.

Scheduling Administrative Shell Scripts on UNIX

Schedule administrative scripts to run overnight or perform tasks that you need to repeat periodically.
You can use the **crontab** system utility to schedule administrative tasks.

Scheduling Administrative Shell Scripts on Windows

Schedule administrative scripts to run overnight or perform tasks that you need to repeat periodically.

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

Multiplex Server Synchronization

Sybase IQ generally propagates DDL and DML changes to secondary servers, so you only need to synchronize a secondary server for certain server management actions.

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

The multiplex automatically makes committed changes, including global schema changes, changes to data in IQ tables visible on all servers in the multiplex.

You only need to synchronize in the following situations:

- 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

See also

- *Adding a Catalog Dbspace* on page 52
- *Dropping a Catalog Dbspace* on page 53
- *Changing an IQ_SYSTEM_MAIN File Path* on page 48
- *Adding Space to IQ_SYSTEM_MAIN on a Coordinator* on page 49
- *Restoring IQ Store Data When Restoring the Database in the Same Location* on page 86
- *Restoring IQ Store Data When Restoring Database in a Different Location* on page 84

Synchronizing Individual Secondary Servers with Sybase Central

Copy the coordinator's version of the database catalog to a single secondary server while running queries on other servers.

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 with Sybase Central

Copy the coordinator's version of the database catalog to secondary servers.

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.

See also

- *Updates on IQ_SYSTEM_MAIN* on page 47

Editing the params.cfg File to Start Servers with Sybase Central

The *params.cfg* file holds values for switches to control server startup features such as main and temp cache settings and error logging.

Prerequisites

Before you start the server, review and edit the `params.cfg` file located in the database directory for the server.

If you use Sybase Central to start the database:

Task

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`).

The *Utility Guide* describes startup switches. For more about `params.cfg`, see the *System Administration Guide: Volume 1.*

Adding Multiplex Servers from Sybase Central

Follow these steps to add multiplex servers from Sybase Central.

1. Check the server status. The coordinator must be running in order to add secondary servers.
2. Right-click the server name in the Multiplexes folder.
3. Choose Add Server from the popup.
4. Follow the instructions in the Create Server Wizard.

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. After adding this file, run **sp_iqmpxvalidate** to make sure that no errors are reported on this server.

Renaming Multiplex Servers with Sybase Central

You can use Sybase Central to change the name of a multiplex server.

1. Open the server property sheet.
2. Change the server name on the Configuration tab.

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

Changing the Database File Path with Sybase Central

If you have to move your database, for example to a disk with more space, you can change the file path with Sybase Central.

1. Open the server's property sheet.
2. On the Configuration tab, change the database file path.

Changing Server Roles with Sybase Central

You can use Sybase Central to change a reader to a write server.

1. Open the server property sheet.
2. On the General tab, change the server role (reader/writer).

Changing Host and Port with Sybase Central

You can use Sybase Central to change the host and port information for a multiplex server.

1. Open the server’s property sheet.
2. On the Configuration tab, change the host/port information.

Including or Excluding Servers with Sybase Central

If a multiplex secondary server will be shut down for an extended period of time, exclude that server from the multiplex.

Excluding a server allows the coordinator to ignore that 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.

Designating Failover Node with Sybase Central

You must designate a failover node to continue the coordinator role in the event that the current coordinator is unable to continue.

1. Right-click the Multiplex Name in the Sybase Central tree view.
2. Choose Designate Failover.

See also

- *Designated Failover Node* on page 37
- *sp_iqmpxinfo Procedure* on page 125

Dropping Multiplex Servers with Sybase Central

Dropping a secondary server removes it from the multiplex configuration.

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.

Multiplex Administration with Sybase Control Center

If you prefer a web-based tool for monitoring and administering Sybase IQ servers, use Sybase Control Center.

Sybase Control Center can add, drop, or synchronize secondary servers, or include or exclude secondary servers from the multiplex. You can also use Sybase Control Center to convert a simplex to multiplex or perform multiplex failover.

See the Sybase Control Center for Sybase IQ online help in SCC or at *http://sybooks.sybase.com/nav/summary.do?prod=10680*.

Administration Authorities

Administration authorities enable users to perform well-defined sets of database administration tasks.

For an overview of authorities that affect both simplex and multiplex servers, see *System Administration Guide: Volume 1 > Managing User IDs and Permissions > Database Permissions and Authorities Overview*.

MULTIPLEX ADMIN Authority

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.

Prerequisites

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

Task

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

When you create a new user who is to be granted an 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.

Prerequisites

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`:

Task

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
   ```

Multiplex Login Management

Login policies control the capabilities of users and connections to a database.

For an overview of Sybase IQ login policies, see *System Administration Guide: Volume 1 > Managing User IDs and Permissions > Login Management*.

Coordinator Failure

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*.

See also

- *DROP MULTIPLEX SERVER Statement* on page 101
- *Dropping Multiplex Servers with Interactive SQL* on page 21

Designated Failover Node

A multiplex requires a designated failover node to take over as coordinator if the current coordinator is not running.

You must use the failover node 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.

You can perform failover from the command line, from Sybase Central, or from Sybase Control Center. If you prefer a web-based management tool, use Sybase Control Center to designate a failover node or perform failover.

See the Sybase Control Center for Sybase IQ online help in SCC or at *http://sybooks.sybase.com/nav/summary.do?prod=10680*.

See also

- *sp_iqmpxinfo Procedure* on page 125
- *Designating Failover Node with Sybase Central* on page 34
- *Designating Failover Node with Interactive SQL* on page 20
- *Viewing the Multiplex Topology* on page 71
- *Printing the Multiplex Topology* on page 72

Requirements for Failover

You must 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.

See also

- *Synchronizing Servers from the Command Line* on page 21
- *Replacing the Coordinator from the Command Line* on page 39
- *Viewing the Multiplex Topology* on page 71
- *Printing the Multiplex Topology* on page 72

Replacing the Coordinator with 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 from the Command Line

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

Prerequisites

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.

Task

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 from the Command Line,"* but in step 2 (dbbackup), the connection string specified with the **-c** parameter must contain the new coordinator's connection parameters.

See also

- *Requirements for Failover* on page 38

Replacing the Coordinator using Sybase Control Center

If you prefer a web-based tool for multiplex failover, use Sybase Control Center.

Sybase Control Center can designate the failover node or perform coordinator node failover.

See the Sybase Control Center for Sybase IQ online help in SCC or at *http://sybooks.sybase.com/nav/summary.do?prod=10680*.

Multiplex Transactions

Multiplex transactions that modify shared objects behave according to certain rules.

Multiplex Transactions Overview

Transactions are either local or global.

Local Transactions

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.

Global Transactions

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 multiplex 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.

Coordinator Failure During Global Transaction

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.

DML Commands

In IQ 15.0, global transactions can be run from the coordinator and any write server. DML commands that 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 Commands

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 5. Multiplex table data scope

Table type	Data scope
IQ base	Global
IQ temporary	Connection
Global temporary table	Connection

Table type	Data scope
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.

Tables in Join Indexes

Some restrictions apply to tables in join indexes.

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 also

- *Server Startup Utility (start_iq)* on page 137

DDL Commands

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

DDL Command Scope

DDL commands may be propagated to all nodes or be local to the executing node.

DDL command scope is 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.

Local DDL Commands

DDL commands that affect temporary objects, which lack a permanent object ID, are local.

Local commands affect these object types:

- Local temporary tables
- Local procedures
- Temporary options

Global DDL Commands

DDL commands that create, alter, or drop a permanent object ID in the `ISYSOBJECT` table are global.

Global commands affect 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)

DDL Command Dependencies Example 1

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`.

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.

DDL Command Dependencies Example 2

When creating objects that are global, make sure that they do not depend on objects that are local.

This example creates a global object with a dependency on a local object. Assume that 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.

Restricted statements include:

- 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 6. 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

Unlisted statements are unrestricted.

Preserving Rows

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.

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"
```

Dbspace Updates in Multiplex

Dbspace updates on multiplex differ slightly from those on simplex databases.

In general, when updating shared dbspaces:

- Only the coordinator can manipulate shared IQ dbspaces.
- Before you can drop a shared dbspace file, you must alter the file to read-only mode. All servers must have advanced to the version corresponding to the **ALTER READONLY** statement, and must have returned all reserved blocks on the file.
- **CREATE DBSPACE**, **ALTER DBSPACE** and **DROP DBSPACE** commands for main and catalog dbspaces are allowed only on the coordinator

- For IQ_SHARED_TEMP behavior, see *Multiplex Transactions > DDL Commands > Role Restriction > Dbspace Updates in Multiplex > Updates on IQ_SHARED_TEMP*.

Due to delays in TLV propagation and version maintenance, there may be a delay of a few minutes before you can drop an empty dbfile or dbspace. You see a `Command not replayed` error. 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."

Never enter **ALTER DBSPACE DROP FILE** when connected using **-iqro**. **ALTER DBSPACE DROP FILE** returns the error `Modifications not permitted for read-only database` if the server is started with the **-iqro** option.

Updates on IQ_SYSTEM_TEMP

Familiarize yourself with requirements before updating IQ_SYSTEM_TEMP dbspaces.

When updating IQ_SYSTEM_TEMP:

- Create, alter, and drop temporary dbspace files only from the node where they reside. The syntax for these commands is the same as for simplex temporary dbspaces.
- Secondary servers must be synchronized 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 when you can run **ALTER DBSPACE ALTER FILE RENAME PATH.** If secondary servers are not synchronized, you may see a "Command not replayed" error.

Before updating dbspaces, see the overview of dbspaces and dbfiles in *System Administration Guide: Volume 1 > Overview of System Administration > Data Storage in Sybase IQ*.

Updates on IQ_SYSTEM_MAIN

The IQ_SYSTEM_MAIN dbspace manages important database structures including the free list, which lists blocks in use.

Before adding space to IQ_SYSTEM_MAIN, shut down secondary nodes. Active secondary nodes shut down automatically if a dbfile is added to IQ_SYSTEM_MAIN. 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 applies only to the IQ_SYSTEM_MAIN dbspace. Other dbspace operations cause no disruption and all nodes in the multiplex continue to run. See *Adding space to IQ_SYSTEM_MAIN on a coordinator*.

When updating IQ_SYSTEM_MAIN:

- If any shared IQ main dbspace files have paths inaccessible from a secondary node, that secondary node cannot access the file or any contents of that file until the path is corrected. See *Files on Shared Disk Arrays*.
- **ALTER DBSPACE ALTER FILE RENAME PATH** is prohibited on IQ_SYSTEM_MAIN.

After adding new files to IQ_SYSTEM_MAIN, synchronize and restart secondary nodes.

See also

- *Adding Space to IQ_SYSTEM_MAIN on a Coordinator* on page 49
- *Synchronizing Multiple Servers with Sybase Central* on page 32
- *Synchronizing Servers from the Command Line* on page 21

Changing an IQ_SYSTEM_MAIN File Path

The paths of all dbfiles in IQ_SYSTEM_MAIN must be visible to the secondary nodes. Otherwise, the secondary node does not start, and reports `Error opening DBFILE 'filepath'`.

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:

   ```
   ALTER DBSPACE IQ_SYSTEM_MAIN ALTER FILE
   mainfile READONLY
   ```
4. Run:

   ```
   sp_iqemptyfile mainfile
   ```
5. Drop the dbfile:

   ```
   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'
   ```

   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. See *Multiplex Server Synchronization*.

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.
```

See also

- *Multiplex Server Synchronization* on page 31

Adding Space to IQ_SYSTEM_MAIN on a 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.

Should the coordinator abort due to lack of space, 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**:

   ```
   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:

Platform	Action
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.
Windows	`ALTER DBSPACE IQ_SYSTEM_MAIN ADD FILE mymainfile '\\.\PhysicalDrive3'`

4. Restart the coordinator normally, without the **-iqmpx_sn 1** switch.
5. Synchronize secondary nodes. For details, see *Multiplex Server Synchronization*.

See also

- *Updates on IQ_SYSTEM_MAIN* on page 47
- *Multiplex Server Synchronization* on page 31

Replacing Dbfiles in IQ_SYSTEM_TEMP on a Multiplex Coordinator

To replace a damaged dbfile, drop the file, restart the server, and add the dbfile in the coordinator.

