SYBASE®
An SAP® Company

Administration: Multiplex

SAP Sybase IQ 16.0

DOCUMENT ID: DC01839-01-1600-01
LAST REVISED: February 2013

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

Contents

Multiplex Overview

Administering SAP® 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 manage workloads across multiple nodes using SAP Sybase IQ multiplex capability.

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

Introduction to SAP Sybase IQ Multiplex

SAP 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 Sybase Control Center 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.

SAP Sybase IQ Multiplex Architecture

SAP 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.

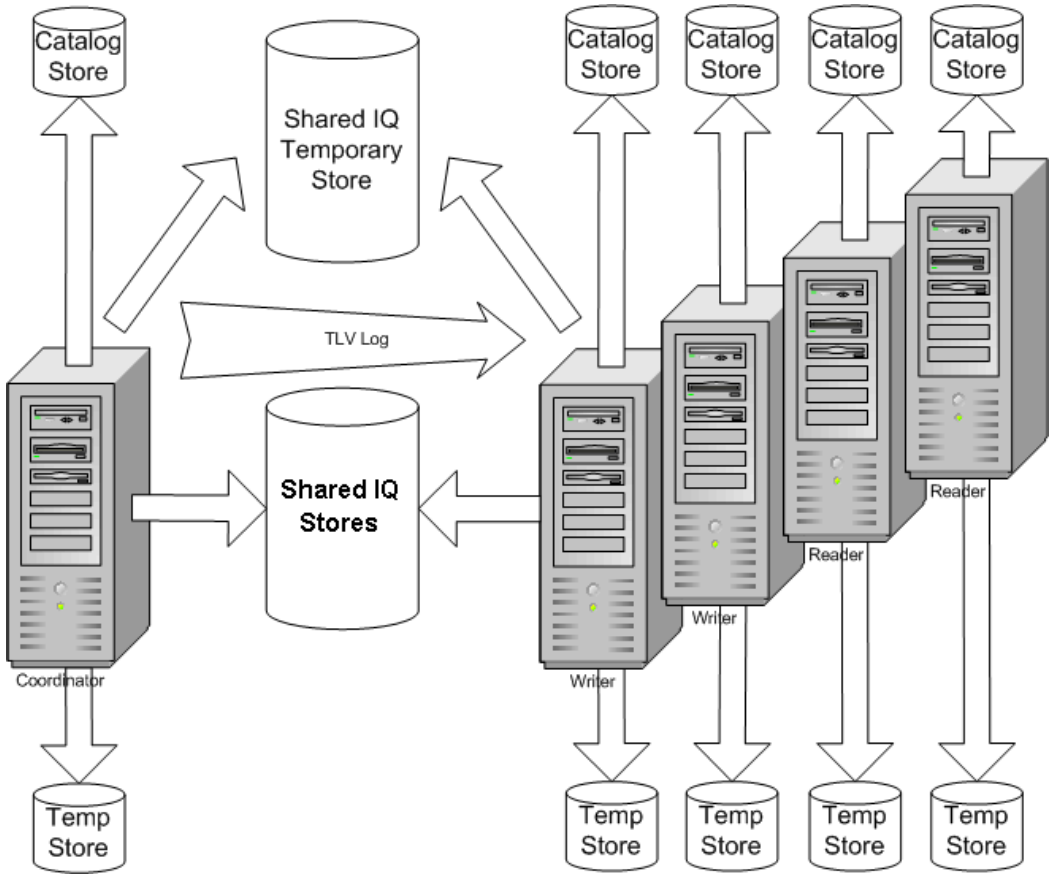
The shared IQ store and shared temporary store are common to all servers.

Multiplex Architecture Diagram

SAP 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



See also

- *Data Storage* on page 3
- *Multiplex Configuration* on page 4
- *Files on Shared Disk Arrays* on page 6
- *Communication Infrastructure* on page 7

Data Storage

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.

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

Table 1. Dbspace management

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

Data managed by SQL Anywhere is not shared. Each node requires a separate copy of such data. *Introduction to Sybase IQ* describes the distinction between data managed solely by SAP Sybase IQ and data that is managed by underlying SQL Anywhere® software.

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

- *Multiplex Architecture Diagram* on page 2
- *Multiplex Configuration* on page 4
- *Files on Shared Disk Arrays* on page 6
- *Communication Infrastructure* on page 7
- *UNIX or Linux Shared Array Path Definitions* on page 6
- *Windows Shared Disk Array Path Definitions* on page 7

Multiplex Configuration

SAP 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.

  SAP 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.

See also

- *Multiplex Architecture Diagram* on page 2
- *Data Storage* on page 3
- *Files on Shared Disk Arrays* on page 6
- *Communication Infrastructure* on page 7

Coordinator Node

Each multiplex configuration has one and only one *coordinator node* that is capable of read-only and read-write operations and is also responsible for bookkeeping to manage other nodes.

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

All nodes in a multiplex configuration are secondary nodes except the coordinator.

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.

Files on Shared Disk Arrays

SAP 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.

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

- *Multiplex Architecture Diagram* on page 2
- *Data Storage* on page 3
- *Multiplex Configuration* on page 4
- *Communication Infrastructure* on page 7
- *UNIX or Linux Shared Array Path Definitions* on page 6
- *Windows Shared Disk Array Path Definitions* on page 7
- *Adding Space to IQ_SYSTEM_MAIN on a Coordinator* on page 38
- *Synchronizing Servers* on page 24

UNIX or Linux Shared Array Path Definitions

On UNIX or Linux, access shared disks through absolute paths or soft links, as shown in these examples.

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 12

Windows Shared Disk Array Path Definitions

On Windows, access shared disks through absolute paths or the Disk Physical number, as shown in these examples.

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 12

Communication Infrastructure

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

See also

- *Multiplex Architecture Diagram* on page 2
- *Data Storage* on page 3
- *Multiplex Configuration* on page 4
- *Files on Shared Disk Arrays* on page 6

Internode Communication (INC)

INC provides transactional communication between coordinator and secondary nodes.

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 internode 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`.

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 120
- *MPX_MAX_UNUSED_POOL_SIZE Option* on page 121

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*.

Create Multiplex Servers

Convert a single SAP Sybase IQ server into a multiplex.

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

Planning the Configuration

Before you create a multiplex, consider hardware and software requirements.

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

See also

- *Main Store Requirements* on page 12
- *Hardware Requirements* on page 12
- *Java External Environment in a Multiplex* on page 13
- *Multiplex Planning Worksheet* on page 13

Setting Up Windows Access to Raw Devices

Windows systems restrict raw device access to user accounts with Administrator privilege.

To run the SAP Sybase IQ servers using an account that lacks Administrator privilege, you must enable new device access permissions for that account after each system reboot.

The **rawaccedit** utility sets permissions for devices for the current session.

Set up read-write access for the write servers and read access for query servers.

1. Type the following at a command prompt:
   ```
   rawaccedit
   ```
2. In the IQ Raw Device Access window, type the name of the user and the device to which you want to grant access.

You can use Alt+N to tab to the User's Name box and Alt+D to tab to the Raw Device Name box.

To specify...	Type...
An unpartitioned raw device	Type the physical drive number. Unpartitioned drives are named `\\.\PhysicalDriveN`, where N is a number starting with 0. To find the physical drive numbers, Run **Accessories > System Tools > System Information**.
A partitioned raw device	Type the letter assigned to that partition.

3. Click **Add**.
4. Correct any errors in the user name and device name that display in the top panel and click **Update ACL and Exit**.

Device access permissions remain until you reboot Windows.

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.

Migrate your database from an earlier release of SAP Sybase IQ to the current release.

See also

- *Multiplex Storage Requirements* on page 11
- *Hardware Requirements* on page 12
- *Java External Environment in a Multiplex* on page 13
- *Multiplex Planning Worksheet* on page 13
- *UNIX or Linux Shared Array Path Definitions* on page 6
- *Windows Shared Disk Array Path Definitions* on page 7

Hardware Requirements

An SAP 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 SAP Sybase IQ installed. For upgrade instructions, see the *Migration Guide* for your platform..

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.

SAP Sybase IQ does not support:

- Multiplexes of SAP 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.

See also

- *Multiplex Storage Requirements* on page 11
- *Main Store Requirements* on page 12
- *Java External Environment in a Multiplex* on page 13
- *Multiplex Planning Worksheet* on page 13

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 or the Interactive SQL **INSTALL JAVA** statement to install the Java class file and JAR.

See also

- *Multiplex Storage Requirements* on page 11
- *Main Store Requirements* on page 12
- *Hardware Requirements* on page 12
- *Multiplex Planning Worksheet* on page 13

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. SAP 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

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.	
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.	

See also

- *Multiplex Storage Requirements* on page 11
- *Main Store Requirements* on page 12
- *Hardware Requirements* on page 12
- *Java External Environment in a Multiplex* on page 13

Converting Databases to Multiplex

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 an SAP 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{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
   '<database path>/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{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.

Multiplex Database Files

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

Table 3. Contents of multiplex database directories

Folder, Directory, or File-name	Purpose
`dbname`.db	File that contains the catalog store. This cannot be a raw device.
`dbname`.iqmsg	File that contains messages from SAP 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.

Manage Multiplex Servers

You can use Interactive SQL and command line utilities to manage multiplex servers.

Enter Interactive SQL statements individually at the command line or run them in scripts to query, modify, or load Sybase servers, including SAP Sybase IQ.

Examples in this document show SQL statements and utility command lines. If you prefer a graphical interface, manage and monitor your multiplex with Sybase® Control Center. For details, see the Sybase Control Center for SAP Sybase IQ online help in SCC or at *http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680*.

Multiplex System Privileges

Two specific system privileges are required to perform authorized tasks in a multiplex environment.

ACCESS SERVER LS System Privilege

Allows logical server connection using the SERVER logical server context.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the WITH NO ADMIN OPTION clause is used by default.

MANAGE MULTIPLEX System Privilege

Allows administrative tasks related to multiplex server management.

The MANAGE MULTIPLEX system privilege allows a user to:

- Issue multiplex-related CREATE, ALTER, DROP, or COMMENT statements on logical server policies
- Issue multiplex-related CREATE, ALTER, DROP, or COMMENT statements on logical servers
- Perform exclusive assignment of a dbspace to logical servers
- Release a populated dbspace from the exclusive use of a logical server

Note: The MANAGE MULTIPLEX system privilege also manages failover configurations, and is required for a manual failover.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the WITH NO ADMIN OPTION clause is used by default.

DBO System Role in a Multiplex Environment

By default, the DBO system role is granted the SYS_AUTH_DBA_ROLE compatibility role, ensure that the DBO system role is granted all privileges necessary to execute multiplex management stored procedures.

If you use the **ALTER ROLE** statement to migrate the SYS_AUTH_DBA_ROLE compatibility role to a new user-defined role, the new role is automatically granted to the DBO system role, provided that SYS_AUTH_DBA_ROLE has not been revoked from the DBO system role.

The SYS_AUTH_DBA_ROLE is immutable. However, once migrated to a new user-defined role, any underlying system privileges can be individually revoked from the new role and granted to other user-defined roles. When this occurs, either the user-defined role to which the system privileges are granted or each individually revoked system privileges must be granted to the DBO system role.

This ensures that all system privileges required to execute multiplex management stored procedures remain granted to the DBO system role.

Starting Multiplex Servers

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.)

To start and stop IQ multiplex servers interactively, use the start server utility, **start_iq**. To start IQ multiplex servers in scripts, use command line parameters.

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

```
start_iq @/host1/mpxdevices/params.cfg -n host1_test1_writer -x
"tcpip{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 MANAGE MULTIPLEX system privilege 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 162

Adding Multiplex Servers

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
   ```

   This command creates the definition for a secondary writer server `mpxnode_w2`.
3. 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.
4. Remove the transaction log files in the directory where the secondary server will run:

   ```
   rm -rf mpxtest.log
   ```
5. Run **dblog** to reset the transaction log:

   ```
   dblog -r -t mpxtest.log mpxtest.db
   ```
6. Start the secondary server:

   ```
   start_iq -STARTDIR /host2/mpx @params.cfg -n mpxnode_w1 -x
   "tcpip{port=2957}" mpxtest.db
   ```
7. Start Interactive SQL and connect to the secondary multiplex node:

   ```
   dbisql -c "uid=DBA;pwd=sql;eng=mpxnode_w1;links=tcpip{port=2957}"
   ```
8. 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 *ALTER DATABASE* in *Reference: Statements and Options*.

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 dbfile in IQ_SYSTEM_TEMP.

See also

- *CREATE MULTIPLEX SERVER Statement* on page 107

Adding a Temporary Dbfile

You must add a temporary dbfile to IQ_SYSTEM_TEMP on a newly added server.

Prerequisites

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

Task

1. Start Interactive SQL and connect to the secondary multiplex node:

   ```
   dbisql -c "uid=DBA;pwd=sql;eng=mpxnode_w1;links=tcpip{port=2957}"
   ```

2. 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 *ALTER DATABASE* in *Reference: Statements and Options*.

3. Run **sp_iqmpxvalidate** to make sure that no errors are reported on this server.

Including or Excluding Servers

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 a Failover Node

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 27
- *sp_iqmpxinfo Procedure* on page 162

Dropping Multiplex Servers

Dropping a secondary server removes it from the multiplex configuration.

Prerequisites

If the target server is running, shut it down before dropping it.

You cannot drop the coordinator node and the designated failover node unless it is the last secondary node.

The free list is an internal structure that manages space allocation for a dbspace. A write server that is holding free list cannot be dropped.

Task

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

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

Next

When the last secondary server is removed, the multiplex is converted back to simplex and the coordinator shuts down.

See also

- *Coordinator Failure* on page 26
- *Global Transaction Resiliency* on page 42
- *ALTER LS POLICY Statement* on page 87
- *DROP MULTIPLEX SERVER Statement* on page 109
- *MPX_LIVENESS_TIMEOUT Option* on page 119

Reclaiming Free List

A normal restart of the write server gives up its free list, so that you can shut it down and drop it.

Prerequisites

If, for some reason, the write server cannot start, follow this process.

Task

1. Verify that the write server process is dead and cannot be started.
2. Restart the coordinator with the **-iqmpx_reclaimwriterfreelist** switch. This forces the coordinator to reclaim the writer's free list and you can then drop it.

   Warning! If the write server process is still writing to the database when the coordinator reclaims its free list, database corruption may result.

Altering Multiplex Servers

You can use Interactive SQL to alter a multiplex server.

Prerequisites

The coordinator must be running.

Task

Start the server, connect to it, and issue an **ALTER MULTIPLEX SERVER** command:

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

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

Changing Host and Port

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.

Changing Server Roles

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 the Database File Path

Change the file path with Interactive SQL to move your database.

For example, you could move your database to a disk with more space.

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.

Renaming Multiplex Servers

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 commits.

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 89

Synchronizing Servers

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

Prerequisites

Log in as a user with the BACKUP DATABASE privilege and perform the following steps on the computer where the secondary server runs.

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

- *Updates on IQ_SYSTEM_MAIN* on page 36
- *Files on Shared Disk Arrays* on page 6
- *Before You Restore* on page 72
- *RESTORE DATABASE Statement* on page 110
- *sp_iqdbspace Procedure* on page 144

Multiplex Login Management

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

For an overview of SAP Sybase IQ login policies, see *Administration: User Management and Security*.