1. Use the **-iqnotemp** parameter to start the database.

   The only temporary file operation allowed on a database while running with **-iqnotemp** is to drop one or more of the temporary file definitions.
2. Drop the last file in IQ_SYSTEM_TEMP:

   ```
   ALTER DBSPACE IQ_SYSTEM_TEMP DROP FILE filename
   ```
3. Stop the database.
4. Start the database in single-node mode with no files to empty the checkpoint log:

   ```
   start_iq @params.cfg mpxtest.db -iqmpx_ov 1
   -iqmpx_sn 1 -n mpxnode_c
   ```
5. Add a “dummy” dbfile to the coordinator. You will need to drop this file and add it again in coordinator mode, because files added in single-node mode have a null server ID; they are owned by the coordinator instead of the server that added the file. To add more space to IQ_SYSTEM_MAIN with a new file on a raw device, use syntax like the following:

Platform	Syntax

UNIX	`ALTER DBSPACE IQ_SYSTEM_TEMP ADD FILE tempfile`
	`'/dev/rdsk/c4t600A0B80005A7F5D0000024'`
	where tempfile is the logical or chosen logical name of the new dbfile.

Windows	`ALTER DBSPACE IQ_SYSTEM_TEMP ADD FILE`
	`tempfile '\\.\PhysicalDrive3'`

6. Stop and restart the server.
7. Drop the dummy file.

Updates on IQ_SHARED_TEMP

When you update IQ_SHARED_TEMP dbspaces, certain restrictions apply.

These rules affect IQ_SHARED_TEMP dbspace updates:

- Only the coordinator can manipulate shared IQ dbspaces.
- Start the coordinator in single-node mode before dropping files from IQ_SHARED_TEMP. The first file made read-write in IQ_SHARED_TEMP must be the last file dropped. You may also drop files in IQ_SHARED_TEMP in simplex.

  Note: Never enter **ALTER DBSPACE DROP FILE** when connected using **-iqro**. If you start the server with **-iqro**, adding or dropping files returns the error `Modifications not permitted for read-only database`.
- Updates on the IQ_SHARED_TEMP store require SPACE ADMIN authority.

- **ALTER FILE RENAME PATH** is not allowed for the IQ_SHARED_TEMP dbspace.
- **ALTER DBSPACE ALTER FILE READONLY** is not allowed for the IQ_SHARED_TEMP dbspace.

Shared File Status Data

When opening databases and executing certain commands, multiplex nodes maintain a file status for each shared dbspace file.

The coordinator maintains status data for all nodes, and each secondary node maintains its own file status data.

All included servers must return valid read-write status for a new IQ_SHARED_TEMP file, or the statement returns an error. If all nodes can access the newly added file, they return a `Valid status` message. All secondary servers return file status for all shared files to the coordinator. Sybase IQ maintains shared file status information for dbfiles of all shared dbspaces, but uses the information only in dbspace update commands that target the IQ_SHARED_TEMP dbspace.

If a secondary node does not have write access to one or more files in IQ_SHARED_TEMP, that node will not be eligible for participating in any DQP operations. Diagnose distributed file problems using **sp_iqmpxfilestatus**.

Adding Dbfiles to Shared Dbspaces

Perform these steps in Interactive SQL or Sybase Central.

Prerequisites

When adding space to IQ_SHARED_TEMP, consider the distributed query processing workload. Dbspace commands on IQ_SHARED_TEMP require SPACE ADMIN authority.

Task

1. Connect to the coordinator.
2. Add the file. On the ADD FILE clause, specify either a full path to a raw device, or a soft link. These examples specify full paths:

Platform	Syntax

UNIX	`ALTER DBSPACE IQ_SHARED_TEMP ADD FILE mydbfilename`
	`'/dev/rdsk/c4t600A0B80005A7F5D0000024'`

Windows	`ALTER DBSPACE IQ_SHARED_TEMP ADD FILE mydbfilename`
	`'\\\\.\\PhysicalDrive2'`

This example specifies a soft link on a UNIX system:

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

You need not synchronize or restart any secondary servers.

3. Confirm that the file is visible to all secondary nodes:

   ```
   sp_iqmpxfilestatus
   ```

4. Shared files on IQ_SYSTEM_MAIN and user main dbspaces are implicitly read-write. Dbfiles on IQ_SHARED_TEMP are created read-only. If the dbfile is on a shared temporary dbspace, alter the new file to read-write status:

   ```
   ALTER DBSPACE IQ_SHARED_TEMP ALTER FILE mydbfilename
   READWRITE
   ```

Next

If an error is returned regarding the file status issues from one or more nodes, run the **sp_iqmpxfilestatus** procedure to troubleshoot and correct problems. If you postpone correction, you can force read-write status on IQ_SHARED_TEMP dbfiles:

```
ALTER DBSPACE IQ_SHARED_TEMP ALTER FILE mydbfilename
FORCE READWRITE
```

Use of the FORCE READWRITE clause returns an error on IQ_SYSTEM_MAIN and user main dbfiles.

Never enter **ALTER DBSPACE ADD FILE** when connected using **-iqro**. **ALTER DBSPACE ADD FILE** returns the error `Modifications not permitted for read-only database` if the server is started with the **-iqro** option.

Adding a Catalog Dbspace

Under normal circumstances, you will need not add catalog dbspaces; they 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.

1. Shut down all servers in the multiplex.
2. Start the coordinator in single node mode:

   ```
   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 without the **-iqmpx_sn1** switch:

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

5. Synchronize all secondary servers in the multiplex. See *Multiplex Server Synchronization*.

See also

- *Multiplex Server Synchronization* on page 31

Dropping a Catalog Dbspace

Under normal circumstances, you need not remove catalog dbspaces; they automatically grow as needed.

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. In Interactive SQL, enter:

   ```
   DROP DBSPACE DspCat2
   ```
4. Restart the coordinator normally, without the **-iqmpx_sn** 1 switch.
5. Synchronize all secondary servers in the multiplex. See *Multiplex Server Synchronization*.

See also

- *Multiplex Server Synchronization* on page 31

Dynamic Collisions

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.

Dynamic collisions only occur on secondary nodes. In an interactive setting, a dynamic collision results in a forced disconnect of a user connection from 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 with a message similar to:

```
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, schedule schema changes when the multiplex is not heavily loaded.

Logical Servers

Logical servers provide resource provisioning for the IQ multiplex, allowing you to group together a subset of computing resources (multiplex servers) as a logical entity.

Logical Servers Overview

A logical server allows one or more servers of a multiplex to be grouped together to serve one or more applications or users.

When you connect to a logical server and run a query, Sybase IQ distributes the query execution to only those nodes that are members of the logical server. You can dynamically add or drop member nodes for a logical server to accommodate the changing resource needs of applications.

Access to Sybase IQ multiplex nodes is provided only via logical servers. You can create multiple logical servers to serve different groups of applications or users. When you create a logical server, explicitly select one or more multiplex servers to add them as members. A multiplex node can be a member of more than one logical server. A logical server can have both reader and writer nodes. You can also make the coordinator node a member of a logical server, although specific rules govern coordinator membership.

A workload directed to a logical server can only be processed by member servers. Only queries that can be parallelized are eligible for work distribution; all other operations are confined to the node of initial connection. Users must connect to a node that has the role (reader or writer) required for the intended operation.

Built-in Logical Servers

OPEN and SERVER are built-in logical servers that always exist; their membership is implicitly defined, and is never stored in the catalog.

OPEN Logical Server

A node that is not a part of any user-defined logical server is implicitly a member of the OPEN logical server. The OPEN logical server consists of all multiplex nodes that are not members of any user-defined logical servers.

You cannot directly manipulate membership of a multiplex node into the OPEN logical server; it changes implicitly when you add multiplex nodes to, or drop them from, user-defined logical servers. The OPEN logical server may be empty when each server of a multiplex is a member of one or more logical servers.

Note: When there are no user-defined logical servers, the OPEN logical server consists of all multiplex servers.

The OPEN logical server supports these use cases:

- A large multiplex deployment is likely to have some applications that require provisioning of their own resources. It is also likely to have other applications that are used in an ad hoc manner, and do not require dedicated resources. These ad hoc applications can be served by the OPEN logical server, rather than by user-defined logical servers.
- In a utility computing environment where resources are used on a chargeback basis, you can keep the OPEN logical server empty as all applications and users access the multiplex with their assigned logical servers.

SERVER Logical Server

On each multiplex server, the SERVER logical server has implicit membership to that server only, allowing certain privileged users to connect to any node of the multiplex and execute server-level administrative tasks without knowing which logical servers include that node.

You must have one of these authorities to connect to a multiplex with the SERVER logical server context:

- DBA
- MULTIPLEX ADMIN
- SPACE ADMIN
- OPERATOR

NONE Logical Server

The NONE logical server is defined to be always empty.

Effective Logical Server Configuration

Not all member nodes of a logical server may be available at all times due to failure or exclusion from the multiplex.

The effective logical server configuration represents the current dynamic constitution of the logical server consisting of all member nodes that are actually available for use. The effective logical server configuration is based on the static logical server configuration and dynamic state of the multiplex.

Coordinator as a Logical Server Member

Logical server memberships that are defined using the multiplex server name of the current coordinator server are ineffective.

These memberships become effective again when the multiplex server no longer acts as the current coordinator.

Logical Membership of the Coordinator

You can define the coordinator's logical membership in a logical server using the **FOR LOGICAL COORDINATOR** clause instead of the multiplex server name of the current coordinator.

The logical membership of the coordinator means the coordinator is always available to the logical server as its member, regardless of which multiplex node plays the coordinator role.

Note: The coordinator role can move from one multiplex server to another, for example, during a failover. If a logical server has logical membership of the coordinator, the new coordinator server becomes part of the effective configuration of the logical server.

The current coordinator node is available only to those logical servers that have the coordinator as its logical member, and not to those logical servers that have membership to the current coordinator node using the multiplex node name. If the current coordinator is both a logical member and a named member of a logical server, then it is available to the logical server, but only by virtue of its logical membership. The named membership is still considered ineffective.

The coordinator node membership rules ensure predictability of the workload on the coordinator. The coordinator is available to a known set of logical servers and that does not change as the coordinator fails over from one multiplex node to another.

Logical Server Policy

Each logical server is associated with a logical server policy. A logical server policy affects certain aspects and behavior of associated logical servers via configurable logical server policy options.

Sybase IQ database includes a built-in root logical server policy that applies to all logical servers. The only change you can make in the root policy is to set the **ALLOW_COORDINATOR_AS_MEMBER** option to ON (the default) or OFF. You cannot drop the root logical server policy.

Note: You cannot create user-defined logical server policies.

ALLOW_COORDINATOR_AS_MEMBER Option

Use `ALLOW_COORDINATOR_AS_MEMBER` option of the root logical server policy to control the coordinator's eligibility to participate in a user-defined logical server. Set the option to OFF to prevent coordinator from being used as a member of any user-defined logical servers. The default setting is ON.

You can define logical membership of the coordinator as well as named membership to the current coordinator server even when `ALLOW_COORDINATOR_AS_MEMBER` is set to OFF, but these memberships will not be effective.

Note: The named memberships for the current coordinator server are always ineffective, regardless of the setting of the `ALLOW_COORDINATOR_AS_MEMBER` option. Changing the setting only affects logical memberships of the coordinator. For example, changing the current setting of `ALLOW_COORDINATOR_AS_MEMBER` from OFF to ON causes the current coordinator to become available to logical servers that have the logical membership of the coordinator.

Logical Server Administration Using Interactive SQL

If you prefer the flexibility of scripts or are familiar with Interactive SQL from other products, use Interactive SQL to administer logical servers.

Creating a Logical Server Using Interactive SQL

Create a user-defined logical server using Interactive SQL.

1. Connect to the database as a user with DBA or MPX ADMIN authority.
2. Execute a **CREATE LOGICAL SERVER** statement.

   For example, to create a user-defined logical server `ls1` with three multiplex nodes, `n1`, `n2`, and `n3`, as its members:

   ```
   CREATE LOGICAL SERVER ls1 MEMBERSHIP (n1, n2, n3);
   ```

See also

- *CREATE LOGICAL SERVER Statement* on page 98

Altering a Logical Server Using Interactive SQL

Alter a user-defined logical server using Interactive SQL.

1. Connect to the database as a user with DBA or MPX ADMIN authority.
2. Execute an **ALTER LOGICAL SERVER** statement.

   For example, to alter a user-defined logical server `ls1` by adding multiplex nodes `n4` and `n5`:

   ```
   ALTER LOGICAL SERVER ls1 ADD MEMBERSHIP (n4, n5) ;
   ```

See also

- *ALTER LOGICAL SERVER Statement* on page 89

Altering Root Logical Server Policy Using Interactive SQL

Alter the root logical server policy of a user-defined logical server using Interactive SQL.

1. Connect to the database as a user with DBA or MPX ADMIN authority.
2. Execute an **ALTER LS POLICY** statement:

```
ALTER LS POLICY root ALLOW_COORDINATOR_AS_MEMBER=ON;
```

See also

- *ALTER LS POLICY Statement* on page 94

Adding a Comment to a Logical Server Using Interactive SQL

Comment on a user-defined logical server using Interactive SQL.

1. Connect to the database as a user with DBA or MPX ADMIN authority.
2. Execute a **COMMENT ON LOGICAL SERVER** statement.

   For example, to add a comment to a user-defined logical server `ls1`, enter:

```
COMMENT ON LOGICAL SERVER ls1 IS ‘ls1: Primary Logical Server’;
```

See also

- *COMMENT ON LOGICAL SERVER Statement* on page 97

Dropping a Logical Server Using Interactive SQL

Drop a user-defined logical server using Interactive SQL.

1. Connect to the database as a user with DBA or MPX ADMIN authority.
2. Execute a **DROP LOGICAL SERVER** statement.

   For example, to drop a user-defined logical server `ls1`:

```
DROP LOGICAL SERVER ls1
```

See also

- *DROP LOGICAL SERVER Statement* on page 100

Results of Dropping Logical Servers

Dropping a user-defined logical server results in dropping all node membership definitions.

Each login policy that has an explicit assignment to the logical server drops the logical server assignment from the login policy. However, if the logical server is the only one that is assigned to the login policy, then the logical server assignment for the login policy is set to NONE.

Note: Existing connections to a logical server remain unaffected when it is dropped. Sybase recommends that you ensure that there are no active connections for a logical server when it is being dropped.

See also

- *Dropping a Logical Server Using Sybase Central* on page 61

Logical Server Administration Using Sybase Central

If you are a new user, or more comfortable with graphical interfaces, use Sybase Central to administer logical servers.

Creating a Logical Server Using Sybase Central

Create a user-defined logical server using Sybase Central.

1. In Sybase Central, connect to the database as a user with DBA or MPX ADMIN authority.
2. In the left pane, right-click the Logical Servers folder and select New > Logical Server.
3. Type a name for the logical server. Click Next.
4. Select the multiplex node to add to the logical server. Click Add, then click Next.
5. Type a comment describing the logical server, and click Finish.

Altering a Logical Server Using Sybase Central

Alter a user-defined logical server using Sybase Central.

1. In Sybase Central, connect to the database as a user with DBA or MPX ADMIN authority.
2. In the Logical Servers folder list, right-click the logical server name and select Properties.
3. In the General tab, enter any comments.
4. In the Servers tab, select the available multiplex nodes to add to the logical server. Click Add.
5. (Optional) Click Remove to delete selected multiplex nodes.
6. Click Apply.
7. Click OK.

Altering Root Logical Server Policy Using Sybase Central

Alter the root logical server policy of a user-defined logical server using Sybase Central.

1. In Sybase Central, connect to the database as a user with DBA or MPX ADMIN authority.
2. In the left pane, click the Logical Server Policy folder.
3. In the right pane, right-click the ‘root’ logical server policy and select Properties.

4. Select the value of ALLOW_COORDINATOR_AS_MEMBER.
5. Click Apply.
6. Click OK

Adding a Comment to a Logical Server Using Sybase Central

Comment on a user-defined logical server using Sybase Central.

1. In Sybase Central, connect to the database as a user with DBA or MPX ADMIN authority.
2. In the Logical Servers folder list, right-click the logical server name and select Properties.
3. In the General tab, type a comment describing the logical server.
4. Click Apply.
5. Click OK.

Dropping a Logical Server Using Sybase Central

Drop a user-defined logical server using Sybase Central.

1. In Sybase Central, connect to the database as a user with DBA or MPX ADMIN authority.
2. In the Logical Servers folder list, right-click the logical server name and select Delete.
3. Click Yes to confirm the deletion.

See also

- *Results of Dropping Logical Servers* on page 59

Logical Server Administration Using Sybase Control Center

If you prefer a web-based tool for managing logical servers, use Sybase Control Center.

Sybase Control Center can add or delete logical servers, change logical server properties or policies, configure logical server node membership, or alter a logical server assignment.

See the Sybase Control Center for Sybase IQ online help in SCC or at *http://sybooks.sybase.com/nav/summary.do?prod=10680*.

Logical Server Access

Multiplex servers can only be accessed by using logical servers. Login policies control user access to the logical servers.

You can assign one or more logical servers to a login policy. All users using the login policy can only access those multiplex servers that are effective member of the assigned logical servers. You can make one of these assignments to a login policy:

- One or more user-defined logical servers, and/or the OPEN logical server.
- SERVER — allows access to all of the multiplex nodes, subject to the semantics of the SERVER logical server.
- NONE — denies access to any multiplex server.

Note: You cannot specify SERVER or NONE in combination with other logical servers.

Node Membership Overlap

When multiple logical servers are assigned to a single login policy, no overlap in node membership is allowed among those logical servers, including the logical membership of the coordinator.

An automatic membership overlap check is enforced during changes to:

- Logical server membership — If one or more member nodes are added to a logical server, each login policy that allows access to this logical server is examined to ensure that the new membership does not overlap with memberships of other logical servers assigned to the login policy. Attempt to change a logical server membership fails if such an overlap occurs.
- Logical server assignment of a login policy — If one or more user-defined logical servers are added to a login policy, a check ensures that the new logical server assignment of the login policy does not cause a membership overlap among assigned logical servers. Attempt to change logical server assignment for a login policy fails if such an overlap occurs.

Note: Logical servers can be defined with overlapping memberships as long as they are not assigned to the same login policy.

Logical Server Assignment Inheritance

A login policy inherits logical server assignments from the root login policy when it has no logical server assignments of its own. By default, the logical server assignment of the root login policy includes only the OPEN logical server. This ensures that all multiplex servers are accessible when default configuration is in use.

The effective logical server assignment of a login policy is either from the logical server assignment that is made explicitly to the login policy or from the inheritance of logical server assignments of the root login policy.

You can completely remove existing logical server assignments from a user-defined login policy by setting the logical server assignment to DEFAULT. Setting the logical server assignment to DEFAULT in the root login policy restores the default assignment of OPEN logical server.

Note: Logical server assignment of NONE is different from the case when there are no logical server assignments.

Logical Server Context of a Connection

When you establish a user connection, the logical server context of the connection is automatically determined by the user's login policy and the current node:

- When effective logical server assignment of the user's login policy is one or more logical servers, the logical server context of the connection is based on the current node's unambiguous membership into one of the specified logical servers.
- A connection fails if the current node is not a member of any of the logical servers assigned to the user's login policy.
- A connection fails if effective logical server assignment of user's login policy is SERVER logical server and user does not have sufficient authority.
- A connection also fails if logical server assignment for the user's login policy is NONE.

Lock-down Prevention

To prevent a lock-down situation due to login policy logical server access configuration, DBA connections will not fail even if the login policy prevents access to the node.

When the connection cannot be accepted based upon login policy's logical server access configuration, then the DBA connection is accepted with SERVER logical server context.

Setting Logical Server Assignment When Creating a Login Policy Using Sybase Central

Create logical server assignment using Sybase Central.

1. In Sybase Central, connect to the database as a user with DBA or USER ADMIN authority.
2. In the left pane, right-click the Logical Policies folder and select New > Login Policy.
3. Set the appropriate policy options, then click Next.
4. Select the logical server assignment type and add the logical servers to the login policy. Click Add, then click Next.
5. Select the appropriate logical server option override values, then click Next.
6. Type a comment describing the login policy, and click Finish.

Setting Logical Server Assignment When Altering a Login Policy Using Sybase Central

Alter logical server assignment using Sybase Central.

1. In Sybase Central, connect to the database as a user with DBA or USER ADMIN authority.
2. In the left pane, click Login Policies.
3. In the right pane, right-click a login policy and choose Properties.
4. In the General tab, alter the appropriate policy options.

5. In the Logical Server Assignment tab, alter the logical server assignment type and add or remove the logical servers to the login policy. Click Add or Remove, then click Next
6. In the Logical Server Option Override tab, alter the appropriate logical server option override values.
7. Click Apply.
8. Click OK.

Setting Logical Server Assignment When Altering a Login Policy Using Interactive SQL

Alter logical server assignment using Interactive SQL.

1. Connect to the database as a user with DBA or USER ADMIN authority.
2. Execute an **ALTER LOGIN POLICY** statement. The **LOGICAL SERVER** clause lets you configure values of certain login policy options on a per logical server basis.

   For example, to alter a logical server assignment by assigning logical server `ls1` to the login policy `lp1`, enter:

   ```
   ALTER LOGIN POLICY lp1 ADD LOGICAL SERVER ls1;
   ```

See also

- *ALTER LOGIN POLICY Statement* on page 90

Login Policy Options

You can specify a per logical server level setting for the login policy options **max_connections** and **dqp_enabled**.

Logical server overrides for **locked** login policy option are not allowed.

Note: You cannot use the **ALTER LOGIN POLICY** command with the **MULTIPLEX SERVER** override clause for login policy options.

locked Option

Logical servers assigned to a login policy can be accessed only when the **locked** option is OFF.

Set the **locked** option to ON to lock all login policy users out. Use this setting to temporarily deny access to login policy users.

max_connections Option

The **max_connections** setting is enforced at each member node of the logical server and can be set at the base level or the per logical server level.

Note: Once a user reaches a limit of maximum connections on a logical server member node, additional user connections on that node fail. This occurs even when other member nodes may not have reached the limit for number of connections for the user.

dqp_enabled Login Policy Option

dqp_enabled login policy option enables or disables DQP for all database users assigned to that login policy.

You can also specify logical server specific overrides for **dqp_enabled** login policy option. The possible settings for the option are ON and OFF.

Note: By default, **dqp_enabled** is set to ON in the root login policy.

Distributed Query Processing Options

Query execution may be distributed to all available and eligible member nodes of the logical server. Not all users connecting to a logical server may require workload distribution at the same time.

You can enable or disable the DQP option for sets of users or at an individual connection level by setting a **dqp_enabled** login policy option and/or the temporary database option **dqp_enabled**.

See also

- *Distributed Query Processing* on page 67

dqp_enabled Temporary Database Option

Temporary database option **dqp_enabled** allows you to enable or disable DQP at the connection level.

You can set the temporary database option **dqp_enabled** to OFF to disable DQP for the current connection. You can set the option to ON (the default value) to enable DQP for the current connection, but only when DQP is enabled for the user by that user's login policy for the logical server of the current connection.

Setting **dqp_enabled** to ON results in an error if DQP is disabled based upon the user's login policy:

```
Invalid setting for option 'dqp_enabled'
```

Note: Any changes you make to a user's login policy options affect new connections only. Login policy option settings for existing connections are based upon the time the connection was initially established.

Logical Server Configuration

Only multiplexes support logical servers.

Information about built-in logical servers and logical server policies can remain in the catalog in a simplex environment, but are not used.

Logical Servers and New Simplex Databases

When you create a new simplex database, catalogs are created with the information for built-in logical servers OPEN, SERVER, and NONE.

The root logical server policy is also automatically created. The OPEN logical server is assigned to the root login policy.

Note: Executing DDL commands to create a new logical server returns an error.

Multiplex Database Conversion Into a Simplex Database

Before converting a multiplex into a simplex database, drop all user-defined logical servers.

Use **DROP MULTIPLEX SERVER** clause **WITH DROP LOGICAL SERVER** to drop all user-defined logical servers.

The catalog retains this information:

- Built-in logical servers
- ‘root’ logical server policy
- Login policy assignments to built-in logical servers
- Logical-server-level overrides for the login policy option. These could exist only for the built-in logical servers.