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')
```

High Availability

SAP Sybase IQ Multiplex is a highly available system.

As with any system, failures are possible on any node. A multiplex with many nodes improves service availability should one or more nodes fail, because the work load can transfer to other participating nodes.

Many SAP Sybase IQ multiplex features can help you build and maintain systems with high service availability. Among these are coordinator failover, transaction resiliency, and the use of logical servers to isolate read-only query workload.

Reader Node Failure

Failure of software, hardware, or local disk on a reader causes a temporary outage for connected clients.

Clients connected to any other node will continue to function without disruption. Any distributed query being executed on the reader where the reader is not the leader node for that query will be completed by the leader node and thus cause no disruption to clients connected to the rest of the nodes.

Clients connected to the failed reader will see an outage. When the clients try to reconnect, they can be redirected to a node that is up using the login redirection feature or using a third party redirector. Depending on the severity of the failure, the failed node can be restarted if it is a software issue or restarted after fixing the hardware or disk issue.

Writer Node Failure

Failure of software, hardware, or local disk on a writer causes a temporary outage for connected clients. Read-write operations are rolled back on the failed node.

Clients connected to any other node will continue to function without disruption. Any distributed query being executed on the writer where the writer is not the leader node for that

query will be completed by the leader node and thus cause no disruption to clients connected to the rest of the nodes. Any read-write transaction on the failed writer will be disrupted and the transaction rolled back.

Clients connected to the failed writer will see an outage. When the clients try to reconnect, they can be redirected to a node that is up using the login redirection feature or using a third party redirector. Depending on the severity of the failure, the failed node can be restarted if it is a software issue or restarted after fixing the hardware or disk issue.

Coordinator Failure

If the current coordinator node fails, or must be shut down for maintenance, clients on other nodes can be affected. Read-write operations are rolled back on the failed node.

Clients connected to the failed coordinator experience an outage. When the clients try to reconnect, they can be redirected to a node that is up using the login redirection feature or using a third party redirector. Depending on the severity of the failure, the failed node can be restarted if it is a software issue or restarted after fixing the hardware or disk issue.

Client Location	Result
Reader node where DQP is not enabled	Not affected by coordinator failure
Reader node where DQP is enabled	These nodes periodically require space on IQ_SHARED_TEMP. When that happens, these DQP transactions are suspended. (See *Global Transaction Resiliency.* on page 42) The clients experience a pause until the coordinator is brought back up or failed over. If the coordinator cannot be brought back up or failed over within a user controlled time out period, then these DQP transactions roll back and the clients experience an outage.
Writer node	The clients on writer nodes that are doing read-write operations periodically need more space in shared main dbspaces or require global locks on tables they modify. When that happens, these transactions suspend. The clients experience a pause until the coordinator is brought back up or failed over. If the coordinator cannot be brought back up or failed over within a user controlled time out period, then these read-write transactions roll back and the clients experience an outage.

These dependencies make it critical that the coordinator stay up at all times. If the coordinator fails, restart the node immediately or promote another server to be the coordinator, also called *manual failover*.

See also

- *ALTER LS POLICY Statement* on page 87

- *DROP MULTIPLEX SERVER Statement* on page 109
- *MPX_LIVENESS_TIMEOUT Option* on page 119
- *Dropping Multiplex Servers* 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 or from Sybase Control Center, if you prefer a web-based management tool. For more information, see the Sybase Control Center for SAP Sybase IQ online help in SCC or at *http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680*.

See also

- *sp_iqmpxinfo Procedure* on page 162
- *Designating a Failover Node* on page 21

Replacing the Coordinator (Manual Failover)

Make sure that the coordinator is no longer running before you replace it.

Prerequisites

- The coordinator process must be dead before manual failover.

  Note: 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, 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.
- Use a reader for the designated failover node, if possible. Readers have no pending writeable transactions, which makes failover easier.
- The designated coordinator node must be included and part of the multiplex.

Task

1. Ensure that coordinator process is dead.

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

   If there were any read-write transactions running on secondary nodes when the original coordinator was shut down, these transactions roll 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 **iqsrv16** process.
- On Windows, log into the coordinator machine. Start Task Manager and look for the process name `iqsrv16.exe`. Stop the `iqsrv16.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 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 starts, the failover process is complete and the designated failover node is the new coordinator node. After failover, on the next transaction, 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 once you synchronize it against the new coordinator.

To perform failover using Sybase Control Center, see the Sybase Control Center for SAP Sybase IQ online help in SCC or at *http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680*.

Synchronizing the Former Coordinator

To start a former coodinator, synchronize it against the new coordinator.

Prerequisites

Perform the following steps on the computer where the former coordinator runs.

Task

1. Remove the `.LOG` transaction log file in the former coordinator directory.
2. Back up the catalog store to the directory where the former coordinator runs. Specify the new coordinator's connection parameters in the -c string:

```
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 former coordinator using the **start_iq** command.

Coordinator Failure and Restart

If the coordinator restarts during a global transaction, due to shutdown, failover or server failure, transaction behavior depends on the user-defined timeout and the command being executed.

Table 4. Communication to Coordinator Resumes Before Timeout

Command Status	Command Behavior	Result
Actively executing command	Command suspends and transfers control to server, except for ROLLBACK, which executes locally on writer.	Commands succeed
New DML command	Command suspends and transfers control to server. ROLLBACK and ROLLBACK TO SAVEPOINT execute locally instead of suspending.	Commands succeed

Table 5. Communication Failure Exceeds Timeout

Command Status	Command Behavior	Result
Suspended DML command on connection	The suspended command fails and returns an error.	Transaction rolls back
No suspended DML command on connection	The next command returns an error.	Transaction rolls back

For example, if a transaction suspends and the write server becomes inaccessible, you can roll back the transaction to release resources held by the coordinator for suspended transactions belonging to that writer.

SAN or Shared Disk Failure

These failures are the most serious because they can cause outage for all multiplex nodes.

Check the disk vendor support for high availability for this kind of failure.

Manage Transactions

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

Local Transactions

A *local transaction* does not modify a shared object.

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.

All multiplex transactions begin as local transactions.

Global Transactions

A *global transaction* 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.

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 transaction-level versioning (TLV) log.

DML Commands

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.

You can run global transactions from the coordinator and any write server.

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 6. Multiplex table data scope

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

SAP 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.

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. SAP 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
```

SAP 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 7. 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 on a multiplex:

- 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 be at 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.

Due to 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.

Start the coordinator in single-node mode (**-iqmpx_sn -1**) 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: Do not start the coordinator in single-node mode with **-iqro**, or adding or dropping a file returns the error `SQL Anywhere Error -757: Modifications not permitted for read-only database`.

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 *Data Storage* in *Administration: Database*.

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.

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.
- **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 38
- *Synchronizing Servers* on page 24

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.

The IQ_SYSTEM_MAIN dbspace manages important database structures including the freelist, which tracks which blocks are in use. Shut down secondary nodes before adding space

to IQ_SYSTEM_MAIN. 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.
```

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.

See also

- *Updates on IQ_SYSTEM_MAIN* on page 36
- *Files on Shared Disk Arrays* on page 6

Replacing Dbfiles in IQ_SYSTEM_TEMP on a 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.
- Updates on the IQ_SHARED_TEMP store require the MANAGE ANY DBSPACE system privilege.
- You cannot execute **ALTER FILE RENAME PATH** on the IQ_SHARED_TEMP dbspace.
- You cannot execute **ALTER DBSPACE ALTER FILE READONLY** on the IQ_SHARED_TEMP dbspace.

Note: If you start the coordinator in single node mode with **-iqro**, adding or dropping a file returns the error `SQL Anywhere Error -757: Modifications not permitted for read-only database`.

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 the file status for all shared files to the coordinator. SAP 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 cannot participate in any DQP operations. Use **sp_iqmpxfilestatus** to diagnose distributed file problems.

Adding Dbfiles to Shared Dbspaces

Use **ALTER DATABASE ADD FILE** statement in Interactive SQL to add dbfiles to shared dbspaces.

Prerequisites

When adding space to IQ_SHARED_TEMP, consider the distributed query processing workload. Dbspace commands on IQ_SHARED_TEMP require the MANAGE ANY DBSPACE system privilege.

Task

1. Connect to the coordinator.

   Note: Do not start the coordinator in single-node mode (**-iqmpx_sn 1**) with **-iqro**, or adding a file returns the error `SQL Anywhere Error -757: Modifications not permitted for read-only database.`

2. Add the file. In 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 as 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
```

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. You can force read-write status on IQ_SHARED_TEMP dbfiles if you need to postpone problem correction:

```
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.

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_sn 1** switch:

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

5. Synchronize all secondary servers in the multiplex.

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.

Global Transaction Resiliency

DML read-write transactions on multiplex writer nodes survive temporary communication failures between coordinator and writer nodes and temporary failure of the coordinator due to server failure, shutdown or failover.

When a user connects to a writer node and executes read-write DML commands against shared objects, the writer starts a global transaction on the coordinator. The transaction starts on an internal internode communication (INC) connection from writer to coordinator.

For example, INSERT or LOAD commands on shared database objects are global transactions. If a failure occurs, the global transaction and corresponding INC connection is suspended.

If the temporary failure resolves within a user-defined timeout period, the global transaction continues as if there was no failure. The user can commit, roll back, or continue the transaction. Use the MPX_LIVENESS_TIMEOUT option to set the timeout period, which defaults to an hour (default value 3600).

If the failure persists longer than the user-defined timeout period, the global transaction cannot resume and the user must roll back the whole transaction.

If there is a DML command actively executing while the failure happens, the command behavior depends on the user-defined timeout and the command type.

To check connection status (active or suspended), use the **sp_iqconnection** system procedure on a writer node or **sp_iqmpxsuspendedconninfo** system procedure on a coordinator. Run **sp_iqmpxincstatistics** for a snapshot of the aggregate statistics of the INC status since server startup.

This feature does not affect transactions initiated on the coordinator.

See also

- *ALTER LS POLICY Statement* on page 87
- *DROP MULTIPLEX SERVER Statement* on page 109
- *MPX_LIVENESS_TIMEOUT Option* on page 119
- *Dropping Multiplex Servers* on page 21
- *Troubleshoot Transactions* on page 43
- *sp_iqconnection Procedure* on page 139
- *sp_iqmpxincstatistics Procedure* on page 161

- *sp_iqmpxsuspendedconninfo Procedure* on page 164

Troubleshoot Transactions

Commands are unaffected by many communication or coordinator failures, but certain cases require user action.

Communication Failure or Coordinator Failure and Restart During Global Transaction

If internode communication (INC) fails or the coordinator fails or is shut down during a writer-initiated global transaction, transactions suspend and resume automatically if the INC is restored before a user-specified timeout expires.

Delays in command execution may indicate INC suspend and resume operations. If INC is interrupted, the coordinator suspends a global transaction for an hour. The transaction resumes successfully as soon as INC is restored. If the timeout value elapses, the transaction fails. Set the **MPX_LIVENESS_TIMEOUT** database option to change the timeout period.

The following cases describe the behavior of writer nodes.

Table 8. Communication to Coordinator Resumes Before Timeout

Writer Command Status	Command Behavior	Result
Actively executing command	Command suspends, except for ROLLBACK, which executes locally on writer.	Command succeeds.
New DML command	Command suspends and resumes, except for ROLLBACK and ROLLBACK TO SAVEPOINT, which execute locally on the writer.	If communication is restored, resumed commands succeed.

Table 9. Communication Failure Exceeds Timeout

Writer Command Status	Command Behavior	Result
Suspended DML command on connection	The suspended command fails and returns an error about the non-recoverable state of the transaction.	You must roll back the transaction. Rollback happens automatically if the suspended command is COMMIT or ROLLBACK to SAVEPOINT.
No suspended DML command on connection	The next command returns an error about the non-recoverable state of the transaction.	You must roll back the transaction.

To check connection status, use the **sp_iqconnection** system procedure on a writer node or the **sp_iqmpxsuspendedconninfo** system procedure on a coordinator.

Run **sp_iqmpxincstatistics** for a snapshot of the aggregate statistics of the INC status since server startup.

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 a scenario that updates both local and proxy tables in the same transaction, where “best effort” is used to commit both local and global components of a transaction.

Reclaiming Resources Held for Global Transactions

If a writer node with one or more active global transactions becomes inaccessible, the coordinator does not immediately release resources held for global transactions.

The coordinator waits for a period of `2 * Mpx_Liveness_Timeout` for the writer to resume the global transaction. If the writer does not resume by the end of the period, the coordinator releases resources. To free the resources held for the global transactions immediately, kill the corresponding suspended INC connections on the coordinator.

1. Run the stored procedure **sp_iqmpxsuspendedconninfo** to locate the suspended connection:

```
call sp_iqmpxsuspendedconninfo

ConnName              ConnHandle      GlobalTxnId
====================  =============   =============
```

```
'IQ_MPX_SERVER_P5'              15            3920

MPXServerName        TimeInSuspendedState
==================   ======================
'HP1_12356_IQ_mpx'                        50

SuspendTimeout
===============
           7200
```

2. On the server specified in `MPXServerName`, issue a drop command that specifies the `ConnHandle` for the connection:

```
call "DROP CONNECTION 15"
```

Coordinator Failover and Global Transactions

If the coordinator fails over during a writer-initiated global transaction before a user-specified timeout expires, transactions suspend and resume automatically.

Delays in command execution may indicate internode communication (INC) suspend and resume operations.

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.

Manage Resources Through Logical Servers

You must use logical servers to access multiplex servers.

Logical servers provide resource provisioning for the IQ multiplex by grouping 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, SAP 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 SAP 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.

See also

- *Use Built-in Logical Servers* on page 49
- *Creating a Logical Server* on page 52
- *Altering a Logical Server* on page 53
- *Dropping a Logical Server* on page 54
- *Connecting to a Logical Server* on page 54
- *Configure Logical Server Policies* on page 56
- *Manage Logical Server Membership* on page 57
- *Redirecting Logins* on page 59
- *Disabling Login Redirection* on page 60
- *Login Policies* on page 61

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

To specify logical membership for the coordinator in a user-defined logical server, use the **FOR LOGICAL COORDINATOR** clause instead of the 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 Servers and Simplex Configurations

Only multiplexes support logical servers, but information about built-in logical servers and logical server policies can remain, unused, in the catalog in a simplex environment.

Logical Servers and New Simplex Databases

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

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.

Use 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.

See also

- *Logical Servers Overview* on page 47
- *Creating a Logical Server* on page 52
- *Altering a Logical Server* on page 53
- *Dropping a Logical Server* on page 54
- *Connecting to a Logical Server* on page 54
- *Configure Logical Server Policies* on page 56
- *Manage Logical Server Membership* on page 57

- *Redirecting Logins* on page 59
- *Disabling Login Redirection* on page 60
- *Login Policies* on page 61

ALL Logical Server

The ALL logical server allows access to all logical servers.

When you specify the ALL logical server, there is no need to list the names as you add or drop servers.