The information is reused if you convert the simplex database back to multiplex.

Connection in Simplex

Connecting to a simplex database is unaffected by the login policy setting of logical server assignments. The connections have no logical server context.

The login policy **locked** option is still applied before connections are accepted by the server.

Distributed Query Processing

Distributed query processing (DQP) improves performance in IQ multiplex configurations by spreading work across multiple nodes in the cluster.

You need not set any configuration options to activate distributed query processing. Unless you disable DQP by turning off the **dqp_enabled** Login Policy option or the **dqp_enabled** temporary database option, DQP occurs automatically for qualifying queries when:

- The server is part of a multiplex where servers have established MIPC connections. Servers establish these connections automatically on startup. To verify, run **sp_iqmpxinfo** and look for *active* status in the `mipc_public_state` column.

  If you configured private interconnect on the multiplex and expect DQP to run on private interconnect, run **sp_iqmpxinfo** and look for *active* status in the `mipc_private_state` column.
- The logical server of the current connection has at least one another member node available.

  Note: By partitioning multiplex server nodes into logical servers, you control which multiplex nodes participate in distributed query processing.
- The shared temporary dbspace has writable files available. See *System Administration Guide: Volume 1 > Database Object Management > Database Definition > Database Creation with SQL > Database File Placement* and *System Administration Guide: Volume 1 > Database Object Management > Setting Up a Sybase IQ Database > Space Allocation* .

Not all users connecting to a logical server may need to have their workload distributed at all times. Use the distributed query processing options to enable or disable DQP for sets of users, and at the individual connection level. Use the **dqp_enabled** Login Policy option to enable/ disable DQP for sets of users. Use the **dqp_enabled** temporary database option to enable/ disable DQP at the individual connection level.

See also

- *Distributed Query Processing Options* on page 65

Leader and Worker Nodes

In distributed query processing, leader nodes pass work units to worker nodes and the work is performed by threads running on both the leader and worker nodes.

The leader node can be any node in the cluster where a query originates. A worker node can be any node in the cluster that is capable of accepting distributed query processing work. Do not confuse these nodes with multiplex coordinator, writer, and reader nodes.

You can view details of distributed query processing thread usage using the **sp_iqcontext** system stored procedure.

When a query is submitted to a node, work units may be distributed, but only to those nodes that are members of the logical server of the current connection. Multiplex nodes that are not members of the current connection's logical server do not take part in the distributed query processing for that query. The leader node automatically chooses worker nodes for the distributed query from within the same logical server as the leader node. If you exclude multiplex nodes from a logical server, no distributed query processing occurs on those nodes for that logical server.

Query Portions

Queries are distributed in separate portions.

These query portions are never distributed:

- Read-write query portions (including DDL, INSERT, LOAD, UPDATE, and DELETE)
- Query portions that reference temporary tables
- Query portions that reference SYSTEM dbspace tables
- Query portions that reference proxy tables
- Query portions that reference nondeterministic functions, such as NEWID

Distributed Query Performance

In general, the more nodes and resources that are available, the better the potential query performance.

Distributed query processing uses the available memory and CPU resources of all nodes of the logical server.

The amount of improvement benefit depends on the type of query, the size of the query, and the current workload of the nodes in the logical server.

Note: If you change the properties of multiplex server, including the server name, hostname, and port, then you must wait at least two minutes after restarting the multiplex server for it to participate in a DQP eligible query. In the first two minutes after restarting the server, if a DQP eligible query is executed, then the server may not participate.

It is unlikely that any two runs of the same query result in exactly the same work distribution — as load levels change in the cluster, so does the load distribution. Distributed query performance is determined by the overall workload of the logical server at any given time. Similarly, in a single run of a query with a long processing time, the work distribution changes over the course of query execution as the load balance changes across worker nodes.

Note: The **-iqmc** and **-iqtc** switches allow different cache sizes for each node in a multiplex, but this may have adverse affects. For example, if a node worker is configured with a much smaller cache than the leader, hash joins on the leader will operate in a paging mode that disallows parallelism.

A high-speed private interconnect is preferred for best distributed query performance, but not required. See the *Installation and Configuration Guide > Preparing for Installation > Planning Your Installation > Planning for Distributed Query Processing or High Availability.*

Note: Do not use the **NOEXEC** option to examine DQP performance. **NOEXEC** is not useful for troubleshooting DQP.

Sybase Central Performance and Statistics Monitor

The topology view graphically represents the entire multiplex environment, and complements the existing list view, making it easy to visually monitor the state of the multiplex environment.

Topology View

The graphical detail pane is available from the Multiplexes container in the Sybase IQ plug-in for Sybase Central.

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* and *Requirements for Failover*.

See also

- *Viewing the Multiplex Topology* on page 71
- *Printing the Multiplex Topology* on page 72

Viewing the 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.

See also

- *Topology View* on page 71
- *Designated Failover Node* on page 37
- *Requirements for Failover* on page 38

Printing the Multiplex Topology

You can print the performance graph in the Sybase Central topology view.

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.

See also

- *Topology View* on page 71
- *Designated Failover Node* on page 37
- *Requirements for Failover* on page 38

Saving the Graph

You can save the performance monitor graph in Sybase Central.

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

Changing the Refresh Rate of the Multiplex Status Monitor

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.

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

Viewing Node Information

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

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

The topology view displays the communication link from each secondary node to the coordinator 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* and *Pooled Connections*.

See also

- *Heartbeat Connections* on page 8
- *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.

Performance Monitor Access

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.

The Configure Collection Rates Dialog

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.

Monitoring a Single Statistic

Launch the Configure Collection Rates dialog from the multiplex-level performance monitor if you only need to 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. *Performance Monitor Access* 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

Sybase Central lets you save the Performance Monitor 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 Views

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

Customize the chart to change chart settings and chart refresh (monitor GUI heartbeat) rate.

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

CPU usage statistics display the percentage used by system, owner, and total CPU usage.

Table 7. CPU usage

Name	Description	Monitored by default?
CPU Usage	IQ process CPU usage percentage, including both system and user usage	Yes

Name	Description	Monitored by default?
CPU System Usage	IQ process CPU system usage percentage	No
CPU User Usage	IQ process CPU user usage percentage	No

Memory Usage Statistics

Memory usage statistics show the current allocation and the maximum allocation, in megabytes.

Table 8. 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

Cache statistics display the main, temporary, and catalog hits and reads per second, the percentage of each cache in use, pinned, and dirty, and the number of pinned pages.

Table 9. 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
Main Cache Reads	Number of main cache page lookups per second	No
Catalog Cache Current Size	Current catalog cache size, in megabytes	No

Name	Description	Monitored by default?
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

Thread statistics display the number of SA and IQ threads in use and the number of IQ threads available.

Table 10. 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

Connection statistics display the number of user connections, the number of user connections and disconnections per minute, and the number of INC incoming and outgoing connections.

Table 11. 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

Request statistics display the number of requests, unscheduled requests, waiting, and active operations.

Table 12. 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
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

Transaction statistics display the number of active user and INC transactions and the number of active load table statements.

Table 13. 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

Store I/O statistics display the number of kilobytes per second read from and written to the catalog store, main store, and temporary store.

Table 14. 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

Dbspace statistics display the amount of space available and in use per file and the amount of space available and in use per dbspace.

Table 15. 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

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

Network Statistics

Network statistics display the number of bytes per second sent and received during client server communications and the number of available and total communication buffers.

Table 16. 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

Multiplex Backup and Restore Operations

Execute **BACKUP** and **RESTORE** SQL commands only on the coordinator node.

Checking Database Consistency

Check database consistency before backing up.

The **sp_iqcheckdb** procedure performs various functions, depending on the parameters specified.

The four modes for checking and repairing database consistency are described in *Reference: Building Blocks, Tables, and Procedures > System Procedures > sp_iqcheckdb Procedure*.

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

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

Backup Requirements

Execute the **BACKUP** SQL command only on the coordinator node.

- 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.
- If using virtual backup, 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.

Backing Up IQ Stores and Catalog Stores

Follow these guidelines to back up multiplex databases.

1. 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.

2. Issue a SQL BACKUP command. For complete syntax, see *Reference: Statements and Options > SQL Statements > BACKUP Statement*.
3. Back up the IQ store as described in *System Administration Guide: Volume 1 > Data Backup, Recovery, and Archiving > Types of Backups*.

Before You Restore

Before you restore, 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 *Multiplex Server Synchronization*. 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.

For complete syntax, see *Reference: Statements and Options > SQL Statements > RESTORE Statement*.

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.

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

See also

- *Restoring IQ Store Data When Restoring the Database in the Same Location* on page 86
- *Restoring IQ Store Data When Restoring Database in a Different Location* on page 84

Restoring IQ Store Data When Restoring Database in a Different Location

Restore operations vary depending on where you restore the data.

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. Confirm that the database shut down successfully:

Platform	Actions
UNIX	`% ps -ef | grep iqsrv15` If you see an active **iqsrv15** process with name of a multiplex, stop the process.
Windows	In 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 *System Administration Guide: Volume 1 > Data Backup, Recovery, and Archiving > System-Level Backups > Back Up the Right Files*.

5. Shut down the original coordinator, then start the utility database from the coordinator server directory using the coordinator server's name:

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

6. Connect to the utility database (`utility_db`).

```
% dbisql -c "eng=coordinator_svr;uid=DBA;pwd=SQL;
dbn=utility_db" -host myhost -port 1234
```

7. Run the **RESTORE** command with a new location path for the database and its dbspaces. To restore certain dbspace files to a different path, specify a `RENAME` clause. Perform full and any incremental restore operations in sequence, without stopping the utility database.

Warning! Stopping the utility database between full and incremental restore operations may invalidate the catalog and render the restored database unusable.

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, including the database file extension (.DB):

    ```
    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.
12. When you restart the coordinator, you see a warning in the server log about the multiplex environment being invalid. This warning displays 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 *Multiplex Server Synchronization* .
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, replace steps 9 through 16 with:

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

See also

- *Before You Restore* on page 84
- *Multiplex Server Synchronization* on page 31

Restoring IQ Store Data When Restoring the Database in the Same Location

Restore operations vary depending on where you are restoring the data.

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. Confirm that the database shut down successfully:

Platform	Actions
UNIX	`% ps -ef | grep iqsrv15` If you see an active **iqsrv15** process with name of a multiplex, stop the process.
Windows	In 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 *System Administration Guide: Volume 1 > Data Backup, Recovery, and Archiving > System-Level Backups > Back Up the Right Files*.
   - 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 operating system 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 the Database in a Different Location*.

5. Start the utility database from the coordinator server directory. Use any valid identifier as the server name except the name of a registered secondary server. If you use the coordinator's name, rename the coordinator after the restore.