See also

- *AUTO Logical Server* on page 50
- *COORDINATOR Logical Server* on page 50
- *NONE Logical Server* on page 51
- *OPEN Logical Server* on page 51
- *SERVER Logical Server* on page 52

AUTO Logical Server

Specify the AUTO logical server to prevent login redirection.

If the node belongs to multiple logical servers, using the AUTO logical server returns an error. If the node belongs to a single logical server, AUTO uses the logical server context of the current node.

See also

- *ALL Logical Server* on page 50
- *COORDINATOR Logical Server* on page 50
- *NONE Logical Server* on page 51
- *OPEN Logical Server* on page 51
- *SERVER Logical Server* on page 52

COORDINATOR Logical Server

COORDINATOR is a built-in logical server that consists of the current coordinator node.

Clients can use the COORDINATOR logical server to connect to the current coordinator without knowing its name. This simplifies connection because the coordinator role may pass from one node to another, for example, during failover.

You cannot drop the COORDINATOR logical server.

See also

- *ALL Logical Server* on page 50
- *AUTO Logical Server* on page 50

- *NONE Logical Server* on page 51
- *OPEN Logical Server* on page 51
- *SERVER Logical Server* on page 52

NONE Logical Server

The NONE logical server is defined to be always empty.

See also

- *ALL Logical Server* on page 50
- *AUTO Logical Server* on page 50
- *COORDINATOR Logical Server* on page 50
- *OPEN Logical Server* on page 51
- *SERVER Logical Server* on page 52

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.

See also

- *ALL Logical Server* on page 50
- *AUTO Logical Server* on page 50
- *COORDINATOR Logical Server* on page 50
- *NONE Logical Server* on page 51
- *SERVER Logical Server* on page 52

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.

SERVER is the default logical server for **dbbackup** and **dbstop** commands.

When you connect using the SERVER logical server context, SAP Sybase IQ ignores the NodeType connection parameter.

You must have the ACCESS SERVER LS system privilege to connect to a multiplex with the SERVER logical server context:

See also

- *ALL Logical Server* on page 50
- *AUTO Logical Server* on page 50
- *COORDINATOR Logical Server* on page 50
- *NONE Logical Server* on page 51
- *OPEN Logical Server* on page 51

Creating a Logical Server

Create a user-defined logical server using Interactive SQL.

1. Connect to the database as a user with the MANAGE MULTIPLEX system privilege.
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

- *Logical Servers Overview* on page 47
- *Use Built-in Logical Servers* on page 49
- *Altering a Logical Server* on page 53
- *Dropping a Logical Server* on page 54
- *Connecting to a Logical Server* on page 54
- *Configure Logical Server Policies* on page 56
- *Manage Logical Server Membership* on page 57
- *Redirecting Logins* on page 59
- *Disabling Login Redirection* on page 60

- *Login Policies* on page 61
- *CREATE LOGICAL SERVER Statement* on page 98

Commenting on Logical Servers

To simplify administration, comment on user-defined logical servers.

1. Connect to the database as a user with the MANAGE MULTIPLEX system privilege.
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

Altering a Logical Server

Alter a user-defined logical server using Interactive SQL.

1. Connect to the database as a user with the MANAGE MULTIPLEX system privilege.
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

- *Logical Servers Overview* on page 47
- *Use Built-in Logical Servers* on page 49
- *Creating a Logical Server* on page 52
- *Dropping a Logical Server* on page 54
- *Connecting to a Logical Server* on page 54
- *Configure Logical Server Policies* on page 56
- *Manage Logical Server Membership* on page 57
- *Redirecting Logins* on page 59
- *Disabling Login Redirection* on page 60
- *Login Policies* on page 61
- *ALTER LOGICAL SERVER Statement* on page 79

Dropping a Logical Server

Drop a user-defined logical server using Interactive SQL.

1. Connect to the database as a user with the MANAGE MULTIPLEX system privilege.
2. Execute a **DROP LOGICAL SERVER** statement.

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

   ```
   DROP LOGICAL SERVER ls1
   ```

See also

- *Logical Servers Overview* on page 47
- *Use Built-in Logical Servers* on page 49
- *Creating a Logical Server* on page 52
- *Altering a Logical Server* on page 53
- *Connecting to a Logical Server* on page 54
- *Configure Logical Server Policies* on page 56
- *Manage Logical Server Membership* on page 57
- *Redirecting Logins* on page 59
- *Disabling Login Redirection* on page 60
- *Login Policies* on page 61
- *DROP LOGICAL SERVER Statement* on page 108

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. Ensure that there are no active connections for a logical server when it is being dropped.

Connecting to a Logical Server

Use the **LogicalServer** and **NodeType** connection parameters to establish the logical server context for a new user connection.

In a configuration file or at the Interactive SQL command line:

1. To specify the target logical server, add the `LogicalServer=<target-logical-server>` clause to a connect statement.
2. To specify the target logical server role, add the `NodeType={READER|WRITER|ANY}` clause to a connect statement.

   For example, to connect an application that specifically needs to execute user-defined functions on member nodes with the READER role, use:

   ```
   dbisql -c
   "uid=DBA;pwd=sql;eng=host4_iqdemo;LS=LogSvr1;NODETYPE=READER;
   ```

   Note: When you connect using the SERVER logical server context, SAP Sybase IQ ignores the NodeType connection parameter.

Connections fail if:

- The current node is not a member of any logical server assigned to the user's login policy.
- The current login policy assigns SERVER logical server, and the user lacks sufficient system privilege.
- The current login policy assigns NONE as logical server.
- The current login policy assigns COORDINATOR as the logical server, and the user connects to a secondary server without enabling login redirection.

For new and upgraded databases, the **default_logical_server** login policy is AUTO. User-defined login policies use the value from the root login policy, which defaults to AUTO. When the default logical server policy is AUTO, login redirection never occurs, even if the LOGIN REDIRECTION logical server policy option. is set ON.

Under the AUTO setting, if the connection string fails to specify **LogicalServer**, SAP Sybase IQ automatically determines logical server context as follows:

Current Physical Node Status	Result
Node belongs to multiple logical servers assigned to a single login policy. A user belonging to that login policy logs in to the common node, but SAP Sybase IQ cannot determine which logical server to use.	Connection refused and error raised due to overlapping server scenario.
Node belongs to a single logical server.	Connection succeeds and context is that logical server.

See also

- *Logical Servers Overview* on page 47
- *Use Built-in Logical Servers* on page 49
- *Creating a Logical Server* on page 52
- *Altering a Logical Server* on page 53

Configure Logical Server Policies

A logical server policy is associated with each logical server. Configure logical server policy options to control behavior of all associated logical servers.

An SAP Sybase IQ database includes a built-in root logical server policy that applies to all logical servers. You cannot drop the root logical server policy.

See **CREATE LS POLICY** and **ALTER LS POLICY** for valid logical server policy options.

See also

- *Logical Servers Overview* on page 47
- *Use Built-in Logical Servers* on page 49
- *Creating a Logical Server* on page 52
- *Altering a Logical Server* on page 53
- *Dropping a Logical Server* on page 54
- *Connecting to a Logical Server* on page 54
- *Manage Logical Server Membership* on page 57
- *Redirecting Logins* on page 59
- *Disabling Login Redirection* on page 60
- *Login Policies* on page 61
- *ALTER LS POLICY Statement* on page 87
- *CREATE LS POLICY Statement* on page 104
- *Altering Root Logical Server Policy* on page 56

Altering Root Logical Server Policy

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

1. Connect to the database as a user with the MANAGE MULTIPLEX system privilege.
2. Execute an **ALTER LS POLICY** statement:

   To alter the root policy:

   ```
   ALTER LS POLICY root TEMP_DATA_IN_SHARED_TEMP=OFF;
   ```

   To alter a user-defined logical server policy:

```
CREATE LS POLICY lsp1 TEMP_DATA_IN_SHARED_TEMP=OFF;
ALTER LS POLICY lps1 TEMP_DATA_IN_SHARED_TEMP=ON;
```

See also

- *Configure Logical Server Policies* on page 56
- *ALTER LS POLICY Statement* on page 87
- *CREATE LS POLICY Statement* on page 104

Manage Logical Server Membership

You can only access multiplex servers 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 members of the assigned logical servers. You can make one of these assignments to a login policy:

- ALL – allows access to all logical servers, so that you need not specify server names when you add or drop them.
- AUTO – prevents login redirection. Returns an error if the node belongs to multiple logical servers. If the node belongs to a single logical server, uses the logical server context of the current node.
- COORDINATOR – allows access to the current coordinator node, so that, should the coordinator role passes from one server to another, you need not specify the new server name.
- One or more user-defined logical servers.
- OPEN – allows access to all multiplex nodes that are not members of any user-defined logical servers.
- SERVER – allows access to all multiplex nodes, subject to the semantics of the SERVER logical server.
- NONE – denies access to any multiplex server.

Note: Do not combine SERVER or NONE with other logical server assignments.

See also

- *Logical Servers Overview* on page 47
- *Use Built-in Logical Servers* on page 49
- *Creating a Logical Server* on page 52
- *Altering a Logical Server* on page 53
- *Dropping a Logical Server* on page 54
- *Connecting to a Logical Server* on page 54
- *Configure Logical Server Policies* on page 56
- *Redirecting Logins* on page 59

- *Disabling Login Redirection* on page 60
- *Login Policies* on page 61

Logical Server Assignment Inheritance

A login policy without a logical server assignment inherits assignments from the root login policy.

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.

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

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.

Changing Logical Server Assignments

Alter logical server assignments using Interactive SQL.

1. Connect to the database as a user with the MANAGE ANY LOGIN POLICY system privilege.
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 80

Removing Logical Server Assignments

Remove all existing logical server assignments from a user-defined login policy.

To remove existing logical server assignments from a user-defined login policy, set 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.

1. Connect to the database as a user with the MANAGE ANY LOGIN POLICY system privilege.
2. Execute an **ALTER LOGIN POLICY** statement that assigns the logical server to DEFAULT.

   For example, to alter a logical server assignment by assigning logical server `DEFAULT` to the user-defined login policy `lp1`, enter:

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

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.

Redirecting Logins

SAP Sybase IQ provides load balancing when a user tries to log in to an overloaded node by redirecting the attempted login to a node that is less loaded in the same logical server.

1. Create a logical server policy where login redirection is enabled, for example:

```
CREATE LS POLICY lsp1 LOGIN_REDIRECTION=ON
```

2. Associate the new policy with a logical server:

```
ALTER LOGICAL SERVER ls1 POLICY lsp1
```

If login redirection is enabled and a connection is allowed, SAP Sybase IQ redirects connections when:

- The initial connection node is not a member of the target logical server.
- The initial connection node is a member of the target logical server, but has a role other than that requested.
- The initial node is a member of the target logical server and has the requested role, but the user has reached the limit of maximum connections on the current logical server member node.

Note: SERVER is the default logical server for **dbbackup** and **dbstop** commands.

See also

- *Logical Servers Overview* on page 47
- *Use Built-in Logical Servers* on page 49
- *Creating a Logical Server* on page 52
- *Altering a Logical Server* on page 53
- *Dropping a Logical Server* on page 54
- *Connecting to a Logical Server* on page 54
- *Configure Logical Server Policies* on page 56
- *Manage Logical Server Membership* on page 57
- *Disabling Login Redirection* on page 60
- *Login Policies* on page 61

Disabling Login Redirection

You can disable login redirection for all logical servers governed by a named logical server policy, or at the connection level.

1. Disable login redirection at the logical server level.

```
ALTER LS POLICY mypolicy LOGIN_REDIRECTION=OFF
```

   You cannot enable login redirection at the connection level once you disable it at the logical server level:

2. Disable login redirection at the connection level if you do not need to affect all servers associated with a particular logical server policy. When disabled, no redirection occurs for the connection, and the connection fails if the node specified cannot satisfy connection requirements of the target logical server and requested role.

For example, if an application needs to target specific nodes within a logical server that contains data in tables not shared between nodes, such as SQL Anywhere tables, disable at the connection level:

```
dbisql -c
"uid=dba;pwd=sql;eng=mpxnode_w1;links=tcpip;redirect=off"
```

See also

- *Logical Servers Overview* on page 47
- *Use Built-in Logical Servers* on page 49
- *Creating a Logical Server* on page 52
- *Altering a Logical Server* on page 53
- *Dropping a Logical Server* on page 54
- *Connecting to a Logical Server* on page 54
- *Configure Logical Server Policies* on page 56
- *Manage Logical Server Membership* on page 57
- *Redirecting Logins* on page 59
- *Login Policies* on page 61

Login Policies

A *login policy* defines the rules that SAP Sybase IQ follows to establish user connections. Each login policy is associated with a set of options called login policy options.

Login management commands that you execute on any multiplex server are automatically propagated to all servers in the multiplex. For best performance, execute these commands, or any DDL, on the coordinator.

Warning! Migrating databases from version 12.7 removes existing login management settings. You must re-create them after migration.

See also

- *Logical Servers Overview* on page 47
- *Use Built-in Logical Servers* on page 49
- *Creating a Logical Server* on page 52
- *Altering a Logical Server* on page 53
- *Dropping a Logical Server* on page 54
- *Connecting to a Logical Server* on page 54
- *Configure Logical Server Policies* on page 56
- *Manage Logical Server Membership* on page 57
- *Redirecting Logins* on page 59
- *Disabling Login Redirection* on page 60

Modifying the Root Login Policy

You can modify the option values for the root login policy, but you cannot drop the policy.

Prerequisites

Requires the MANAGE ANY LOGIN POLICY system privilege.

Task

Each new database is created with a default login policy, called the root policy. When you create a user account without specifying a login policy, the user becomes part of the root login policy.

To modify the options of the root login policy, execute:

```
ALTER LOGIN POLICY ROOT {login_policy_options}
```

See also

- *Login Policy Options* on page 84
- *ALTER LOGIN POLICY Statement* on page 80
- *CREATE LOGIN POLICY Statement* on page 100

Creating a New Login Policy

Any options that are not explicitly set when creating a login policy inherit their values from the root login policy..

Prerequisites

Requires the MANAGE ANY LOGIN POLICY system privilege.

Task

Login policy names must be unique. An error message appears if the login policy name already exists.

To create a new login policy, execute:

```
CREATE LOGIN POLICY policy_name {login_policy_options}
```

Example:

This statement creates the `Test1` login policy with PASSWORD_LIVE_TIME option set to 60 days:

```
CREATE LOGIN POLICY Test1
password_life_time=60
```

See also

- *Login Policy Options* on page 84
- *ALTER LOGIN POLICY Statement* on page 80

- *CREATE LOGIN POLICY Statement* on page 100

Modifying an Existing Login Policy

Use Interactive SQL to change the options for an existing login policy.

Prerequisites

Requires the MANAGE ANY LOGIN POLICY system privilege.

Task

To alter the options of an existing login policy, execute:

ALTER LOGIN POLICY *policy-name {login_policy_options}*

Example:

This statement alters the LOCKED and MAX_CONNECTIONS options on the `Test1` login policy:

```
ALTER LOGIN POLICY Test1
locked=on
max_connections=5
```

See also

- *Login Policy Options* on page 84
- *ALTER LOGIN POLICY Statement* on page 80
- *CREATE LOGIN POLICY Statement* on page 100

Displaying a List of Users Assigned a Login Policy

Before you can drop a login policy, ensure that it is not currently assigned to any users. To display the users assigned to a login policy, select from the SYSUSER system view based on the login policy identifier.

Example:

This statement lists users assigned to the `lp1` login policy:

```
SELECT user_name FROM sysuser WHERE login_policy_id = ( SELECT
login_policy_id FROM
    sysloginpolicy WHERE login_policy_name='lp1' )
```

See also

- *Login Policy Options* on page 84
- *ALTER LOGIN POLICY Statement* on page 80
- *CREATE LOGIN POLICY Statement* on page 100

Deleting a Login Policy

You cannot delete the root login policy or one currently assigned to a user.

Prerequisites

Requires the MANAGE ANY LOGIN POLICY system privilege.

Task

1. Verify that no users are currently assigned the login policy to be dropped.
2. To drop a login policy, execute:

```
DROP LOGIN POLICY policy_name
```

Assigning a Login Policy When Creating a New User

If you do not assign a login policy when creating a user account, the account is assigned the root login policy.

Prerequisites

Requires the MANAGE ANY LOGIN POLICY system privilege.

Task

Assign a login policy other than the root login policy when creating a new user. A user can be assigned only one login policy at a time.
To assign a login policy, execute:

```
CREATE USER userID
[ IDENTIFIED BY password ]
[ LOGIN POLICY policy-name ]
```

Note: You cannot specify multiple user IDs in the same **CREATE USER** command when assigning a login policy to users.

Example:

This statement creates a user called `Joe` with the password `welcome`, and assigns the login policy `Test2`:

```
CREATE USER Joe
IDENTIFIED BY welcome
LOGIN POLICY Test2
```

See also

- *Login Policy Options* on page 84
- *ALTER LOGIN POLICY Statement* on page 80
- *CREATE LOGIN POLICY Statement* on page 100

Assigning a Login Policy to an Existing User

Use Interactive SQL to assign a login policy to an existing user.

Prerequisites

Requires the MANAGE ANY LOGIN POLICY system privilege.

Task

1. To change the login policy assigned to a user, execute:

```
ALTER USER userID
LOGIN POLICY policy_name
```

2. Have the user log out and back in to apply the new login policy.

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 setting the **DQP_ENABLED** logical server policy option to 0, DQP occurs automatically for qualifying queries when:

- The server is part of a multiplex where servers have established MIPC (multiplex interprocess communication) 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 other member node available.

  Note: To control which multiplex nodes participate in distributed query processing, partition nodes into logical servers.
- The shared temporary dbspace has writable files available. See *IQ Shared Temporary Dbspace* in *Administration: Database*.

  Note: To use DQP without configuring a shared temporary dbspace, set the `DQP_ENABLED` logical server policy option to 2 to enable DQP over the network.

Use temporary database options to control DQP for the current connection. Setting the temporary database option `DQP_ENABLED` to OFF disables DQP for all queries executed on the current connection.

If the `DQP_ENABLED` logical server policy option is set to 1 and the `DQP_ENABLED` database option is set to ON, setting `DQP_ENABLED_OVER_NETWORK` to ON forces queries executed on the current connection to use DQP over the network.

If the `DQP_ENABLED` logical server policy option is set to 2 to force all queries to use DQP over the network, setting the database option `DQP_ENABLED` to OFF forces queries executed on the current connection to run in simplex mode.

If you enable DQP over the network at the connection level, the only way to disable it at the connection level is to set `DQL_ENABLED` database option OFF. This forces all queries run on the connection to run in simplex mode. If you enable DQL over the network, no queries use the shared temporary store.

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.

If a leader node fails, query processing ends, as it would on a single server. You can connect to another server to run the query, but this does not happen automatically.

Many types of queries can survive failures on worker nodes, either due to disconnect or timeout. If a worker fails, the leader executes pending work for the worker and assigns no further work from the current query fragment to that worker. The MPX_WORK_UNIT_TIMEOUT database option specifies the timeout duration in seconds (default 60).

Some queries support worker node failures at any time during the query, while others cannot once any intermediate results have been sent. The query plan detail displays statistics about work units that have been assumed by the leader. Queries that cannot support work retry on the leader are cancelled immediately.

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 *Planning for Distributed Query Processing or High Availability* in the *Installation and Configuration Guide*.

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

Back Up and Restore

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 recommended, 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.

See also

- *BACKUP Statement* on page 91
- *Backing Up Multiplex Databases* on page 72

Backing Up Multiplex Databases

Follow these guidelines to back up multiplex databases.

Prerequisites

Read the rest of the backup topics for complete details before you perform a backup.

Task

1. To back up the IQ store and catalog store on a multiplex database, connect to the coordinator using an account with the BACKUP DATABASE system privilege.
2. Issue a **BACKUP** statement, which backs up:
   - The catalog store (SYSTEM dbspace file), typically named *dbname*.db
   - All dbspace files of the IQ store
3. Make a copy of the `params.cfg` file for each server. **BACKUP** does not back it up.
4. Save the lengths of the IQ temporary store and all dbspace files on the coordinator.

See also

- *Backup Requirements* on page 71
- *BACKUP Statement* on page 91

Before You Restore

You can perform read-write restore operations only against a server running the utility database.

Before you restore, verify with Sybase Technical Support that a restore operation is necessary.

You cannot run restore operations against a secondary server.

Note: It is never necessary to restore a coordinator node due to secondary node problems. If you cannot open your database on a secondary server, synchronize the server.

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. Run **sp_iqdbspace** to determine offline status.

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

See also

- *RESTORE DATABASE Statement* on page 110

- *sp_iqdbspace Procedure* on page 144
- *Restoring IQ Store Data When Restoring the Database in the Same Location* on page 75
- *Restoring IQ Store Data When Restoring Database in a Different Location* on page 73
- *Synchronizing Servers* on page 24

Restoring IQ Store Data When Restoring Database in a Different Location

Restore operations vary, depending on the location from which you restore.

Prerequisites

- Confirm that there are database home directories for each server. If not, create them or restore them from file system backups.
- 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). The multiplex at the original location where the backup was taken may continue running.

  Note: If automatic startup is enabled in your ODBC configuration, there may be users on the same machine as the server who are set up to automatically start the server. Prevent this from happening while you are restoring the database.
- Confirm that the database shut down successfully:

Platform	Actions
UNIX	`% ps -ef | grep iqsrv16` If you see an active **iqsrv16** process with name of a multiplex, stop the process.
Windows	In Task Manager, check the Processes tab for `iqsrv16.exe`, or find the IQ Server icon in the system tray and stop it using right-click and **Shutdown**.

- Make a file system copy of the `.iqmsg` file. If you have message log archiving configured, see *Back Up the Right Files* in *Administration: Backup, Restore, and Data Recovery*.

Task

1. 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)
```

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

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

3. 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.

4. 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**).
5. 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.

6. Restart the coordinator without the single-node or override switch.
7. Re-create 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.

8. When you restart the coordinator, you see a warning in the server log about the multiplex environment being invalid. This warning is generated if IQ_SYSTEM_TEMP dbspace does not contain any files, and is the case for all the secondary nodes you created in step 7. Ignore this warning for now.
9. Synchronize and restart the secondary servers.
10. Connect to each secondary server and add files in IQ_SYSTEM_TEMP.
11. 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 5 through 11 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 72
- *RESTORE DATABASE Statement* on page 110
- *sp_iqdbspace Procedure* on page 144

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).

   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 iqsrv16` If you see an active **iqsrv16** process with name of a multiplex, stop the process.
Windows	In Task Manager, look on the Processes tab for `iqsrv16.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 *Back Up the Right Files* in *Administration: Backup, Restore, and Data Recovery*.
   - 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* on page 75.

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 *RESTORE Statement* in *Reference: Statements and Options*.
8. Shut down the utility database.
9. Make sure that the temporary dbspaces exist as before, on raw devices or as files of the correct length. See *Back Up the Right Files* in *Administration: Backup, Restore, and Data Recovery*. 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.
11. Start the secondary servers.

See also

- *Before You Restore* on page 72
- *RESTORE DATABASE Statement* on page 110
- *sp_iqdbspace Procedure* on page 144

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).

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.

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-ls-clause } [ WITH STOP SERVER ]
```

Parameters

- ***alter-ls-clause –***

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

- ***ls-member –***

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

Applies to

Multiplex only.

Examples

- **Example 1 –** 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 –** adds logical membership of COORDINATOR and drop 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)
```

- **Example 3 –** changes the logical server policy for logical server `ls2` to policy `lsp1`.

```
ALTER LOGICAL SERVER ls2 POLICY lsp1
```

Usage

logical-server-name refers to an existing user-defined 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.
- A logical server membership change causes a node to belong to multiple logical servers assigned to a single login policy. Logical server membership in a login policy cannot overlap.

WITH STOP SERVER automatically shuts down all servers in the logical server when the TEMP_DATA_IN_SHARED_TEMP option is changed directly or indirectly.

This statement enforces consistent shared system temporary store settings across physical nodes shared by logical servers.

Permissions

Requires the MANAGE MULTIPLEX system privilege.

See also

- *Altering a Logical Server* on page 53

ALTER LOGIN POLICY Statement

Changes existing login policies or configures logical server access.

Syntax

Syntax 1

```
ALTER LOGIN POLICY policy-name
{ { ADD | DROP | SET } LOGICAL SERVER ls-assignment-list
[ LOGICAL SERVER ls-override-list ])
```

- ***ls-assignment-list –***

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

- ***ls-override-list –***

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

- ***ls-name –***

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

- ***policy-option-value –***

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

Syntax 2

```
ALTER LOGIN POLICY policy-name
AUTO_UNLOCK_TIME=0 – UNLIMITED
| CHANGE_PASSWORD_DUAL_CONTROL=[ON | OFF]
| DEFAULT_LOGICAL_SERVER=[logical_server_name | ALL | AUTO | COORDINATOR |
NONE | OPEN | SERVER]
| LOCKED=[ON | OFF]
| MAX_CONNECTIONS=0 – 2147483647
| MAX_DAYS_SINCE_LOGIN=0 – 2147483647
| MAX_FAILED_LOGIN_ATTEMPTS=0 – 2147483647
| MAX_NON_DBA_CONNECTIONS=0 – 2147483647
| PASSWORD_EXPIRY_ON_NEXT_LOGIN=[ON | OFF]
| PASSWORD_GRACE_TIME=0 – 2147483647
| PASSWORD_LIFE_TIME=0 – 2147483647
| ROOT_AUTO_UNLOCK_TIME=0 – UNLIMITED
| LDAP_PRIMARY_SERVER=server_name
| LDAP_SECONDARY_SERVER=server_name
| LDAP_AUTO_FAILBACK_PERIOD=0 – 2147483647
| LDAP_FAILOVER_TO_STD=[ON | OFF]
| LDAP_REFRESH_DN=NOW
```

Applies to

Simplex and multiplex.

Examples

- **Example 1 –** see *Logical Server Access Configuration* and *Multiplex Login Policy Configuration* .
- **Example 2 –** sets the password_life_time value to `UNLIMITED` and the max_failed_login_attempts value to `5` in the `Test1` login policy.

```
ALTER LOGIN POLICY Test1
password_life_time=UNLIMITED
max_failed_login_attempts=5;
```

Permissions

Requires the MANAGE ANY LOGIN POLICY system privilege.

See also

- *Creating a New Login Policy* on page 62
- *Displaying a List of Users Assigned a Login Policy* on page 63
- *Modifying the Root Login Policy* on page 62
- *Modifying an Existing Login Policy* on page 63
- *Assigning a Login Policy When Creating a New User* on page 64
- *Changing Logical Server Assignments* on page 58
- *DQP_ENABLED Option* on page 116

Logical Server Access Configuration

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 `lp1`:

```
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

ADD, **DROP**, or **SET** clauses let you 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, a corresponding row exists in `ISYSIQLOGINPOLICYLSINFO`.

Multiplex Login Policy Configuration

Configure login policies for multiplex servers.

Example

This example overrides the login policy settings on a logical server, increasing the maximum number of connections on logical server `ls1`:

```
ALTER LOGIN POLICY lp1 max_connections=20 LOGICAL SERVER ls1;
```

Usage

Applies only to multiplex.

Any login management commands you execute on any multiplex server automatically propagate to all servers in the multiplex. For best performance, execute these commands, or any DDL, on the coordinator.

An override at the 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, a corresponding row exists in `ISYSIQLSLOGINPOLICYOPTION`.

Login Policy Options

Available options for root and user-defined login policies.

Option	Description
AUTO_UNLOCK_TIME	The time period after which locked accounts not granted the MANAGE ANY USER system privilege are automatically unlocked. This option can be defined in any login policy, including the root login policy. • **Values –** 0 – unlimited • **Initial value for Root policy –** Unlimited • **Applies to –** All users not granted the MANAGE ANY USER system privilege.
CHANGE_PASSWORD_DUAL_CONTROL	Requires input from two users, each granted the CHANGE PASSWORD system privilege, to change the password of another user. • **Values –** ON, OFF • **Initial value for Root policy –** OFF • **Applies to –** All users.

DEFAULT_LOGICAL_SERVER	If the connection string specifies no logical server, the user connects to the `DEFAULT_LOGICAL_SERVER` setting specified in the user's login policy.
	• **Values –**
	  • Name of an existing user-defined logical server
	  • ALL – allows access to all logical servers.
	  • AUTO – value of the default logical server in the root login policy.
	  • COORDINATOR – the current coordinator node.
	  • NONE – denies access to any multiplex server.
	  • OPEN – use alone or with the name of a user-defined logical server. Allows access to all multiplex nodes that are not members of any user-defined logical servers.
	  • SERVER – allows access to all of the multiplex nodes, subject to the semantics of the SERVER logical server.
	• **Initial value for Root policy –** AUTO
	• **Applies to –** All users. Requires MANAGE MULTIPLEX system privilege.

Option	Description
LOCKED	If set ON, users cannot establish new connections. This setting temporarily denies access to login policy users. Logical server overrides for this option are not allowed. • **Values –** ON, OFF • **Initial value for Root policy –** OFF • **Applies to –** All users except those with the MANAGE ANY USER system privilege.
MAX_CONNEC-TIONS	The maximum number of concurrent connections allowed for a user. You can specify a per-logical-server setting for this option. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** Unlimited • **Applies to –** All users except those with the SERVER OPERATOR or DROP CONNECTION system privilege.
MAX_DAYS_SINCE_LOGIN	The maximum number of days that can elapse between two successive logins by the same user. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** Unlimited • **Applies to –** All users except those with the MANAGE ANY USER system privilege.
MAX_FAILED_LOG-IN_ATTEMPTS	The maximum number of failed attempts, since the last successful attempt, to log into the user account before the account is locked. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** Unlimited • **Applies to –** All users.
MAX_NON_DBA_CONNECTIONS	The maximum number of concurrent connections that a user without SERVER OPERATOR or DROP CONNECTION system privileges can make. This option is supported only in the root login policy. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** Unlimited • **Applies to –** All users except those with the SERVER OPERATOR or DROP CONNECTION privilege.