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

6. Connect to the utility database (`utility_db`):

   ```
   % dbisql -c "eng=utility_startup_svr;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 *Reference: Statements and Options >SQL Statements > RESTORE Statement*.
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 *System Administration Guide: Volume 1 > Data Backup, Recovery, and Archiving > System-Level Backups > Back Up the Right Files*. 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 *Multiplex Server Synchronization* .
11. Start the secondary servers.

See also

- *Before You Restore* on page 84
- *Multiplex Server Synchronization* on page 31

Selective Restore Operations in a Multiplex Environment

Restore databases either completely or selectively (by restoring only read-write dbspaces, or a set of read-only dbspaces or read-only files).

See *System Administration Guide: Volume 1 > Data Backup, Recovery and Archiving* and *Reference: Statements and Options > RESTORE Statement*.

Restoring Read-only Backups for a Coordinator

This procedure lets you restore a coordinator without renaming the utility database to use the coordinator's name. This is the only supported way to do a read-only selective restore for a multiplex coordinator.

Use this procedure to correct problems resulting from inadvertently restoring read-only dbspaces from a read-write archive or vice versa.

1. Start the utility server with any server name except that of a secondary node.
2. Connect to the utility_db and run the **RESTORE** statement for the read-write database. Use **RENAME** clauses to move dbfiles to the corresponding locations.
3. Disconnect and stop the utility server.
4. Start the restored database. If the database has been moved to a different location, start the server with **-iqmpx_sn 1**and **-iqmpx_ov 1** flags.
5. Run **ALTER DBSPACE <dbspace name>** offline for the RO dbspaces that have been backed up on the separate RO backup only.
6. Disconnect and stop the server.
7. Start the utility database with any server name except that of a secondary node.
8. Connect to the utility server and run the restore command for the RO dbspace.

Multiplex Reference

Certain SQL statements, utilities database options, and system objects have special syntax for multiplex functionality.

See also

- *Administrative Shell Scripts* on page 30
- *Managing Servers with Sybase Central* on page 29

SQL Statements

Many SQL statements include special syntax to support multiplex functionality.

ALTER LOGICAL SERVER Statement

Modifies configuration for the existing user-defined logical server in the database.

Syntax

```
ALTER LOGICAL SERVER  logical-server-name
    { alter_clause }
```

Parameters

- **alter_clause: –**

```
{ADD MEMBERSHIP   '(' { ls-member, ... } ')'
| DROP MEMBERSHIP '(' { ls-member, ... } ')'
}
```

- **ls-member: –**

```
FOR LOGICAL COORDINATOR
| mpx-server-name
```

Examples

- **Example 1 –** This example alters a user-defined logical server by adding multiplex nodes `n1` and `n2` to logical server `ls1`:

```
ALTER LOGICAL SERVER ls1 ADD MEMBERSHIP (n1, n2)
```

- **Example 2 –** This example adds logical membership of COORDINATOR and drops a named membership of the current coordinator node `n1` from logical server `ls1`:

```
ALTER LOGICAL SERVER ls1 ADD MEMBERSHIP (FOR LOGICAL COORDINATOR)
ALTER LOGICAL SERVER ls1 DROP MEMBERSHIP (n1)
```

Usage

Applies to multiplex only.

logical-server-name refers to an existing user-defined logical server name, in other words, it cannot be a built-in or reserved logical server name. The SYS.ISYSIQLSMEMBER system table stores definitions for the logical server memberships.

A member node that is added to or dropped from a logical server starts or stops accepting logical server connections only after the TLV log corresponding to **ALTER LOGICAL SERVER** is played on that node. Existing connections of a logical server continue to run on a node when that node is dropped from the logical server, however, distributed processing is stopped for these connections.

An error is returned if:

- Any *ls-member* specified with the **ADD MEMBERSHIP** clause is already a member of the logical server.
- Any *ls-member* specified with the **DROP MEMBERSHIP** clause is not an existing member of the logical server.
- Logical server membership change causes membership overlap check to fail.

Permissions

Must have DBA or MPX ADMIN authority.

See also

- *Altering a Logical Server Using Interactive SQL* on page 58

ALTER LOGIN POLICY Statement

This statement performs two functions. It can modify option values for existing login policies in the database or configure logical server access.

Syntax

```
ALTER LOGIN POLICY policy-name {alter-clause}
```

Parameters

- **alter-clause: –**

```
{ { ADD | DROP | SET } LOGICAL SERVER ls-assignment-list
      | policy-option-name = policy-option-value [ LOGICAL SERVER
ls-override-list ]
}
```

- **ls-assignment-list: –**

```
{ { ls-name, ...} | SERVER | NONE | DEFAULT }
```

- **ls-override-list: –**

  ```
  { ls-name, ...}
  ```

- **ls-name: –**

  ```
  { OPEN | user-defined-ls-name }
  ```

- **policy-option-value: –**

  ```
  { UNLIMITED | DEFAULT | value }
  ```

Examples

- **Example 1 –** See *Logical Server Access Configuration* and *Login Policy Option Configuration*.

Usage

Applies to multiplex only.

See *Logical Server Access Configuration* and *Login Policy Option Configuration*.

Permissions

Must have DBA or USER ADMIN authority.

See also

- *Setting Logical Server Assignment When Altering a Login Policy Using Interactive SQL* on page 64
- *DQP_ENABLED Option* on page 103

Logical Server Access Configuration

You can use ALTER LOGIN POLICY to configure logical server access.

Example 1

Assume that the root login policy allows access to logical servers `ls4` and `ls5` and login policy `lp1` exists with no logical server assignment. The statement below effectively assigns login policy `lp1` to logical servers `ls4` and `ls5`.

Assign logical server `ls1` to login policy `lp1`.

```
ALTER LOGIN POLICY lp1 ADD LOGICAL SERVER ls1
```

Example 2

This statement allows access of logical servers `ls2` and `ls3` from login policy `ls1`.

```
ALTER LOGIN POLICY lp1 ADD LOGICAL SERVER ls2, ls3
```

Example 3

Modify login policy `lp1` to allow access to `ls3` and `ls4`only.

```
ALTER LOGIN POLICY lp1 ADD LOGICAL SERVER ls4
```

```
ALTER LOGIN POLICY lp1 DROP LOGICAL SERVER ls1, ls2
```

or

```
ALTER LOGIN POLICY lp1 SET LOGICAL SERVER ls3, ls4
```

Example 4

Modify login policy`lp1` to deny access to any logical servers.

```
ALTER LOGIN POLICY lp1 SET LOGICAL SERVER NONE
```

Example 5

Drop current logical server assignments of login policy`lp1` and allow it to inherit the logical server assignments of the root login policy.

```
ALTER LOGIN POLICY lp1 SET LOGICAL SERVER DEFAULT
```

Usage

Applies to multiplex only.

ADD, **DROP**, or **SET** clauses let your configure the logical server assignments of a login policy:

- **ADD** – adds new logical server assignments to a login policy
- **DROP** – deletes existing logical server assignments from a login policy
- **SET** – replaces all logical server assignments for a login policy with a new set of logical server

Use only one **ADD**, **DROP**, or **SET** clause. Use **SERVER**, **NONE**, and **DEFAULT** only with the **SET** clause. Specify a particular logical server name only once per **ls-assignment list** or **ls-override list**.

An error is returned if:

- Any logical server specified with the **ADD** clause is already assigned to the login policy.
- Any logical server specified with the **DROP** clause is currently not assigned to the login policy.
- Logical server assignment change may cause a membership overlap among assigned logical servers.

`SYS.ISYSIQLOGINPOLICYLSINFO` stores logical server assignment information. For each logical server override of a login policy option of a login policy, a corresponding row exists in `ISYSIQLOGINPOLICYLSINFO`.

Login Policy Option Configuration

You can use ALTER LOGIN POLICY to configure login policy options.

Example

The following example overrides the login policy settings on two logical servers, enabling distributed query processing on logical server `ls1` and increasing the maximum number of connections on logical server `ls2`

```
ALTER LOGIN POLICY lp1 dqp_enabled=ON LOGICAL SERVER ls1;
```

```
ALTER LOGIN POLICY lp2 max_connections=20 LOGICAL SERVER ls2;
```

Usage

Applies to multiplex only.

A logical server level override means that a particular login policy option has different settings for different logical servers. `SYS.ISYSIQLSLOGINPOLICYOPTION` stores login policy option values for logical server override. For each logical server override of a login policy option of a login policy, a corresponding row exists in `ISYSIQLSLOGINPOLICYOPTION`.

Note: You can only specify a logical server override setting for login policy options **max_connections** and **dqp_enabled**. (The **dqp_enabled** option only affects multiplex servers. See *Multiplex Reference > Database Options > dqp_enabled Option*.) You cannot specify logical server overrides for the root login policy.

Table 17. Login Policy Options

Option	Description	Values	Initial value for ROOT policy	Applies to
`dqp_enabled`	If ON, enables distributed query processing for users assigned to that login policy.	ON, OFF	ON	Multiplex servers only. All users including those with DBA authority...
`locked`	If the value for this option is ON, users are prohibited from establishing new connections	ON, OFF	OFF	Users without DBA authority only
`max_connections`	The maximum number of concurrent connections allowed for a user.	0 – 2147483647	Unlimited	Users without DBA authority only

Option	Description	Values	Initial value for ROOT policy	Applies to
`max_days_since_login`	The maximum number of days that can elapse between two successive logins by the same user.	0 – 2147483647	Unlimited	Users without DBA authority only
`max_failed_login_attempts`	The maximum number of failed attempts, since the last successful attempt, to login to the user account before the account is locked.	0 – 2147483647	Unlimited	Users without DBA authority only
`max_non_dba_connections`	The maximum number of concurrent connections that a user without DBA authority can make. This option is only supported in the root login policy.	0 – 2147483647	Unlimited	Users without DBA authority only. Only to the root login policy.
`password_expiry_on_next_login`	If the value for this option is ON, the user's password will expire in the next login.	ON, OFF	OFF	All users including those with DBA authority
`password_grace_time`	The number of days before password expiration during which login is allowed but the default post_login procedure issues warnings.	0 – 2147483647	0	All users including those with DBA authority
`password_life_time`	The maximum number of days before a password must be changed.	0 – 2147483647	Unlimited	All users including those with DBA authority

Note: Sybase IQ no longer supports the **MULTIPLEX SERVER** override clause. Using **ALTER LOGIN POLICY** statement with the **MULTIPLEX SERVER** override clause for login policy options returns an error.

ALTER LS POLICY Statement

Modifies some or all option values for the existing root logical server policy in the database.

Syntax

```
ALTER LS POLICY policy-name option-value-list
```

Parameters

- **option-value-list: –**

  ```
  {option-name=value}
  ```

Examples

- **Example 1 –** The following example alters the logical server policy:

  ```
  ALTER LS POLICY root
  ALLOW_COORDINATOR_AS_MEMBER=ON;
  ```

  Note: ALLOW_COORDINATOR_AS_MEMBER is the only logical server policy option, and root is the only logical server policy. You cannot create logical server policies.

Usage

Applies to multiplex only.

Permissions

Must have DBA or MPX ADMIN authority.

See also

- *Altering Root Logical Server Policy Using Interactive SQL* on page 59

ALTER MULTIPLEX RENAME Statement

Renames the multiplex and stores the multiplex name in SYS.ISYSIQINFO system table.

Syntax

```
ALTER MULTIPLEX RENAME multiplex-name
```

Usage

Applies to multiplex only.

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

Changes the name, catalog file path, role, or status of the given server.

Syntax

Syntax 1:

```
ALTER MULTIPLEX SERVER server-name server-option
```

Syntax 2:

```
ALTER MULTIPLEX SERVER PRIVATE NULL
```

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* ...} { **PRIVATE HOST** ' *hostname* ' **PORT** *port number* ...}

  Note: Sybase recommends you shut down the target server before you exclude it. If you do not, an excluded server automatically shuts down and requires **ALTER MULTIPLEX SERVER** *server-name* **STATUS INCLUDED** and a synchronize to rejoin the multiplex.

Examples

- **Example –** This example excludes secondary server `mpx_writer1`:

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

Usage

Applies to multiplex only.

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 shut down 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.

Note: Sybase recommends that the target server be shut down 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.

See also

- *Renaming Multiplex Servers with Interactive SQL* on page 19

COMMENT ON LOGICAL SERVER Statement

Comments on the user-defined logical server.

Syntax

```
COMMENT ON LOGICAL SERVER logical-server-name IS 'comment'
```

Examples

- **Example –** This example creates a comment on a user-defined logical server *ls1*.

```
COMMENT ON LOGICAL SERVER ls1 IS ‘ls1: Primary
Logical Server’;
```

Usage

Applies to multiplex only.

Permissions

Must have DBA or MPX ADMIN authority.

See also

- *Adding a Comment to a Logical Server Using Interactive SQL* on page 59

CREATE LOGICAL SERVER Statement

Creates a user-defined logical server.

Syntax

```
CREATE LOGICAL SERVER  logical-server-name [MEMBERSHIP
    '(' { ls-member, ...} ')']
```

Parameters

- **ls-member: – FOR LOGICAL COORDINATOR** | *mpx-server-name*

Examples

- **Example 1 –** This example creates a user-defined logical server *ls1* with three multiplex nodes as its members.

  ```
  CREATE LOGICAL SERVER ls1 MEMBERSHIP ( n1, n2, n3 )
  ```

Usage

Applies to multiplex only.

The catalog stores the logical server and its membership definitions. To define a logical membership to the coordinator, specify FOR LOGICAL COORDINATOR in the MEMBERSHIP clause.

When no members are specified during the creation of a logical server, the logical server is created empty.

Note: Implicit logical server membership definitions such as those for OPEN and SERVER logical servers are not stored at all.

The SYS.ISYSLOGICALSERVER system table stores information about logical servers.

The SYS.ISYSLOGICALMEMBER system table stores information about logical server membership definitions.

logical-server-name can be any user specified identifier except:

- OPEN
- SERVER
- NONE
- DEFAULT
- COORDINATOR
- ALL

Changing ALLOW_COORDINATOR_AS_MEMBER option of the root logical server policy from ON to OFF does not affect the membership information stored in the catalog. Instead, it only affects the effective configuration of the logical server.

You can define a logical server membership to the current coordinator either by specifying the multiplex server name or by using FOR LOGICAL COORDINATOR clause, even when ALLOW_COORDINATOR_AS_MEMBER is set to OFF. Membership definition is stored in the catalog, but is inactive while that multiplex server acts as the coordinator.

Permissions

Must have DBA or MULTIPLEX ADMIN authority.

See also

- *Creating a Logical Server Using Interactive SQL* on page 58

CREATE MULTIPLEX SERVER Statement

Creates a multiplex server.

Syntax

Variable declaration:

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

Parameters

- **host-port-list: –** {[**PRIVATE**] **HOST** ' *hostname* ' **PORT** *port number* }

Examples

- **Example 1 –** See *Adding Multiplex Servers with Interactive SQL*.

Usage

Applies to multiplex only.

If you plan to use UNIX soft (symbolic) links for server paths, create the soft link before you run **CREATE MULTIPLEX SERVER**. When you start the new server, the database file path must match the database file path specified when creating that server.

Choose the name of the multiplex server (*server-name*) according to the rules for server startup option **-n**. See "Database server startup" 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.

The host-port-list keyword PRIVATE specifies that the particular HOST PORT pair is for private interconnection. A separate private interconnection for multiplex interprocess communication (MIPC) enables highly available and high-performance network configurations.Sybase IQ Sybase IQ automatically opens private ports; you need not list them in the host-port-list used to start the server. All public and private ports require unique port numbers to avoid conflicts.

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.

Permissions

Must have DBA or MULTIPLEX ADMIN authority.

See also

- *Multiplex Administration with Interactive SQL* on page 17
- *Adding Multiplex Servers with Interactive SQL* on page 18

DROP LOGICAL SERVER Statement

Drops a user-defined logical server.

Syntax

```
DROP LOGICAL SERVER  logical-server-name
```

Examples

- **Example 1 –** This example drops a user-defined logical server *ls1*.

  ```
  DROP LOGICAL SERVER ls1
  ```

Usage

Applies to multiplex only.

Sybase IQ performs the following catalog changes internally when dropping a logical server:

- Drops all membership definitions of the logical server.
- Drops its logical server assignment from each login policy that has an explicit assignment to the subject logical server. If it is the only logical server assigned to the login policy, Sybase IQ sets the logical server assignment for the login policy to NONE.
- Removes the logical server entry from ISYSIQ.LOGICALSERVER.

Permissions

Must have DBA or MULTIPLEX ADMIN authority.

See also

- *Dropping a Logical Server Using Interactive SQL* on page 59

DROP MULTIPLEX SERVER Statement

Deletes a server from the multiplex.

Syntax

```
DROP MULTIPLEX SERVER  {server-
name} [drop_mpx_server_clause]
```

Parameters

- **drop_mpx_server_clause: –** { **WITH DROP MEMBERSHIP** | **WITH DROP LOGICAL SERVER** }

Examples

- **Example 1 –**

  ```
  DROP MULTIPLEX SERVER writer1
  ```

Usage

Applies to multiplex only.

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. Please refer to section 3.3.1 for details on set of rules that apply when a multiplex is converted to a simplex.

Clause **WITH DROP MEMBERSHIP** – The **DROP MULTIPLEX SERVER** fails with an error, when one or more logical server memberships exist for the multiplex server being dropped. Use the **WITH DROP MEMBERSHIP** clause to drop the logical server along with all of its memberships.

Clause **WITH DROP LOGICAL SERVER** – When dropping the last secondary server, the **DROP MULTIPLEX SERVER** command fails, when there are one or more user-defined logical servers. Use the **WITH DROP LOGICAL SERVER** clause to drop the last secondary server along with all user-defined logical servers.

Note: The **WITH DROP LOGICAL SERVER** clause is only valid when dropping the last secondary server. An error is reported otherwise.

Permissions

Must have DBA or MULTIPLEX ADMIN authority.

See also

- *Requirements for Dropping Multiplex Servers* on page 20
- *Coordinator Failure* on page 37
- *Dropping Multiplex Servers with Interactive SQL* on page 21

Database Options

On a multiplex, database options control connections, distributed query processing and multiplex inter-node communications.

DQP_ENABLED Option

Temporary database option **dqp_enabled** allows you to enable or disable DQP at the connection level.

Allowed Values

ON, OFF

Default

ON

Scope

Can be set temporary for an individual connection. This option takes effect immediately and only affects the current connection.

Description

You can set the temporary database option **dqp_enabled** to OFF to disable DQP for the current connection. You can set the option to ON (the default value) to enable DQP for the current connection, but only when DQP is enabled for the user by that user's login policy for the logical server of the current connection.

Setting **dqp_enabled** to ON results in an error if DQP is disabled based upon the user's login policy:

```
Invalid setting for option 'dqp_enabled'
```

Note: Any changes you make to a user's login policy options affect new connections only. Login policy option settings for existing connections are based upon the time the connection was initially established.

See also

- *ALTER LOGIN POLICY Statement* on page 90

MPX_AUTOEXCLUDE_TIMEOUT Option

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 `MPX_HEARTBEAT_FREQUENCY` setting in minutes. For example, if the `MPX_HEARTBEAT_FREQUENCY` setting is 120 (2 minutes), `MPX_AUTOEXCLUDE_TIMEOUT` must be divisible by 2.

Default

60 minutes

Scope
Can be set only for the PUBLIC group. Requires DBA permissions to set the option. Setting takes effect immediately and persists across server restarts.

MPX_HEARTBEAT_FREQUENCY Option

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

Scope
Can be set only for the PUBLIC group. Requires DBA permissions to set the option. Setting takes effect immediately and persists across server restarts.

MPX_IDLE_CONNECTION_TIMEOUT Option

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

Scope
Can be set only for the PUBLIC group. Requires DBA permissions to set the option. Setting takes effect immediately and persists across server restarts.

MPX_MAX_CONNECTION_POOL_SIZE Option

Maximum number of connections allowed in the connection pool on a secondary node.

Allowed Values
1 to 1000

Default
10

Scope

Can be set only for the PUBLIC group. Requires DBA permissions to set the option. Setting takes effect immediately and persists across server restarts.

Description

INC connections are inter-server connections between secondary nodes and the coordinator node. An INC connection is associated with each user connection on a secondary server doing a DDL or read-write operation. The connection is active until that command commits or rolls back; it then returns to the pool. If these transactions are short lived, then the default setting of `MPX_MAX_CONNECTION_POOL_SIZE` suffices for many user connections running DDL or RW operations. If many concurrent connections run DDL or read-write operations, or the transactions take a long time, increase the value of `MPX_MAX_CONNECTION_POOL_SIZE`. For example, increase the value when many user connections do concurrent loads without committing.

Exceeding `MPX_MAX_CONNECTION_POOL_SIZE` returns `SQL Anywhere Error -1004000: The number of connections in the connection pool have exceeded the upper limit`.

To estimate the pool size required, consider the setting of the `-gm` server option. The `-gm` setting indicates how many users can connect to the secondary server; the INC connections are not included, but will add to this number. Use application requirements to assess how many read-write or DDL operations are likely to occur per user, and increase the pool size accordingly.

Each connection (INC or user) carries a memory overhead depending on `-gn` setting and number of cores. The burden of memory and thread contention may affect Sybase IQ server response times. See *Performance and Tuning Guide > Managing System Resources > Optimize Memory Use > Determine the Sizes of the Buffer Caches > Memory Overhead.*

See also

- *Pooled Connections* on page 8
- *Multiple Writers for Parallel Loads* on page 5

MPX_MAX_UNUSED_POOL_SIZE Option

Maximum number of unused connections in the connection pool on a secondary node.

Allowed Values

0 to maximum pool size

Default

0

Scope

Can be set only for the PUBLIC group. Requires DBA permissions to set the option. Setting takes effect immediately and persists across server restarts.

See also

- *Pooled Connections* on page 8
- *Multiple Writers for Parallel Loads* on page 5

MPX_WORK_UNIT_TIMEOUT Option

Time, in seconds, before a multiplex DQP leader reassigns incomplete distributed work to another DQP worker node.

Allowed Values

0 to 3600 seconds.

DQP work units are typically sized to span only a few seconds. If a worker node goes offline or experiences an unusually high workload, DQP work previously assigned to that worker node is reassigned to another node after the given timeout.

Default

60 seconds

Typically you do not need to change this option from its default value. However, increase this option in rare cases where a query has very large intermediate results that cause individual work units to time out.

Decrease this option if unreliable networks or servers cause distributed work to be lost and the timeout interval is unacceptably long. Note that setting this option too low can cause unnecessary early timeouts.

Scope

Can be set temporary for an individual connection or persistently across server restarts. Requires DBA permissions to set the option. Setting takes effect immediately.

System Tables

Certain system tables support multiplex functionality.

ISYSIQINFO System Table

This table indicates the database characteristics as defined when the Sybase IQ database was created using **CREATE DATABASE**. It always contains only one row.

Column name	Column type	Description
last_full_backup	TIMESTAMP	The completion time of the most recent backup.
last_incr_backup	TIMESTAMP	The completion time of the most recent incremental backup.
create_time	TIMESTAMP NOT NULL	The date and time created.
update_time	TIMESTAMP NOT NULL	The date and time of the last update.
file_format_version	UNSIGNED INT NOT NULL	The file format number of files for this database.
cat_format_version	UNSIGNED INT NOT NULL	The catalog format number for this database.
sp_format_version	UNSIGNED INT NOT NULL	The stored procedure format number for this database.
block_size	UNSIGNED INT NOT NULL	The block size specified for the database.
chunk_size	UNSIGNED INT NOT NULL	The number of blocks per chunk as determined by the block size and page size specified for the database.
file_format_date	CHAR(10) NOT NULL	The date when file format number was last changed.
dbsig	BINARY(136) NOT NULL	Used internally by catalog.
multiplex_name	CHAR(128) NULL	Used internally by catalog.

Column name	Column type	Description
last_multiplex_mode	TINYINT NULL	The 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.

Constraint: Primary key(create_time)

ISYSIQLOGICALSERVER System Table

ISYSIQLOGICALSERVER stores logical server and the correspondence between logical server and associated logical server policy information.

See also

- *SYSIQLOGICALSERVER System View* on page 109

ISYSIQLOGINPOLICYLSINFO System Table

ISYSIQLOGINPOLICYLSINFO stores the login policy logical server assignment information.

See also

- *SYSIQLOGINPOLICYLSINFO System View* on page 110

ISYSIQLSLOGINPOLICYOPTION System Table

ISYSIQLSLOGINPOLICYOPTION stores the login policy option values that have logical server level settings.

See also

- *SYSIQLSPOLICYOPTION System View* on page 113

ISYSIQLSMEMBER System Table

ISYSIQLSMEMBER stores the logical server membership information.

See also

- *SYSIQLSMEMBER System View* on page 110

ISYSIQLSPOLICY System Table

ISYSIQLSPOLICY stores logical server policies.

See also

- *SYSIQLSPOLICY System View* on page 114

ISYSIQLSPOLICYOPTION System Table

ISYSIQLSPOLICYOPTION stores the logical server policy options.

ISYSIQMPXSERVER System Table

ISYSIQMPXSERVER stores membership properties and version status data for a given multiplex node.

See also

- *SYSIQMPXSERVER System View* on page 113

System Views

Certain system views support multiplex functionality.

SYSIQLOGICALSERVER System View

Presents a readable version of the ISYSIQLOGICALSERVER system table.

The ISYSIQLOGICALSERVER system table stores logical server information and associated logical server policy information.

Column name	Column type	Description
ls_id	UNSIGNED BIGINT NOT NULL	The ID number of the logical server.
ls_object_id	UNSIGNED BIGINT NOT NULL	The logical server object ID number.
ls_policy_id	UNSIGNED BIGINT NOT NULL	The ID number of the logical server policy.
ls_name	CHAR(128) NOT NULL UNIQUE	The logical server name.

Constraints on underlying system table:

- Primary key(ls_id)
- object_id foreign key(ISYSOBJECT)
- ls_policy_id foreign key(ISYSIQLSPOLICY)

See also

- *ISYSIQLOGICALSERVER System Table* on page 108

SYSIQLOGINPOLICYLSINFO System View

Presents a readable version of the table ISYSIQLOGINPOLICYLSINFO.

The ISYSIQLOGNPOLICYLSINFO system table stores the login policy logical server assignment information.

Column name	Column type	Description
login_policy_id	UNSIGNED BIGINT NOT NULL	The ID number of the login policy.
ls_id	UNSIGNED BIGINT NOT NULL	The ID number of the logical server.

Constraints on underlying system table:

- Primary key(login_policy_id, ls_id)
- login_policy_id foreign key(ISYSLOGINPOLICY)
- ls_id foreign key(ISYSIQLOGICALSERVER)

See also

- *ISYSIQLOGINPOLICYLSINFO System Table* on page 108

SYSIQLSMEMBER System View

Presents a readable version of the table ISYSIQLSMEMBER.

`ISYSIQLSMEMBER` stores the logical servers and their corresponding multiplex servers.

Column name	Column type	Description
ls_id	UNSIGNED BIGINT NOT NULL	The ID number of the logical server.
logical_membership_type	TINYNT NOT NULL	The type of the logical membership.
mpx_server_id	UNSIGNED INT NOT NULL	The ID number of the multiplex server.

Column name	Column type	Description
membership_info	UNSIGNED INT NOT NULL	The membership information.

Constraints on underlying system table:

- Primary key(ls_id, logical_membership_id, mpx_server_id)
- ls_id foreign key(ISYSIQLOGICALSERVER)

For logical server memberships that are defined using the multiplex server name, the value of logical_membership_type is 0 and mpx_server_id is the server id of the multiplex server.

For the logical membership of the coordinator, mpx_server_id is 0 and logical_membership_type is 1.

See also

- *ISYSIQLSMEMBER System Table* on page 108

SYSIQLSMEMBERS Consolidated System View

Presents logical server membership information from the ISYSIQLSMEMBERS system table in a readable format.

Column name	Column type	Description
ls_id	UNSIGNED BIGINT NOT NULL	The ID number of the logical server.
ls_name	CHAR(128) NOT NULL	The name of the logical server.
server_id	UNSIGNED INT NOT NULL	The multiplex server identifier of the member, for the membership defined using server name, or 0, for the logical membership of the coordinator.
server_name	CHAR(128) NOT NULL	The multiplex server name of the member for the membership defined using the server name, or 'LOGICAL COORDINATOR' for the logical membership of the coordinator.
membership_type	TINYINT NOT NULL	0 for the membership defined using the server name, or 1 for the logical membership of the coordinator.

SYSIQLSLOGINPOLICIES System View

Presents a readable version of information for logical server assignments from the login policies.

This is a consolidated system view that shows information from SYSIQLOGICALSERVER, ISYSIQLOGINPOLICYLSINFO and ISYSLOGINPOLICY.

Column name	Column type	Description
ls_id	UNSIGNED BIGINT NOT NULL	Logical server identifier.
ls_name	CHAR(128)	Logical server name.
login_policy_id	UNSIGNED BIGINT NOT NULL	The ID number of the login policy.
login_policy_name	char(128)	The name of the login policy.

SYSIQLSLOGINPOLICYOPTION System View

Presents a version of the table ISYSIQLSLOGINPOLICYOPTION in a readable format.

The ISYSIQLSLOGINPOLICYOPTION table stores the logical server level settings for login policy option values.

Column name	Column type	Description
login_policy_id	UNSIGNED BIGINT NOT NULL	The ID number of the login policy.
ls_id	UNSIGNED BIGINT NOT NULL	Logical server identifier.
login_option_name	CHAR(128) NOT NULL	The name of the login policy option.
login_option_value	LONG VARCHAR NOT NULL	The value of the login policy option.

Constraints on underlying system table:

- Primary key(login_policy_id,ls_id, login_option_name)
- login_policy_id foreign key(ISYSLOGINPOLICY)
- ls_id foreign key(ISYSIQLOGICALSERVER)

SYSIQLSPOLICYOPTION System View

Presents a version of the table ISYSIQLSPOLICYOPTION in a readable format.

The ISYSIQLSPOLICYOPTION table stores the logical server policy options.

Column name	Column type	Description
ls_policy_id	UNSIGNED BIGINT NOT NULL	The ID number of the login policy.
ls_policy_option_name	CHAR(128) NOT NULL	The logical server policy option name.
ls_policy_option_value	LONG VARCHAR NOT NULL	The logical server policy option value.

Constraints on underlying system table:

- Primary key(ls_policy_id, ls_policy_option_name)
- ls_policy_id foreign key(ISYSIQLSPOLICY)

See also

- *ISYSIQLSLOGINPOLICYOPTION System Table* on page 108

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).

Column name	Column type	Description
connection_info	LONG VARCHAR NULL	String containing host name and port pairs for public domain connections, delimited by semicolons.
db_path	LONG VARCHAR NOT NULL	Full path to the database file for the server.
private_connection_info	LONG VARCHAR NULL	String containing host name and port pairs for private network connections, delimited by semicolons.

Constraints on underlying system table:

- Primary key(server_id)

See also

- *ISYSIQMPXSERVER System Table* on page 109

SYSIQLSPOLICY System View

Presents a version of the table ISYSIQLSPOLICY in a readable format.

The ISYSIQLSPOLICY system table stores the logical server policies.

Column name	Column type	Description
ls_policy_Id	UNSIGNED BIGINT NOT NULL	The ID number of the logical server policy.
ls_policy_name	CHAR(128) NOT NULL UNIQUE	The logical server policy name.

Constraints on underlying system table:

- Primary key(ls_policy_id)
- object_id foreign key(ISYSOBJECT)

See also

- *ISYSIQLSPOLICY System Table* on page 109

System Procedures

Certain 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

Checks validity of the current database. Optionally corrects allocation problems for dbspaces or databases. 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 to run the stored procedure.

sp_iqconnection Procedure

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 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:

Table 18. 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");
```