Option	Description
PASSWORD_EXPIRY_ON_NEXT_LOGIN	If set ON, the user's password expires at the next login. • **Values –** ON, OFF • **Initial value for Root policy –** OFF • **Applies to –** All users. **Note:** This functionality is not currently implemented when logging in to Sybase Control Center. A user will not be prompted to change their password. He or she will be prompted, however, when logging in to SAP Sybase IQ outside of Sybase Control Center (for example, using Interactive SQL).
PASSWORD_GRACE_TIME	The number of days before password expiration during which login is allowed but the default post_login procedure issues warnings. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** 0 • **Applies to –** All users.
PASSWORD_LIFE_TIME	The maximum number of days before a password must be changed. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** Unlimited • **Applies to –** All users.
ROOT_AUTO_UNLOCK_TIME	The time period after which locked accounts granted the MANAGE ANY USER system privilege are automatically unlocked. This option can be defined only in the root login policy. • **Values –** 0 – unlimited • **Initial value for Root policy –** 15 • **Applies to –** All users granted the MANAGE ANY USER system privilege.

See also

- *Creating a New Login Policy* on page 62
- *Displaying a List of Users Assigned a Login Policy* on page 63
- *Modifying the Root Login Policy* on page 62
- *Modifying an Existing Login Policy* on page 63
- *Assigning a Login Policy When Creating a New User* on page 64

ALTER LS POLICY Statement

Modifies some or all option values for the root logical server policy or a user-created logical server policy.

Syntax

```
ALTER LS POLICY policy-name option-value-list [ WITH STOP SERVER ]
```

Parameters

- **option-value-list –** { *option-name* = *value* } ...
- **option-name –**

```
 ALLOW_COORDINATOR_AS_MEMBER = < ON | OFF >
| DQP_ENABLED = < 0 | 1 | 2 >
| LOGIN_REDIRECTION = < ON | OFF >
| REDIRECTION_WAITERS_THRESHOLD = < num >
| TEMP_DATA_IN_SHARED_TEMP = < ON | OFF >
```

Applies to

Multiplex only.

Examples

- **Example 1 –** alters the logical server policy:

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

- **Example 2 –** alters the logical server policy and causes servers to shut down automatically when the option value changes:

```
ALTER LS POLICY root
TEMP_DATA_IN_SHARED_TEMP=ON WITH STOP SERVER
```

Usage

The option values listed are modified for the new policy. Any unspecified option inherits its value from the root logical server policy.

Table 10. Logical Server Policy Options

Logical Server Policy Option	Allowed Values	Behavior
ALLOW_COORDINATOR_AS_MEMBER	ON, OFF	Can only be set for the ROOT logical server policy. When ON (the default), the coordinator can be a member of any user-defined logical server. OFF prevents the coordinator from being used as a member of any user-defined logical servers.
DQP_ENABLED	0, 1, 2	When set to 0, query processing is not distributed. When set to 1 (the default), query processing is distributed as long as a writable shared temporary file exists. When set to 2, query processing is distributed over the network, and the shared temporary store is not used.
LOGIN_REDIRECTION	ON, OFF	When ON, enables login redirection for logical servers governed by specified login policy. When OFF (the default), disables login redirection at the logical server level, allowing external connection management.
REDIRECTION_WAITERS_THRESHOLD	Integer	Specifies how many connections can queue before SAP Sybase IQ redirects a connection to this logical server to another server. Can be any integer value; default is 5.
TEMP_DATA_IN_SHARED_TEMP	ON, OFF	When ON, all temporary table data and eligible scratch data writes to the shared temporary store, provided that the shared temporary store has at least one read-write file added. You must restart all multiplex nodes after setting this option or after adding a read-write file to the shared temporary store. (If the shared temporary store contains no read-write file, or if you do not restart nodes, data is written to IQ_SYSTEM_TEMP instead.) When OFF (the default), all temporary table data and scratch data writes to the local temporary store.

If you want a smaller IQ_SYSTEM_TEMP dbspace, turn `TEMP_DATA_IN_SHARED_TEMP` ON, which writes temporary data to IQ_SHARED_TEMP instead of IQ_SYSTEM_TEMP. In a distributed query processing environment, however, setting both `DQP_ENABLED` and `TEMP_DATA_IN_SHARED_TEMP` ON may saturate your SAN with additional data in

IQ_SHARED_TEMP, where additional I/O operations against IQ_SHARED_TEMP may adversely affect DQP performance.

WITH STOP SERVER automatically shuts down all servers in the logical server when the `TEMP_DATA_IN_SHARED_TEMP` option is changed directly or indirectly.

This statement enforces consistent shared system temporary store settings across physical nodes shared by logical servers.

Permissions

Requires the MANAGE MULTIPLEX system privilege.

See also

- *Configure Logical Server Policies* on page 56
- *Altering Root Logical Server Policy* on page 56
- *Coordinator Failure* on page 26
- *Global Transaction Resiliency* on page 42
- *Dropping Multiplex Servers* on page 21

ALTER MULTIPLEX RENAME Statement

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

Syntax

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

Applies to

Multiplex only.

Usage

When a multiplex is created, it is named after the coordinator. This statement is automatically committed.

Permissions

Requires the MANAGE MULTIPLEX system privilege.

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 server-name 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: 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.

Applies to

Multiplex only.

Examples

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

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

Usage

Changes the multiplex server, as follows:

- **RENAME –** changes the name of the given server. The server automatically shuts down and the next restart requires the new name.
- **DATABASE –** changes the catalog file path for the given server. The server automatically shuts down and the next restart requires the new catalog path. The user must relocate the catalog file.
- **ROLE –** changes the role of the given server. Users cannot change the role of coordinator or role to coordinator. If the role of the writer node changes 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 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. (Run all DDL statements on the coordinator.) In all cases except when altering role from reader to writer, the named server is automatically shut down.

Note: 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.

Permissions

Requires the MANAGE MULTIPLEX system privilege.

See also

- *Renaming Multiplex Servers* on page 23

BACKUP Statement

Backs up an SAP Sybase IQ database on one or more archive devices.

Syntax

```
BACKUP DATABASE
[ backup-option… ]
TO archive_device [ archive-option... ]
… [ WITH COMMENT string ]
```

Parameters

- **backup-option –**

```
{ READWRITE FILES ONLY |
READONLY dbspace-or-file [, … ] }
CRC { ON | OFF }
ATTENDED { ON | OFF }
BLOCK FACTOR integer
{ FULL | INCREMENTAL | INCREMENTAL SINCE FULL }
VIRTUAL { DECOUPLED |
ENCAPSULATED 'shell_command' }
WITH COMMENT comment
```

- **dbspace-or-file –**

```
{ DBSPACES identifier-list | FILES identifier-list }
```

- **identifier-list –**

```
identifier [, … ]
```

- **archive-option –**

```
SIZE integer STACKER integer
```

Examples

- **Example 1 –** This UNIX example backs up the `iqdemo` database onto tape devices `/dev/rmt/0` and `/dev/rmt/2` on a Sun Solaris platform. On Solaris, the letter `n` after the device name specifies the “no rewind on close” feature. Always specify this feature with **BACKUP**, using the naming convention appropriate for your UNIX platform (Windows does not support this feature). This example backs up all changes to the database since the last full backup:

```
BACKUP DATABASE
INCREMENTAL SINCE FULL
TO '/dev/rmt/0n' SIZE 10000000
TO '/dev/rmt/2n' SIZE 15000000
```

Note: Size units are kilobytes (KB), although in most cases, size of less than 1GB are inappropriate. In this example, the specified sizes are 10GB and 15GB.

- **Example 2 –** These **BACKUP** commands specify read-only files and dbspaces:

```
BACKUP DATABASE READONLY DBSPACES dsp1
TO '/dev/rmt/0'
```

```
BACKUP DATABASE READONLY FILES dsp1_f1, dsp1_f2
TO 'bkp.f1f2'
```

```
BACKUP DATABASE READONLY DBSPACES dsp2, dsp3
READONLY FILES dsp4_f1, dsp5_f2
TO 'bkp.RO'
```

Usage

The SAP Sybase IQ database might be open for use by many readers and writers when you execute a **BACKUP** command. It acts as a read-only user and relies on the Table Level Versioning feature of SAP Sybase IQ to achieve a consistent set of data.

BACKUP implicitly issues a **CHECKPOINT** prior to commencing, and then it backs up the catalog tables that describe the database (and any other tables you have added to the catalog store). During this first phase, SAP Sybase IQ does not allow any metadata changes to the database (such as adding or dropping columns and tables). Correspondingly, a later **RESTORE** of the backup restores only up to that initial **CHECKPOINT**.

The **BACKUP** command lets you specify full or incremental backups. You can choose two kinds of incremental backups. **INCREMENTAL** backs up only those blocks that have changed and committed since the last **BACKUP** of any type (incremental or full). **INCREMENTAL SINCE FULL** backs up all of the blocks that have changed since the last full backup. The first

type of incremental backup can be smaller and faster to do for **BACKUP** commands, but slower and more complicated for **RESTORE** commands. The opposite is true for the other type of incremental backup. The reason is that the first type generally results in N sets of incremental backup archives for each full backup archive. If a restore is required, a user with the SERVER OPERATOR system privilege must **RESTORE** the full backup archive first, and then each incremental archive in the proper order. (SAP Sybase IQ keeps track of which ones are needed.) The second type requires the user with the SERVER OPERATOR system privilege to restore only the full backup archive and the last incremental archive.

Incremental virtual backup is supported using the **VIRTUAL DECOUPLED** and **VIRTUAL ENCAPSULATED** parameters of the **BACKUP** statement.

Although you can perform an OS-level copy of tablespaces to make a virtual backup of one or more read-only dbspaces, use the virtual backup statement, because it records the backup in the SAP Sybase IQ system tables.

READWRITE FILES ONLY may be used with **FULL**, **INCREMENTAL**, and **INCREMENTAL SINCE FULL** to restrict the backup to only the set of read-write files in the database. The read-write dbspaces/files must be SAP Sybase IQ dbspaces.

If **READWRITE FILES ONLY** is used with an **INCREMENTAL** or **INCREMENTAL SINCE FULL** backup, the backup will not back up data on read-only dbspaces or dbfiles that has changed since the depends-on backup. If **READWRITE FILES ONLY** is not specified for an **INCREMENTAL** or **INCREMENTAL SINCE FULL** backup, the backup backs up all database pages that have changed since the depends-on backup, both on read-write and read-only dbspaces.

- **CRC clause –** activate 32-bit cyclical redundancy checking on a per block basis (in addition to whatever error detection is available in the hardware). When you specify this clause, the numbers computed on backup are verified during any subsequent **RESTORE** operation, affecting performance of both commands. The default is ON.
- **ATTENDED clause –** applies only when backing up to a tape device. If **ATTENDED ON** (the default) is used, a message is sent to the application that issued the **BACKUP** statement if the tape drive requires intervention. This might happen, for example, when a new tape is required. If you specify OFF, **BACKUP** does not prompt for new tapes. If additional tapes are needed and OFF has been specified, SAP Sybase IQ gives an error and aborts the **BACKUP** command. However, a short delay is included to account for the time an automatic stacker drive requires to switch tapes.
- **BLOCK FACTOR clause –** specify the number of blocks to write at one time. Its value must be greater than 0, or SAP Sybase IQ generates an error message. Its default is 25 for UNIX systems and 15 for Windows systems (to accommodate the smaller fixed tape block sizes). This clause effectively controls the amount of memory used for buffers. The actual amount of memory is this value times the block size times the number of threads used to extract data from the database. Set **BLOCK FACTOR** to at least 25.

- **FULL clause –** specify a full backup; all blocks in use in the database are saved to the archive devices. This is the default action.
- **INCREMENTAL clause –** specify an incremental backup; all blocks changed since the last backup of any kind are saved to the archive devices.

  The keyword **INCREMENTAL** is not allowed with **READONLY FILES**.
- **INCREMENTAL SINCE FULL clause –** specify an incremental backup; all blocks changed since the last full backup are saved to the archive devices.
- **VIRTUAL DECOUPLED clause –** specify a decoupled virtual backup. For the backup to be complete, you must copy the SAP Sybase IQ dbspaces after the decoupled virtual backup finishes, and then perform a nonvirtual incremental backup.
- **VIRTUAL ENCAPSULATED clause –** specify an encapsulated virtual backup. The 'shell-command' argument can be a string or variable containing a string that is executed as part of the encapsulated virtual backup. The shell commands execute a system-level backup of the IQ store as part of the backup operation. For security reasons, it is recommended that an absolute path be specified in the 'shell-command,' and file protections on that directory be in place to prevent execution of an unintended program.
- **TO clause –** specify the name of the archive_device to be used for backup, delimited with single quotation marks. The archive_device is a file name or tape drive device name for the archive file. If you use multiple archive devices, specify them using separate **TO** clauses. (A comma-separated list is not allowed.) Archive devices must be distinct. The number of **TO** clauses determines the amount of parallelism SAP Sybase IQ attempts with regard to output devices.

BACKUP and **RESTORE** write your SAP Sybase IQ data in parallel to or from all of the archive devices you specify. The catalog store is written serially to the first device. Faster backups and restores result from greater parallelism.

SAP Sybase IQ supports a maximum of 36 hardware devices for backup. For faster backups, specifying one or two devices per core will help to avoid hardware and IO contention. Set the **SIZE** parameter on the **BACKUP** command to avoid creating multiple files per backup device and consider the value used in the **BLOCK FACTOR** clause on the **BACKUP** command.

BACKUP overwrites existing archive files unless you move the old files or use a different *archive_device* name or path.

The backup API DLL implementation lets you specify arguments to pass to the DLL when opening an archive device. For third-party implementations, the archive_device string has this format:

```
'DLLidentifier::vendor_specific_information'
```

A specific example:

```
'spsc::workorder=12;volname=ASD002'
```

The *archive_device* string length can be up to 1023 bytes. The *DLLidentifier* portion must be 1 to 30 bytes in length and can contain only alphanumeric and underscore characters. The

vendor_specific_information portion of the string is passed to the third-party implementation without checking its contents. Do not specify the **SIZE** or **STACKER** clauses of the **BACKUP** command when using third-party implementations, as that information should be encoded in the *vendor_specific_information* portion of the string.

Note: Only certain third-party products are certified with SAP Sybase IQ using this syntax. See the *Release Bulletin* for additional usage instructions or restrictions. Before using any third-party product to back up your SAP Sybase IQ database in this way, make sure it is certified. See the *Release Bulletin*, or see the SAP Sybase Certification Reports for the SAP Sybase IQ product in *Technical Documents at http://www.sybase.com/support/techdocs/*.

For the Sybase implementation of the backup API, you need to specify only the tape device name or file name. For disk devices, you should also specify the SIZE value, or SAP Sybase IQ assumes that each created disk file is no larger than 2GB on UNIX, or 1.5GB on Windows. An example of an archive device for the SAP Sybase API DLL that specifies a tape device for certain UNIX systems is:

```
'/dev/rmt/0'
```

- **SIZE clause –** specify maximum tape or file capacity per output device (some platforms do not reliably detect end-of-tape markers). No volume used on the corresponding device should be shorter than this value. This value applies to both tape and disk files but not third-party devices.

  Units are kilobytes (KB), although in general, less than 1GB is inappropriate. For example, for a 3.5GB tape, specify 3500000. Defaults are by platform and medium. The final size of the backup file will not be exact, because backup writes in units of large blocks of data.

  The **SIZE** parameter is per output device. **SIZE** does not limit the number of bytes per device; **SIZE** limits the file size. Each output device can have a different **SIZE** parameter.

  During backup, when the amount of information written to a given device reaches the value specified by the **SIZE** parameter, **BACKUP** does one of the following:

  - If the device is a file system device, **BACKUP** closes the current file and creates another file of the same name, with the next ascending number appended to the file name, for example, `bkup1.dat1.1`, `bkup1.dat1.2`, `bkup1.dat1.3`.
  - If the device is a tape unit, **BACKUP** closes the current tape and you need to mount another tape.

It is your responsibility to mount additional tapes if needed, or to ensure that the disk has enough space to accommodate the backup.

When multiple devices are specified, **BACKUP** distributes the information across all devices.

Table 11. BACKUP default sizes

Platform	Default SIZE for tape	Default SIZE for disk
UNIX	none	2GB
Windows	1.5GB **SIZE** must be a multiple of 64. Other values are rounded down to a multiple of 64.	1.5GB

- **STACKER clause –** specify that the device is automatically loaded, and specifies the number of tapes with which it is loaded. This value is not the tape position in the stacker, which could be zero. When **ATTENDED** is OFF and **STACKER** is ON, SAP Sybase IQ waits for a predetermined amount of time to allow the next tape to be autoloaded. The number of tapes supplied along with the **SIZE** clause are used to determine whether there is enough space to store the backed-up data. Do not use this clause with third-party media management devices.
- **WITH COMMENT clause –** specify an optional comment recorded in the archive file and in the backup history file. Maximum length is 32KB. If you do not specify a value, a NULL string is stored.

Other issues for **BACKUP** include:

- **BACKUP** does not support raw devices as archival devices.
- Windows systems support only fixed-length I/O operations to tape devices (for more information about this limitation, see your *Installation and Configuration Guide*). Although Windows supports tape partitioning, SAP Sybase IQ does not use it, so do not use another application to format tapes for **BACKUP**. Windows has a simpler naming strategy for its tape devices, where the first tape device is `\\.\tape0`, the second is `\\.\tape1`, and so on.

  Warning! For backup (and for most other situations) SAP Sybase IQ treats the leading backslash in a string as an escape character, when the backslash precedes an n, an x, or another backslash. For this reason, when you specify backup tape devices, you must double each backslash required by the Windows naming convention. For example, indicate the first Windows tape device you are backing up to as '`\\\\.\\tape0`', the second as '`\\\\.\\tape1`', and so on. If you omit the extra backslashes, or otherwise misspell a tape device name, and write a name that is not a valid tape device on your system, SAP Sybase IQ interprets this name as a disk file name.
- SAP Sybase IQ does not rewind tapes before using them. You must ensure the tapes used for **BACKUP** or **RESTORE** are at the correct starting point before putting them in the tape device. SAP Sybase IQ does rewind tapes after using them on rewinding devices.
- During **BACKUP** and **RESTORE** operations, if SAP Sybase IQ cannot open the archive device (for example, when it needs the media loaded) and the **ATTENDED** parameter is

ON, it waits for ten seconds and tries again. It continues these attempts indefinitely until either it is successful or the operation is terminated with a Ctrl+C.
- If you enter Ctrl+C, **BACKUP** fails and returns the database to the state it was in before the backup started.
- If disk striping is used, such as on a RAID device, the striped disks are treated as a single device.

Side effects:

- Automatic commit

Standards

- SQL—Vendor extension to ISO/ANSI SQL grammar.
- Sybase—Not supported by Adaptive Server Enterprise.

Permissions

Requires one of:

- BACK UP DATABASE system privilege.
- You own the database.

See also

- *Backup Requirements* on page 71
- *Backing Up Multiplex Databases* on page 72

COMMENT ON LOGICAL SERVER Statement

Comments on the user-defined logical server.

Syntax

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

Examples

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

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

Usage

Multiplex only.

Permissions

Must have the MANAGE MULTIPLEX system privilege.

See also

- *Commenting on Logical Servers* on page 53

CREATE LOGICAL SERVER Statement

Creates a user-defined logical server.

Syntax

```
CREATE LOGICAL SERVER logical-server-name [
     { ls-create-clause, ...} ] [ WITH STOP SERVER ]
```

Parameters

- **ls-create-clause –** { **MEMBERSHIP** ({ *ls-member*, ...}) | **POLICY** *ls-policy-name* }
- **ls-member – FOR LOGICAL COORDINATOR** | *mpx-server-name*

Applies to

Multiplex only.

Examples

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

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

- **Example 2 –** creates a user-defined logical server *ls1* with three member nodes, and defines the logical server policy name *lsp1*.

  ```
  CREATE LOGICAL SERVER ls1 MEMBERSHIP ( w1_svr, w2_svr, r2_svr )
  POLICY lsp1
  ```

- **Example 3 –** creates servers as in Example 2, except that **WITH STOP SERVER** automatically shuts down all servers in the logical server when the `TEMP_DATA_IN_SHARED_TEMP` option is changed directly or indirectly.

  ```
  CREATE LOGICAL SERVER ls1 MEMBERSHIP ( w1_svr, w2_svr, r2_svr )
  POLICY lsp1 WITH STOP SERVER
  ```

- **Example 4 –** creates a user-defined logical server *ls1* with logical server policy *lspolicy1* and no member nodes.

  ```
  CREATE LOGICAL SERVER ls1 POLICY lspolicy1
  ```

- **Example 5 –** where *n1* is the current coordinator, creates a logical server *ls2* with the named membership of multiplex nodes *n1* and *n3* and logical membership of the coordinator. Also sets the logical server policy of *ls2* to *lspolicy2*.

```
CREATE LOGICAL SERVER ls2 POLICY
MEMBERSHIP FOR LOGICAL COORDINATOR
lspolicy1, n1, n2, n3 POLICY lspolicy2
```

Usage

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. To define a logical server policy name, specify **POLICY**.

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.ISYSIQLOGICALSERVER` system table stores information about the logical server policy for a corresponding logical server.

The `SYS.ISYSLOGICALMEMBER` system table stores definitions for the logical server memberships.

ls-policy-name can be any user-specified identifier except ROOT.

The **POLICY** clause associates a logical server with a user-defined logical server policy. If no **POLICY** clause is specified, the logical server is associated with the root policy.

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

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

Changing the `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 affects only 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 the **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.

WITH STOP SERVER automatically shuts down all servers in the logical server when the `TEMP_DATA_IN_SHARED_TEMP` option is changed directly or indirectly.

This statement enforces consistent shared system temporary store settings across physical nodes shared by logical servers.

Permissions

Requires the MANAGE MULTIPLEX system privilege.

See also

- *Creating a Logical Server* on page 52

CREATE LOGIN POLICY Statement

Creates a login policy in the database.

Syntax

```
CREATE LOGIN POLICY policy-name
AUTO_UNLOCK_TIME=0 - UNLIMITED
| DEFAULT_LOGICAL_SERVER=[logical_server_name | ALL | AUTO | COORDINATOR |
NONE | OPEN | SERVER]
| CHANGE_PASSWORD_DUAL_CONTROL=[ON | OFF]
| LOCKED=[ON | OFF]
| MAX_CONNECTIONS=0 - 2147483647
| MAX_DAYS_SINCE_LOGIN=0 - 2147483647
| MAX_FAILED_LOGIN_ATTEMPTS=0 - 2147483647
| MAX_NON_DBA_CONNECTIONS=0 - 2147483647
| PASSWORD_EXPIRY_ON_NEXT_LOGIN=[ON | OFF]
| PASSWORD_GRACE_TIME=0 - 2147483647
| PASSWORD_LIFE_TIME=0 - 2147483647
| ROOT_AUTO_UNLOCK_TIME=0 - UNLIMITED
| LDAP_PRIMARY_SERVER=server_name
| LDAP_SECONDARY_SERVER=server_name
| LDAP_AUTO_FAILBACK_PERIOD=0 - 2147483647
| LDAP_FAILOVER_TO_STD=[ON | OFF]
| LDAP_REFRESH_DN=NOW
```

Applies to

Simplex and multiplex.

Examples

- **Example 1 –** creates the `Test1` login policy. This login policy has an unlimited password life and allows the user a maximum of five attempts to enter a correct password before the account is locked.

  ```
  CREATE LOGIN POLICY Test1
  password_life_time=UNLIMITED
  max_failed_login_attempts=5;
  ```

Usage

If you do not specify a login policy option, the value from the root login policy is applied.

Permissions

Requires MANAGE ANY LOGIN POLICY system privilege.

The following system privileges can override the noted login policy options:

Exception System Privilege	Login Policy Option
SERVER OPERATOR or DROP CONNECTION system privilege	MAX_NON_DBA_CONNS MAX_CONNECTIONS
MANAGE ANY USER system privilege	LOCKED MAX_DAYS_SINCE_LOGIN

See also

- *Creating a New Login Policy* on page 62
- *Displaying a List of Users Assigned a Login Policy* on page 63
- *Modifying the Root Login Policy* on page 62
- *Modifying an Existing Login Policy* on page 63
- *Assigning a Login Policy When Creating a New User* on page 64

Login Policy Options

Available options for root and user-defined login policies.

Option	Description
AUTO_UNLOCK_TIME	The time period after which locked accounts not granted the MANAGE ANY USER system privilege are automatically unlocked. This option can be defined in any login policy, including the root login policy. • **Values –** 0 – unlimited • **Initial value for Root policy –** Unlimited • **Applies to –** All users not granted the MANAGE ANY USER system privilege.
CHANGE_PASSWORD_DUAL_CONTROL	Requires input from two users, each granted the CHANGE PASSWORD system privilege, to change the password of another user. • **Values –** ON, OFF • **Initial value for Root policy –** OFF • **Applies to –** All users.

Option	Description

DEFAULT_LOGICAL_SERVER	If the connection string specifies no logical server, the user connects to the `DEFAULT_LOGICAL_SERVER` setting specified in the user's login policy.
	• **Values –**
	  • Name of an existing user-defined logical server
	  • ALL – allows access to all logical servers.
	  • AUTO – value of the default logical server in the root login policy.
	  • COORDINATOR – the current coordinator node.
	  • NONE – denies access to any multiplex server.
	  • OPEN – use alone or with the name of a user-defined logical server. Allows access to all multiplex nodes that are not members of any user-defined logical servers.
	  • SERVER – allows access to all of the multiplex nodes, subject to the semantics of the SERVER logical server.
	• **Initial value for Root policy –** AUTO
	• **Applies to –** All users. Requires MANAGE MULTIPLEX system privilege.

LOCKED	If set ON, users cannot establish new connections. This setting temporarily denies access to login policy users. Logical server overrides for this option are not allowed. • **Values –** ON, OFF • **Initial value for Root policy –** OFF • **Applies to –** All users except those with the MANAGE ANY USER system privilege.
MAX_CONNECTIONS	The maximum number of concurrent connections allowed for a user. You can specify a per-logical-server setting for this option. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** Unlimited • **Applies to –** All users except those with the SERVER OPERATOR or DROP CONNECTION system privilege.

Option	Description
MAX_DAYS_SINCE_LOGIN	The maximum number of days that can elapse between two successive logins by the same user. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** Unlimited • **Applies to –** All users except those with the MANAGE ANY USER system privilege.
MAX_FAILED_LOG-IN_ATTEMPTS	The maximum number of failed attempts, since the last successful attempt, to log into the user account before the account is locked. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** Unlimited • **Applies to –** All users.
MAX_NON_DBA_CONNECTIONS	The maximum number of concurrent connections that a user without SERVER OPERATOR or DROP CONNECTION system privileges can make. This option is supported only in the root login policy. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** Unlimited • **Applies to –** All users except those with the SERVER OPERATOR or DROP CONNECTION privilege.
PASSWORD_EXPI-RY_ON_NEXT_LOG-IN	If set ON, the user's password expires at the next login. • **Values –** ON, OFF • **Initial value for Root policy –** OFF • **Applies to –** All users. **Note:** This functionality is not currently implemented when logging in to Sybase Control Center. A user will not be prompted to change their password. He or she will be prompted, however, when logging in to SAP Sybase IQ outside of Sybase Control Center (for example, using Interactive SQL).
PASS-WORD_GRACE_TIME	The number of days before password expiration during which login is allowed but the default post_login procedure issues warnings. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** 0 • **Applies to –** All users.

Option	Description
PASS-WORD_LIFE_TIME	The maximum number of days before a password must be changed. • **Values –** 0 – 2147483647 • **Initial value for Root policy –** Unlimited • **Applies to –** All users.
ROOT_AUTO_UN-LOCK_TIME	The time period after which locked accounts granted the MANAGE ANY USER system privilege are automatically unlocked. This option can be defined only in the root login policy. • **Values –** 0 – unlimited • **Initial value for Root policy –** 15 • **Applies to –** All users granted the MANAGE ANY USER system privilege.

Multiplex Login Policy Configuration

Configure login policies for multiplex servers.

Example

This example overrides the login policy settings on a logical server, increasing the maximum number of connections on logical server `ls1`:

```
ALTER LOGIN POLICY lp1 max_connections=20 LOGICAL SERVER ls1;
```

Usage

Applies only to multiplex.

Any login management commands you execute on any multiplex server automatically propagate to all servers in the multiplex. For best performance, execute these commands, or any DDL, on the coordinator.

An override at the 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, a corresponding row exists in `ISYSIQLSLOGINPOLICYOPTION`.

CREATE LS POLICY Statement

Creates a user-defined logical server policy.

Syntax

```
CREATE LS POLICY policy-name option-value-list [ WITH STOP SERVER ]
```

Parameters

- **option-value-list –**

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

- **option-name –**

```
 ALLOW_COORDINATOR_AS_MEMBER = < ON | OFF>
| DQP_ENABLED = < 0 | 1 | 2>
| LOGIN_REDIRECTION = < ON | OFF>
| REDIRECTION_WAITERS_THRESHOLD = < num >
| TEMP_DATA_IN_SHARED_TEMP = < ON | OFF >
```

Applies to

Multiplex only.

Examples

- **Example 1 –** creates a user-defined logical server policy named *lspolicy1*:

```
CREATE LS POLICY lspolicy1
ALLOW_COORDINATOR_AS_MEMBER=ON;
```

Usage

You can specify any identifier except ROOT for the policy name.

The option values listed are modified for the new policy. Any unspecified option inherits its value from the root logical server policy.

If you want a smaller IQ_SYSTEM_TEMP dbspace, turn ON the option `TEMP_DATA_IN_SHARED_TEMP` to write temporary data to IQ_SHARED_TEMP instead of IQ_SYSTEM_TEMP. In a distributed query processing environment, however, setting both `DQP_ENABLED` and `TEMP_DATA_IN_SHARED_TEMP` ON could saturate your SAN with additional data in IQ_SHARED_TEMP, where additional I/O operations against IQ_SHARED_TEMP may adversely affect DQP performance.

WITH STOP SERVER automatically shuts down all servers in the logical server when the `TEMP_DATA_IN_SHARED_TEMP` option is changed directly or indirectly.

This statement enforces consistent shared system temporary store settings across physical nodes shared by logical servers.