See SQL Anywhere *Server – Programming*.

Table 19. 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.

Column name	Description
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.
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.

Column name	Description
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.
LSName	The logical server name of the connection. NULL if logical server context is unknown or not applicable.

Example

The following is an example of **sp_iqconnection** output:

```
ConnHandle      Name    Userid                 LastReqTime   ReqType
=== =================== ======  ==========================   =======
 1  'SQL_DBC_100525210' 'DBA'   '2011-03-28  09:29:24.466'  'OPEN'


           IQCmdType    LastIQCmdTime      IQCursors  LowestIQCursorState
====================    ==============     =========  ===================
'IQUTILITYOPENCURSOR'  2011-03-28 09:29:24.0       0             'NONE'


IQthreads   TxnID          ConnCreateTime TempTableSpaceKB TempWorkSpaceKB
=========  =======  =====================  =============== ===============
        0  3352568  2011-03-28 09:29:20.0                0           0


IQconnID satoiq_count iqtosa_count CommLink NodeAdd LastIdle MPXServerName LSName
======== ============ ============ ======== ======= ======== ============  ============
      34           43          2  'local'      ''      244  (NULL)        Finance_LS
```

sp_iqdbsize Procedure

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 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 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 to run the stored procedure.

sp_iqindexinfo Procedure

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 to run the stored procedure.

sp_iqmpxincconnpoolinfo Procedure

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 to run the stored procedure.

Description

The **sp_iqmpxincconnpoolinfo** procedure returns:

Table 20. 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_iqmpxincconnpoolinfo**:

```
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_iqmpxcheckdqpconfig Procedure

sp_iqmpxcheckdqpconfig is a diagnostic tool that checks the DQP configuration for the current connection. If DQP fails, run **sp_iqmpxcheckdqpconfig** to determine if DQP configuration issues are causing the query distribution failure.

Syntax

```
sp_iqmpxcheckdqpconfig
```

Permissions

No special privileges are required to execute the procedure.

Description

Table 21. Column Descriptions

Column Name	Description
DiagMsgID	Uniquely identifies a diagnostic message

Column Name	Description
Description	Diagnostic message describing the issue found with DQP configuration

Table 22. Diagnostic Messages

DiagMsgID	Description
0	`No issues found with DQP configuration`
1	`Database is a simplex`
2	`Multiplex is running in single-node configuration mode`
3	`Login policy option dqp_enabled is set to OFF`
4	`Temporary dqp_enabled connection option is set to OFF`
5	`Logical server context has only one member node`
6	`Coordinator does not participate in DQP since its named membership in the logical server is currently ineffective`
7	`Coordinator does not participate in DQP since its logical membership in the logical server is currently ineffective because ALLOW_COORDINATOR_AS_MEMBER option in Root Logical server policy set to OFF`
8	`There is no dbfile in IQ_SHARED_TEMP dbspace`
9	`All dbfiles in IQ_SHARED_TEMP dbspace are READ ONLY`

DiagMsgID	Description
10	IQ_SHARED_TEMP dbspace is dynamically offline

Example

Sample output from the **sp_iqmpxcheckdqpconfig** procedure:

```
diagmsgid           description
3                   Login policy option dqp_enabled is set to OFF
5                   Logical server context has only one member node
6                   Coordinator does not participate in DQP since its
                    named membership in the logical server is
                    currently ineffective
7                   Coordinator does not participate in DQP since
                    its logical membership in the logical server
                    is currently ineffective because
                    ALLOW_COORDINATOR_AS_MEMBER option in Root
                    Logical server policy set to OFF
8                   There is no dbfile in IQ_SHARED_TEMP dbspace
```

sp_iqmpxdumptlvlog Procedure

Returns the contents of the table version log in a readable format.

Syntax

```
sp_iqmpxdumptlvlog
[main], [asc | desc]
```

Permissions

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

Description

sp_iqmpxdumptlvlog returns the contents of the queue through which the coordinator propagates DML and DDL commands to secondary nodes.

The **asc** or **desc** arguments specify the row order. These arguments require the **main** argument. The default options are:

```
'main', 'asc'.
```

Examples

This example shows the output of **sp_iqmpxdumptlvlog**:

```
RowID     Contents
--------------------------------------------------------------
1         Txn CatId:196 CmtId:196 TxnId:195 Last Rec:1
          UpdateTime: 2011-08-08 15:41:43.621
2         Txn CatId:243 CmtId:243 TxnId:242 Last Rec:5
```

```
          UpdateTime: 2011-08-08 15:42:25.070
3         DDL: Type=34, CatID=0, IdxID=0,
          Object=IQ_SYSTEM_TEMP, Owner=mpx4022_w1
4         CONN: CatID=0, ConnUser=
5         SQL: ALTER DBSPACE "IQ_SYSTEM_TEMP" ADD FILE
          "w1_temp1" '/dev/raw/raw25' FILE ID 16391 PREFIX 65536
          FINISH 0 FIRST BLOCK
1         BLOCK COUNT 3276792 RESERVE 0 MULTIPLEX SERVER
          "mpx4022_w1" COMMITID 242 CREATETIME
          '2011-08-08 15:42:24.860'
6         Txn CatId:283 CmtId:283 TxnId:282 Last Rec:7
          UpdateTime: 2011-08-08 15:42:50.827
7         RFRB TxnID: 242 CmtID:243 ServerID 0 BlkmapID:
          0d00000000000000d2000a0000000000020000000000000000000
          000000000000000000000800350101000000c3800000000000000
          0100000000000000000000000RFID:010005010000000013000000
          0000000000100000000000100RBID:0100050100000000130000
```

sp_iqmpxfilestatus Procedure

If run on the coordinator node, displays file status for coordinator and for every shared dbspace file on every included secondary node. If executed on a secondary node, displays file status for only the current node.

Syntax