Table 12. Logical Server Policy Options

Logical Server Policy Option	Allowed Values	Behavior
ALLOW_COORDINATOR_AS_MEMBER	ON, OFF	Can only be set for the ROOT logical server policy. When ON (the default), the coordinator can be a member of any user-defined logical server. OFF prevents the coordinator from being used as a member of any user-defined logical servers.
DQP_ENABLED	0, 1, 2	When set to 0, query processing is not distributed. When set to 1 (the default), query processing is distributed as long as a writable shared temporary file exists. When set to 2, query processing is distributed over the network, and the shared temporary store is not used.
LOGIN_REDIRECTION	ON, OFF	When ON, enables login redirection for logical servers governed by specified login policy. When OFF (the default), disables login redirection at the logical server level, allowing external connection management.
REDIRECTION_WAITERS_THRESHOLD	Integer	Specifies how many connections can queue before SAP Sybase IQ redirects a connection to this logical server to another server. Can be any integer value; default is 5.
TEMP_DATA_IN_SHARED_TEMP	ON, OFF	When ON, all temporary table data and eligible scratch data writes to the shared temporary store, provided that the shared temporary store has at least one read-write file added. You must restart all multiplex nodes after setting this option or after adding a read-write file to the shared temporary store. (If the shared temporary store contains no read-write file, or if you do not restart nodes, data is written to IQ_SYSTEM_TEMP instead.) When OFF (the default), all temporary table data and scratch data writes to the local temporary store.

Standards

- SQL – vendor extension to ISO/ANSI SQL grammar.
- Sybase – not supported by Adaptive Server Enterprise.

Permissions

Requires the MANAGE MULTIPLEX system privilege.

See also

- *Configure Logical Server Policies* on page 56
- *Altering Root Logical Server Policy* on page 56

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* }

Applies to

Multiplex only.

Usage

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**.

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.

After creating the first secondary node, the coordinator shuts down automatically.

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. SAP 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

Requires the MANAGE MULTIPLEX system privilege.

See also

- *Adding Multiplex Servers* on page 19

DROP LOGICAL SERVER Statement

Drops a user-defined logical server.

Syntax

```
DROP LOGICAL SERVER logical-server-name
 [ WITH STOP SERVER ]
```

Applies to

Multiplex only.

Examples

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

```
DROP LOGICAL SERVER ls1
```

Usage

SAP 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, SAP Sybase IQ sets the logical server assignment for the login policy to NONE.
- Removes the logical server entry from ISYSIQ.LOGICALSERVER.

WITH STOP SERVER automatically shuts down all servers in the logical server when the `TEMP_DATA_IN_SHARED_TEMP` option is changed directly or indirectly.

This statement enforces consistent shared system temporary store settings across physical nodes shared by logical servers.

Permissions

Requires the MANAGE MULTIPLEX system privilege.

See also

- *Dropping a Logical Server* on page 54

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** }

Applies to

Multiplex only.

Examples

- **Example 1 –**

```
DROP MULTIPLEX SERVER writer1
```

Usage

Shut down each multiplex server before dropping it. This statement automatically commits.

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.

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 multiplex 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

Requires the MANAGE MULTIPLEX system privilege.

See also

- *Coordinator Failure* on page 26
- *Global Transaction Resiliency* on page 42
- *Dropping Multiplex Servers* on page 21

RESTORE DATABASE Statement

Restores an SAP Sybase IQ database backup from one or more archive devices.

Syntax

Syntax 1

```
RESTORE DATABASE 'db_file'
FROM 'archive_device' [ FROM 'archive_device' ]…
… [ CATALOG ONLY ]
… [ KEY key_spec ]
… [ [ RENAME logical-dbfile-name TO 'new-dbspace-path']...
     | VERIFY [ COMPATIBLE ] ]
```

Syntax 2

```
RESTORE DATABASE 'database-name'
[ restore-option ... ]
FROM 'archive_device' ...
```

Parameters

- **db_file –** relative or absolute path of the database to be restored. Can be the original location, or a new location for the catalog store file.
- **key_spec –** quoted string including mixed cases, numbers, letters, and special characters. It might be necessary to protect the key from interpretation or alteration by the command shell.
- **restore-option –**

```
READONLY dbspace-or-file [, … ]
KEY key_spec
RENAME file-name TO new-file-path ...
```

Examples

- **Example 1 –** This UNIX example restores the `iqdemo` database from tape devices `/dev/rmt/0` and `/dev/rmt/2` on a Sun Solaris platform. On Solaris, a RESTORE from tape must specify the use of the rewinding device. Therefore, do not include the letter 'n' after the device name, which specifies "no rewind on close." To specify this feature with **RESTORE**, use the naming convention appropriate for your UNIX platform. (Windows does not support this feature.)

```
RESTORE DATABASE 'iqdemo'
FROM '/dev/rmt/0'
FROM '/dev/rmt/2'
```

- **Example 2 –** Restore an encrypted database named `marvin` that was encrypted with the key *is!seCret*.

```
RESTORE DATABASE 'marvin'
FROM 'marvin_bkup_file1'
FROM 'marvin_bkup_file2'
FROM 'marvin_bkup_file3'
KEY 'is!seCret'
```

- **Example 3 –** This example shows the syntax of a **BACKUP** statement and two possible **RESTORE** statements. (This example uses objects in the `iqdemo` database for illustration purposes. Note that `iqdemo` includes a sample user dbspace named `iq_main` that may not be present in your database.)

  Given this **BACKUP** statement:

```
BACKUP DATABASE READONLY DBSPACES iq_main
TO '/system1/IQ16/demo/backup/iqmain'
```

  The dbspace `iq_main` can be restored using either of these **RESTORE** statements:

```
RESTORE DATABASE 'iqdemo' READONLY DBSPACES iq_main
FROM '/system1/IQ16/demo/backup/iqmain'
```

  or

```
RESTORE DATABASE 'iqdemo'
FROM '/system1/IQ16/demo/backup/iqmain'
```

A selective backup backs up either all READWRITE dbspaces or specific read-only dbspaces or dbfiles. Selective backups are a subtype of either full or incremental backups.

Notes:

- You can take a **READONLY** selective backup and restore all objects from this backup (as in the second example above).
- You can take an all-inclusive backup and restore read-only files and dbspaces selectively.
- You can take a **READONLY** selective backup of multiple read-only files and dbspaces and restore a subset of read-only files and dbspaces selectively. See *Permissions*.
- You can restore the read-only backup, only if the read-only files have not changed since the backup. Once the dbspace is made read-write again, the read-only backup is invalid, unless you restore the entire read-write portion of the database back to the point at which the read-only dbspace was read-only.
- Decide which backup subtype to use (either selective or non-selective) and use it consistently. If you must switch from a non-selective to a selective backup, or vice versa, always take a non-selective full backup before switching to the new subtype, to ensure that you have all changes.

- **Example 4 –** Syntax to validate the database archives using the **VERIFY** clause, without performing any write operations:

```
RESTORE DATABASE <database_name.db>
FROM '/sys1/dump/dmp1'
FROM '/sys1/dump/dmp2'
VERIFY
```

When you use validate, specify a different database name to avoid `Database name not unique` errors. If the original database is `iqdemo.db`, for example, use `iq_demo_new.db` instead:

```
RESTORE DATABASE iqdemo_new.db FROM iqdemo.bkp VERIFY
```

Usage

The **RESTORE** command requires exclusive access by a user with the SERVER OPERATOR system privilege to the database. This exclusive access is achieved by setting the **-gd** switch to `DBA`, which is the default when you start the server engine.

Issue the **RESTORE** command before you start the database (you must be connected to the `utility_db` database). Once you finish specifying **RESTORE** commands for the type of backup, that database is ready to be used. The database is left in the state that existed at the end of the first implicit **CHECKPOINT** of the last backup you restored. You can now specify a **START DATABASE** to allow other users to access the restored database.

The maximum size for a complete **RESTORE** command, including all clauses, is 32KB.

When restoring to a raw device, make sure the device is large enough to hold the dbspace you are restoring. SAP Sybase IQ **RESTORE** checks the raw device size and returns an error, if the raw device is not large enough to restore the dbspace.

BACKUP allows you to specify full or incremental backups. There are two kinds of incremental backups. **INCREMENTAL** backs up only those blocks that have changed and committed since the last backup of any type (incremental or full). **INCREMENTAL SINCE FULL** backs up all the blocks that have changed since the last full backup. If a **RESTORE** of a full backup is followed by one or more incremental backups (of either type), no modifications to the database are allowed between successive **RESTORE** commands. This rule prevents a **RESTORE** from incremental backups on a database in need of crash recovery, or one that has been modified. You can still overwrite such a database with a **RESTORE** from a full backup.

Before starting a full restore, you must delete two files: the catalog store file (default name `dbname.db`) and the transaction log file (default name `dbname.log`).

If you restore an incremental backup, **RESTORE** ensures that backup media sets are accessed in the proper order. This order restores the last full backup tape set first, then the first incremental backup tape set, then the next most recent set, and so forth, until the most recent incremental backup tape set. If a user with the SERVER OPERATOR system privilege produced an **INCREMENTAL SINCE FULL** backup, only the full backup tape set and the most recent **INCREMENTAL SINCE FULL** backup tape set is required; however, if there is an **INCREMENTAL** backup made since the **INCREMENTAL SINCE FULL** backup, it also must be applied.

SAP Sybase IQ ensures that the restoration order is appropriate, or it displays an error. Any other errors that occur during the restore results in the database being marked corrupt and unusable. To clean up a corrupt database, do a **RESTORE** from a full backup, followed by any additional incremental backups. Since the corruption probably happened with one of those backups, you might need to ignore a later backup set and use an earlier set.

To restore read-only files or dbspaces from an archive backup, the database may be running and the administrator may connect to the database when issuing the **RESTORE** statement. The read-only file pathname need not match the names in the backup, if they otherwise match the database system table information.

The database must not be running to restore a **FULL**, **INCREMENTAL SINCE FULL**, or **INCREMENTAL** restore of either a **READWRITE FILES ONLY** or an all files backup. The database may or may not be running to restore a backup of read-only files. When restoring specific files in a read-only dbspace, the dbspace must be offline. When restoring read-only files in a read-write dbspace, the dbspace can be online or offline. The restore closes the read-only files, restores the files, and reopens those files at the end of the restore.

You can use selective restore to restore a read-only dbspace, as long as the dbspace is still in the same read-only state.

FROM—Specifies the name of the *archive_device* from which you are restoring, delimited with single quotation marks. If you are using multiple archive devices, specify them using

separate **FROM** clauses. A comma-separated list is not allowed. Archive devices must be distinct. The number of **FROM** clauses determines the amount of parallelism SAP Sybase IQ attempts with regard to input devices.

The backup/restore API DLL implementation lets you specify arguments to pass to the DLL when opening an archive device. For third-party implementations, the *archive_device* string has this format:

```
'DLLidentifier::vendor_specific_information'
```

A specific example is:

```
'spsc::workorder=12;volname=ASD002'
```

The *archive_device* string length can be up to 1023 bytes. The *DLLidentifier* portion must be 1 to 30 bytes in length and can contain only alphanumeric and underscore characters. The *vendor_specific_information* portion of the string is passed to the third-party implementation without checking its contents.

Note: Only certain third-party products are certified with SAP Sybase IQ using this syntax. See the *Release Bulletin* for additional usage instructions or restrictions. Before using any third-party product to back up your SAP Sybase IQ database, make sure it is certified. See the *Release Bulletin*, or see the Sybase Certification Reports for the SAP Sybase IQ product in *Technical Documents*.

For the Sybase implementation of the backup/restore API, you need not specify information other than the tape device name or file name. However, if you use disk devices, you must specify the same number of archive devices on the **RESTORE** as given on the backup; otherwise, you may have a different number of restoration devices than the number used to perform the backup. A specific example of an archive device for the Sybase API DLL that specifies a nonrewinding tape device for a UNIX system is:

```
'/dev/rmt/0n'
```

CATALOG ONLY—Restores only the backup header record from the archive media.

RENAME—Restore one or more SAP Sybase IQ database files to a new location. Specify each *dbspace-name* you are moving as it appears in the `SYSFILE` table. Specify *new-dbspace-path* as the new raw partition, or the new full or relative path name, for that dbspace.

If relative paths were used to create the database files, the files are restored by default relative to the catalog store file (the `SYSTEM` dbspace), and a rename clause is not required. If absolute paths were used to create the database files and a rename clause is not specified for a file, it is restored to its original location.

Relative path names in the **RENAME** clause work as they do when you create a database or dbspace: the main IQ store dbspace, temporary store dbspaces, and Message Log are restored relative to the location of `db_file` (the catalog store); user-created IQ store dbspaces are restored relative to the directory that holds the main IQ dbspace.

Do not use the **RENAME** clause to move the `SYSTEM` dbspace, which holds the catalog store. To move the catalog store, and any files created relative to it and not specified in a **RENAME** clause, specify a new location in the *db_file* parameter.

VERIFY [COMPATIBLE]— Directs the server to validate the specified SAP Sybase IQ database backup archives for a full, incremental, incremental since full, or virtual backup. The backup must be SAP Sybase IQ version 12.6 or later. The verification process checks the specified archives for the same errors a restore process checks, but performs no write operations. All status messages and detected errors are written to the server log file.

You cannot use the **RENAME** clause with the **VERIFY** clause; an error is reported.

The backup verification process can run on a different host than the database host. You must have the BACKUP DATABASE system privilege to run **RESTORE VERIFY**.

If the **COMPATIBLE** clause is specified with **VERIFY**, the compatibility of an incremental archive is checked with the existing database files. If the database files do not exist on the system on which **RESTORE…VERIFY COMPATIBLE** is invoked, an error is returned. If **COMPATIBLE** is specified while verifying a full backup, the keyword is ignored; no compatibility checks need to be made while restoring a full backup.

You must have the database and log files (`.db` and `.log`) to validate the backup of a read-only dbspace within a full backup. If you do not have these files, validate the entire backup by running **RESTORE…VERIFY** without the **READONLY** *dbspace* clause.

Note: The verification of a backup archive is different than the database consistency checker (DBCC) verify mode (`sp_iqcheckdb 'verify...'`). **RESTORE VERIFY** validates the consistency of the backup archive to be sure it can be restored, whereas DBCC validates the consistency of the database data.

Run `sp_iqcheckdb 'verify...'` before taking a backup. If an inconsistent database is backed up, then restored from the same backup archive, the data continues to be in an inconsistent state, even if **RESTORE VERIFY** reports a successful validation.

Other **RESTORE** issues:

- **RESTORE** to disk does not support raw devices as archival devices.
- SAP Sybase IQ does not rewind tapes before using them; on rewinding tape devices, it does rewind tapes after using them. You must position each tape to the start of the SAP Sybase IQ data before starting the **RESTORE**.
- During **BACKUP** and **RESTORE** operations, if SAP Sybase IQ cannot open the archive device (for example, when it needs the media loaded) and the ATTENDED option is ON, it waits for ten seconds for you to put the next tape in the drive, and then tries again. It continues these attempts indefinitely until either it is successful or the operation is terminated with Ctrl+C.
- If you press Ctrl+C, **RESTORE** fails and returns the database to its state before the restoration began.