```
sp_iqmpxfilestatus
```

Permissions

Must have DBA, MULTIPLEX ADMIN, or SPACE ADMIN authority. Users without these authorities must be granted EXECUTE permission to run the stored procedure.

Description

sp_iqmpxfilestatus returns:

Table 23. sp_iqmpxfilestatus columns

Column name	Data type	Description
ServerID	unsigned int	Identifier for the multiplex server, from SYSIQMPXINFO
DBSpaceName	char(128)	Dbspace from which the space is reserved
FileName	char(128)	Logical file name of the dbspace file

Column name	Data type	Description
FileStatus	char(2)	Dbspace file status: • VALID – file path and permissions are correct • INVALID_PATH – path name not accessible • INVALID_PERM – file permissions are incorrect

Example

This example shows sample output of **sp_iqmpxfilestatus**:

```
server_id,server_name,DBSpace_name,FileName,FileStatus
1,'mpx2422_m','IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','VALID'
1,'mpx2422_m','mpx_main1','mpx_main1','VALID'
1,'mpx2422_m','IQ_SHARED_TEMP','sharedfile_dba','VALID'
1,'mpx2422_m','IQ_SHARED_TEMP','sharedfile_dba1','VALID'
2,'mpx2422_w1','IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','VALID'
2,'mpx2422_w1','mpx_main1','mpx_main1','VALID'
2,'mpx2422_w1','IQ_SHARED_TEMP','sharedfile_dba','VALID'
2,'mpx2422_w1','IQ_SHARED_TEMP','sharedfile_dba1','VALID'
3,'mpx2422_r1','IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','VALID'
3,'mpx2422_r1','mpx_main1','mpx_main1','VALID'
3,'mpx2422_r1','IQ_SHARED_TEMP','sharedfile_dba','VALID'
3,'mpx2422_r1','IQ_SHARED_TEMP','sharedfile_dba1','VALID'
```

sp_iqmpxincheartbeatinfo Procedure

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 to run the stored procedure.

Description

The **sp_iqmpxincheartbeatinfo** procedure returns the following:

Table 24. 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

Displays complete multiplex configuration info for every node in the multiplex. Can run on coordinator or secondary nodes.

Note: Users with RESOURCE authority cannot 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 to run the stored procedure.

Users must be licensed for the Multiplex Grid Option to run secondary nodes.

Description

The **sp_iqmpxinfo** procedure returns the following:

Table 25. sp_iqmpxinfo columns

Column name	Data type	Description
server_id	unsigned int	Identifier for the server for which information appears.
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 TCP/IP 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' | '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.
private_connection_info	long varchar	A formatted string containing the host/port portion of the connection string used for private TCP/IP connections between multiplex servers.
mipc_priv_state	char(16)	'active' – MIPC connection to this node is active over the private interconnect | 'not responding' – MIPC connection to this node is not responding over private interconnect.

Column name	Data type	Description
mipc_public_state	char(16)	'active' – MIPC connection to this node is active over the public interconnect. | 'not responding' – MIPC connection to this node is not responding over public interconnect.

Example

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,private_connection_
info,mipc_priv_state,mipc_public_state

1,'my_mpx1','host=(fe80::214:4fff:fe45:be26%2):1362
0,(fd77:55d:59d9:329:214:4fff:fe45:be2
6%2):13620,10.18.41.196:13620','/system3/users
/devices/s16900269/iqmpx1/mpx1.db',
'coordinator','included','coordinator','N/A',
'my_mpx2',0,,,'active','active'

2,'IQ_mpx2','host=system3:13625',
'/system3/users/devices/s16900269
/iqmpx_2/wk0001.db','writer','included',
'writer','active','IQ_mpx20', 'not responding','active'

3,'IQ_mpx3,'host=system3:13630/system3/users/devi
ces/s16900269/iqmpx_3/mpx1.db','reader','included',
'unknown',timed out',
'IQ_mpx20','not responding',
'not responding'
```

See also

- *Checking Server Status in Sybase Central* on page 23
- *Designated Failover Node* on page 37
- *Designating Failover Node with Sybase Central* on page 34
- *Designating Failover Node with Interactive SQL* on page 20
- *Checking Server Status in Interactive SQL* on page 18

sp_iqmpxvalidate Procedure

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

Shows the current version information for this server, including 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 26. 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

Column	Data type	Description
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_iqsharedtempdistrib Procedure

Shows the current shared temp space usage distribution. If run from the coordinator node displays share temp space distribution for all nodes. If run from a secondary node displays shared temp space usage for that node.

Shared temporary space is reserved for each node in the multiplex on demand. Space is reserved for a node in an allocation unit. Nodes can have multiple allocation units reserved based on their dynamic space demands. Allocation units are leased to allow nodes to use more space as needed and return the space to a global pool when not needed. Allocation units expire when space usage decreases and their lease time ends, or when a server shuts down.

Syntax

```
sp_iqsharedtempdistrib()
```

Permissions

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

Description

Column	Data type	Description
ServerID	unsigned bigint	Server ID of the multiplex server, from SYSIQMPXINFO
DBSpaceName	char(128)	Name of the dbspace from which space is reserved

Column	Data type	Description
UnitType	char(10)	Type of allocation unit. Valid values are: • Active – currently reserved and in use by the node. • Expired – reserved for the node but are in transition back to the global space pool. • Quarantined – reserved for the node but quarantined due to node failure.
VersionID	unsigned bigint	Version ID of the unit. For active units, the version when the unit was reserved for the node. For expired units, the version when the unit was expired. For quarantined units, the version when the unit was quarantined.
NBlocks	unsigned bigint	Number of outstanding blocks in the unit

sp_iqspaceinfo Procedure

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 to run the stored procedure.

sp_iqspaceused Procedure

Shows information about space available and space used in the IQ store, IQ temporary store, and IQ global and local shared temporary stores.

Syntax

```
sp_iqspaceused(out mainKB             unsigned bigint,
               out mainKBUsed        unsigned bigint,
               out tempKB            unsigned bigint,
               out tempKBUsed        unsigned bigint,
               out shTempTotalKB     unsigned bigint,
               out shTempTotalKBUsed unsigned bigint,
               out shTempLocalKB     unsigned bigint,
               out shTempLocalKBUsed unsigned bigint)
```

Permissions

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

Usage

sp_iqspaceused returns eight values as unsigned bigint out parameters. This system stored procedure can be called by user-defined stored procedures to determine the amount of main and temporary IQ store space in use.

Description

sp_iqspaceused returns a subset of the information provided by **sp_iqstatus**, but allows the user to return the information in SQL variables to be used in calculations.

Table 27. sp_iqspaceused columns

Column name	Description
mainKB	The total IQ main store space in kilobytes.
mainKBUsed	The number of kilobytes of IQ main store space used by the database. (Secondary multiplex nodes return '(Null)'.)
tempKB	The total IQ temporary store space in kilobytes.
tempKBUsed	The total IQ temporary store space in kilobytes.
shTempTotalKB	The total IQ global shared temporary store space in kilobytes.
shTempTotalKBUsed	The total IQ global shared temporary store space in kilobytes. (Secondary multiplex nodes return '(Null)'.)
shTempLocalKB	The total IQ local shared temporary store space in kilobytes.
shTempLocalKBUsed	The number of kilobytes of IQ local shared temporary store space in use by the database.

Example

sp_iqspaceused requires eight output parameters. The following example shows the creation of a user-defined stored procedure **myspace** that declares the eight output parameters and then calls **sp_iqspaceused**:

```
create procedure dbo.myspace()
begin
  declare mt unsigned bigint;
  declare mu unsigned bigint;
  declare tt unsigned bigint;
  declare tu unsigned bigint;
  declare gt unsigned bigint;
  declare gu unsigned bigint;
  declare lt unsigned bigint;
  declare lu unsigned bigint;
  declare tt_t unsigned bigint;
  declare mt_t unsigned bigint;
  declare gt_t unsigned bigint;
  declare lt_t unsigned bigint;
  call sp_iqspaceused(mt,mu,tt,tu,gt,gu,lt,lu);
  if (tt = 0) then
        set tt_t = 0;
     else
        set tt_t = tu*100/tt;
     end if;
     if (mt = 0) then
        set mt_t = 0;
     else
        set mt_t = mu*100/mt;
     end if;
     if (gt = 0) then
        set gt_t = 0;
     else
        set gt_t = gu*100/gt;
     end if;
     if (lt = 0) then
        set lt_t = 0;
     else
        set lt_t = lu*100/lt;
     end if;
   select cast(mt/1024 as unsigned bigint) as mainMB,
         cast(mu/1024 as unsigned bigint) as mainusedMB,
        mu*100/mt as mainPerCent,
         cast(tt/1024 as unsigned bigint) as tempMB,
         cast(tu/1024 as unsigned bigint) as tempusedMB,
        tu*100/tt as tempPerCent,
         cast(gt/1024 as unsigned bigint) as shTempTotalKB,
         cast(gu/1024 as unsigned bigint) as shTempTotalKBUsed,
        gu*100/gt as globalshtempPerCent,
         cast(lt/1024 as unsigned bigint) as shTempLocalMB,
         cast(lu/1024 as unsigned bigint) as shTempLocalKBUsed,
        lu*100/lt as localshtempPerCent;
end
```

To display the output of **sp_iqspaceused**, run the procedure **myspace**:

```
myspace
```

sp_iqstatus Procedure

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.

Permissions

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

See also

- *sp_iqversionuse Procedure* on page 136

sp_iqtransaction Procedure

Shows information about transactions and versions.

Syntax

```
sp_iqtransaction
```

Permissions

DBA authority required. Users without DBA authority must be granted `EXECUTE` permission 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 28. sp_iqtransaction columns

Column name	Description
Name	The name of the server.

Column name	Description
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.

Column name	Description
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**.
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 internode 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

Shows which versions of servers are in use and by which users.

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.

Permissions

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

See also

- *sp_iqstatus Procedure* on page 133

Example

Output from **sp_iqversionuse** differs on coordinator and secondary servers.

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. 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
42715	'mpxq'	00	1	0	0

Startup and Database Administration Utilities

Certain command-line 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)

Run start_iq at the command line to customize your server startup.

Use server startup switches with the **start_iq** command to configure Sybase IQ multiplex servers at startup.

Table 29. 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.
-iqmpx_sn	1	Runs the current multiplex node in single-node mode. Use single-node mode only to fix problems with the multiplex configuration. You must shut down all other nodes in the multiplex. Sybase recommends that you use single-node mode only on the coordinator.
-iqmpx_reclaimwriter-freelist	server name	This option applies only while restarting a coordinator node. The coordinator forcefully reclaims the free list of the writer node identified by the server name. Use this switch only when a writer fails and cannot be restarted.
-iqmsgnum *num*	0-64 (inclusive)	Specifies the number of message log 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 nonzero. The **IQMsgNumFiles** server property corresponds to -iqmsgnum and takes precedence over the value of **-iqmsgnum**. If the value is not set, the default minimum pool size is `MIN (MAX (4, number of cores/4) , mipc-maxt (if set)).`
-iqmsgsz *size*	integers 0-2047 inclusive, in MB.	Limits the maximum size of the message log. The default value is 0, which specifies no limit on the size of the message file.
-mipcmint *size*	integers 0-256 inclusive	Specifies the minimum number of threads allowed in the shared thread pool for MIPC request handling. Each new MIPC server connection adds two threads to the pool. The value of **-mipcmint** defaults to 0 and cannot exceed the **-mipcmaxt** value. Set this value only if advised to do so by Sybase Technical Support. If the value is not set, the default minimum pool size is `MIN (MAX (4, number of cores/4), mipcmaxt (if set))` .

Startup switch	Values	Description
-mipcmaxt *size*	integers 0-256 inclusive	Specifies the maximum number of threads allowed in the shared thread pool for MIPC request handling. Each new MIPC server connection adds two threads to the pool. The value of **-mipcmaxt** defaults to 0 and must exceed the **-mipcmint** value. Set this value only if advised to do so by Sybase Technical Support. If the value is not set, the default maximum pool size is `MAX (number of cores, mipcmint)` .

For additional switches, see *Sybase IQ Utility Guide > start_iq Database Server Startup Utility > start_iq Server Options*.

Note: The **-iqmc** and **-iqtc** switches allow different cache sizes for each node in a multiplex, but this may have adverse affects. For example, if a node worker is configured with a much smaller cache than the leader, hash joins on the leader will operate in a paging mode that disallows parallelism.

See also

- *Tables in Join Indexes* on page 43

Index