- If disk striping is used, the striped disks are treated as a single device.
- The `file_name` column in the `SYSFILE` system table for the `SYSTEM` dbspace is not updated during a restore. For the `SYSTEM` dbspace, the `file_name` column always reflects the name when the database was created. The file name of the `SYSTEM` dbspace is the name of the database file.

Standards

- SQL—Vendor extension to ISO/ANSI SQL grammar.
- Sybase—Not supported by Adaptive Server Enterprise.

Permissions

The permissions required to execute this statement are set using the **-gu** server command line option, as follows:

- **NONE –** No user can issue this statement.
- **DBA –** Requires the SERVER OPERATOR system privilege.
- **UTILITY_DB –** Only those users who can connect to the `utility_db` database can issue this statement.

See also

- *Before You Restore* on page 72
- *Restoring IQ Store Data When Restoring the Database in the Same Location* on page 75
- *Restoring IQ Store Data When Restoring Database in a Different Location* on page 73
- *Synchronizing Servers* on page 24

Database Options

On a multiplex, database options control connections, distributed query processing and multiplex internode communications.

DQP_ENABLED Option

Temporary database option **DQP_ENABLED** allows you to enable or disable distributed query processing at the connection level.

Allowed Values
ON, OFF

Default
ON

Scope
Option can be set at the database (PUBLIC) or user level. When set at the database level, the value becomes the default for any new user, but has no impact on existing users. When set at the user level, overrides the PUBLIC value for that user only. No system privilege is required to set option for self. System privilege is required to set at database level or at user level for any user other than self.

Requires the SET ANY PUBLIC OPTION system privilege to set this option. Can be set temporary for an individual connection or for the PUBLIC role. Takes effect immediately.

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 80

DQP_ENABLED_OVER_NETWORK Option

Temporary database option **DQP_ENABLED_OVER_NETWORK** allows you to enable or disable distributed query processing over the network at the connection level.

Allowed Values
ON, OFF

Default
OFF

Scope
Requires the SET ANY PUBLIC OPTION system privilege to set this option for PUBLIC or for other user or role.

Can be set temporary for an individual or public.

Description
You can set the temporary database option **DQP_ENABLED_OVER_NETWORK** to ON to enable DQP over the network for the current connection. The OFF (default) setting has no effect, and the setting of the DQP_ENABLED logical server policy option determines whether or not DQP is used over the network for queries on the current connection.

LS Policy Option Setting	Database Option Setting	DQP Query Behavior
DQP_ENABLED 1	DQP_ENABLED_OVER_NETWORK ON	Queries on the current connection execute over network. Other queries use the shared temporary store.
DQP_ENABLED 2	DQP_ENABLED_OVER_NETWORK ON	All queries execute over network
DQP_ENABLED 2	DQP_ENABLED OFF	All queries run in simplex mode.

Note: Any changes you make to a logical server policy option affect new connections only. Logical server policy options for existing connections are based on the time that the connection was initially established.

MPX_AUTOEXCLUDE_TIMEOUT Option

Timeout for autoexcluding a secondary node on the coordinator node. This option does not apply to the designated failover node.

Allowed Values
0 to 10080 minutes (1 week). 0 indicates that the nodes are not autoexcluded. 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
Option can be set at the database (PUBLIC) level only.

Requires the SET ANY SYSTEM OPTION system privilege to set this option. Setting takes effect immediately and persists across server restarts.

MPX_HEARTBEAT_FREQUENCY Option

Interval until the heartbeat thread wakes and performs periodic operations, such as checking for coordinator connectivity and cleaning up the connection pool on the secondary node. The

heartbeat thread maintains a dedicated internal connection from secondary server to coordinator.

Allowed Values
2 seconds to 3600 seconds

Default
60 seconds

Scope
Option can be set at the database (PUBLIC) level only.

Requires the SET ANY SYSTEM OPTION system privilege to set this option. You must restart the server for the change to take effect.

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
Option can be set at the database (PUBLIC) level only.

Requires the SET ANY SYSTEM OPTION system privilege to set this option. Setting takes effect immediately and persists across server restarts.

MPX_LIVENESS_TIMEOUT Option

Time, in seconds, before a heartbeat on a secondary server declares the coordinator offline if the heartbeat fails to reconnect to the coordinator after the first disconnect. This option also determines how long the coordinator keeps a global transaction in a suspended state.

Allowed Values
0 to 604800 (1 week) in seconds

Default
3600 seconds (1 hour)

Scope
This option affects all multiplex nodes and has no node-specific or connection-specific value. Option can be set at the database (PUBLIC) level only.

Requires the SET ANY SYSTEM OPTION system privilege to set this option. If you change the value of `MPX_LIVENESS_TIMEOUT` on a running server, the new value takes effect immediately for connections that might suspend in the future. The changed value also immediately affects the remaining timeout period for all current suspended transactions.

Description
If a writer fails to resume a suspended transaction within the `MPX_LIVENESS_TIMEOUT` period, the transaction can no longer commit, and the user should roll back the transaction. The coordinator keeps a global transaction in a suspended state for a period of `2 * MPX_LIVENESS_TIMEOUT`. If the corresponding writer fails to resume the transaction before the `2 * MPX_LIVENESS_TIMEOUT` period, the coordinator rolls back the suspended transaction.

Always specify an `MPX_LIVENESS_TIMEOUT` value that is a multiple of the current `MPX_HEARTBEAT_FREQUENCY` value, which controls the aliveness check period. The coordinator internally doubles the value of `MPX_LIVENESS_TIMEOUT`.

See also
- *Coordinator Failure* on page 26
- *Global Transaction Resiliency* on page 42
- *Dropping Multiplex Servers* on page 21

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
Option can be set at the database (PUBLIC) level only.

Requires the SET ANY SYSTEM OPTION system privilege to set this 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 SAP Sybase IQ server response times.

See also

- *Pooled Connections* on page 8

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
Option can be set at the database (PUBLIC) level only.

Requires the SET ANY SYSTEM OPTION system privilege to set this option. Setting takes effect immediately and persists across server restarts.

See also

- *Pooled Connections* on page 8

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
Option can be set at the database (PUBLIC) or user level. When set at the database level, the value becomes the default for any new user, but has no impact on existing users. When set at the user level, overrides the PUBLIC value for that user only. No system privilege is required to set option for self. System privilege is required to set at database level or at user level for any user other than self.

Requires the SET ANY PUBLIC OPTION system privilege to set this option. Can be set temporary for an individual connection or for the PUBLIC role. Takes effect immediately.

System Tables

Certain system tables support multiplex functionality.

ISYSIQINFO System Table

This table indicates the database characteristics as defined when the SAP 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.

Column name	Column type	Description
last_incr_backup	TIMESTAMP	The completion time of the most recent incremental backup.
create_time	TIMESTAMP NOT NULL	The date and time when the database was 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.
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 125

ISYSIQLOGINPOLICYLSINFO System Table

ISYSIQLOGINPOLICYLSINFO stores the login policy logical server assignment information.

See also

- *SYSIQLOGINPOLICYLSINFO System View* on page 126

ISYSIQLSLOGINPOLICYOPTION System Table

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

See also

- *SYSIQLSPOLICYOPTION System View* on page 129

ISYSIQLSMEMBER System Table

ISYSIQLSMEMBER stores the logical server membership information.

See also

- *SYSIQLSMEMBER System View* on page 127

ISYSIQLSPOLICY System Table

ISYSIQLSPOLICY stores logical server policies.

See also

- *SYSIQLSPOLICY System View* on page 128

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 129

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 124

SYSIQLOGINPOLICYLSINFO System View

Presents a readable version of the table ISYSIQLOGINPOLICYLSINFO.

The ISYSIQLOGINPOLICYLSINFO 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 124

SYSIQLSLOGINPOLICIES Consolidated View

Describes all the logical server assignments from the login policies.

This consolidated system view 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)

SYSIQLSMEMBER System View

Presents group information from the ISYSIQLSMEMBER table, which stores logical server membership information.

`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.
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 124

SYSIQLSMEMBERS Consolidated View

Describes all user-defined logical server memberships.

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.

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 124

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 124

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 125

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. **sp_iqcheckdb** does not check a partitioned table if partitioned data exists on offline dbspaces.

sp_iqcheckdb reads all storage in the database. On successful completion, the database free list (an internal allocation map) is updated to reflect the true storage allocation for the database. **sp_iqcheckdb** then generates a report listing the actions it has performed.

If an error is found, **sp_iqcheckdb** reports the name of the object and the type of error. **sp_iqcheckdb** does not update the free list if errors are detected.

sp_iqcheckdb also allows you to check the consistency of a specified table, index, index type, or the entire database.

Note: sp_iqcheckdb is the user interface to the SAP Sybase IQ database consistency checker (DBCC) and is sometimes referred to as **DBCC**.

Syntax

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

This is the general syntax of **sp_iqcheckdb**. There are three modes for checking database consistency, and one for resetting allocation maps. The syntax for each mode is listed separately below. If mode and target are not both specified in the parameter string, SAP Sybase IQ returns the error message:

```
At least one mode and target must be specified to DBCC.
```

Parameters

mode: { **allocation** | **check** | **verify** } | **dropleaks**

target: [**indextype** *index-type* […]] **database** | **database resetclocks** | { [**indextype** *index-type*] […] **table** *table-name* [**partition** *partition-name*] […] | **index** *index-name* | […] **dbspace** *dbspace-name*}

Applies to

Simplex and multiplex.

Allocation Mode

sp_iqcheckdb 'allocation *target* [resources *resource-percent*]'

Check Mode

sp_iqcheckdb 'check *target* [resources *resource-percent*]'

Verify Mode

sp_iqcheckdb 'verify *target* [resources *resource-percent*]'

Dropleaks Mode

sp_iqcheckdb 'dropleaks *target* [resources *resource-percent*]'

Usage

Parameter	Description
database	If the target is a database, all dbspaces must be online.

Parameter	Description
index-type	One of the following index types: **FP**, **CMP**, **LF**, **HG**, **HNG**, **WD**, **DATE**, **TIME**, **DTTM**, **TEXT**. If the specified *index-type* does not exist in the target, an error message is returned. If multiple index types are specified and the target contains only some of these index types, the existing index types are processed by **sp_iqcheckdb**.
index-name	May contain owner and table qualifiers: `[[owner.]table-name.]index-name` If *owner* is not specified, current user and database owner (`dbo`) are substituted in that order. If *table* is not specified, *index-name* must be unique.
table-name	May contain an owner qualifier: `[owner.]table-name` If *owner* is not specified, current user and database owner (`dbo`) are substituted in that order. *table-name* cannot be a temporary or pre-join table. **Note:** If either the table name or the index name contains spaces, enclose the *table-name* or *index-name* parameter in double quotation marks: `sp_iqcheckdb 'check index "dbo.sstab.i2" resources 75'`
partition-name	The *partition-name* parameter contains no qualifiers. If it contains spaces, enclose it in double quotation marks. The partition filter causes **sp_iqcheckdb** to examine a subset of the corresponding table's rows that belong to that partition. A partition filter on a table and table target without the partition filter are semantically equivalent when the table has only one partition.

Parameter	Description
dbspace-name	The *dbspace-name* parameter contains no qualifiers. If it contains spaces, enclose it in double quotation marks. The dbspace target examines a subset of the database's pages that belong to that dbspace. The dbspace must be online. The dbspace and database target are semantically equivalent when the table has only one dbspace.
resource-percent	The input parameter *resource-percent* must be an integer greater than zero. The resources percentage allows you to limit the CPU utilization of the database consistency checker by controlling the number of threads with respect to the number of CPUs. If *resource-percent* = 100 (the default value), then one thread is created per CPU. If *resource-percent* > 100, then there are more threads than CPUs, which might increase performance for some machine configurations. The minimum number of threads is one.

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

Allocation problems can be repaired in dropleaks mode.

Privileges

Requires the ALTER DATABASE system privilege. Users without the ALTER DATABASE system privilege must be granted `EXECUTE` permission to run the stored procedure.

Description

sp_iqcheckdb checks the allocation of every block in the database and saves the information in the current session until the next **sp_iqdbstatistics** procedure is issued. **sp_iqdbstatistics** displays the latest result from the most recent execution of **sp_iqcheckdb**.

sp_iqcheckdb can perform several different functions, depending on the parameters specified.

Allocation Mode

Checks allocation with blockmap information for the entire database, a specific index, a specific index type, a specific partition, specific table, or a specific dbspace. Does not check index consistency.

Detects duplicate blocks (blocks for which two or more objects claim ownership) or extra blocks (unallocated blocks owned by an object).

Detects leaked blocks (allocated blocks unclaimed by any object in the specified target) for database or dbspace targets.

When the target is a partitioned table, **allocation mode**:

- Checks metadata of all the table's partition allocation bitmaps
- Checks metadata of the tables allocation bitmap
- Verifies that blockmap entries are consistent with the table's allocation bitmap
- Verifies that none of the table's partition allocation bitmaps overlap
- Checks that rows defined in the table's partition allocation bitmaps form a superset of the table's existence bitmap
- Checks that rows defined in the table's partition allocation bitmaps form a superset of the table's allocation bitmap

Note: sp_iqcheckdb cannot check all allocation problems if you specify the name of a single index, index type, or table in the input parameter string.

Run in allocation mode:

- To detect duplicate or unowned blocks (use database or specific tables or indexes as the target)
- If you encounter page header errors

The DBCC option **resetclocks** is used only with allocation mode. **resetclocks** is used with forced recovery to convert a multiplex secondary server to a coordinator. For information on multiplex capability, see *Using SAP Sybase IQ Multiplex*. **resetclocks** corrects the values of internal database versioning clocks, in the event that these clocks are behind. Do not use the **resetclocks** option for any other purpose, unless you contact SAP Sybase IQ Technical Support.

The **resetclocks** option must be run in single-user mode and is allowed only with the DBCC statement **allocation database**. The syntax of **resetclocks** is:

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

Check Mode

Verifies that all database pages can be read for the entire database, specific index, specific index type, specific table, specific partition, or specific dbspace. If the table is partitioned, then check mode will check the table's partition allocation bitmaps.

Run in check mode if metadata, null count, or distinct count errors are returned when running a query.

Verify Mode

Verifies the contents of non-FP indexes with their corresponding FP indexes for the entire database, a specific index, a specific index type, specific table, specific partition, or specific

dbspace. If the specified target contains all data pages for the FP and corresponding non-FP indexes, then verify mode detects the following inconsistencies:

- Missing key – a key that exists in the FP but not in the non-FP index.
- Extra key – a key that exists in the non-FP index but not in the FP index.
- Missing row – a row that exists in the FP but not in the non-FP index.
- Extra row – a row that exists in the non-FP index but not in the FP index.

If the specified target contains only a subset of the FP pages, then verify mode can detect only the following inconsistencies:

- Missing key
- Missing row

If the target is a partitioned table, then verify mode also verifies that each row in the table or table partition has been assigned to the correct partition.

Run in verify mode if metadata, null count, or distinct count errors are returned when running a query.

Note: sp_iqcheckdb does not check referential integrity or repair referential integrity violations.

Dropleaks Mode

When the SAP Sybase IQ server runs in single-node mode, you can use dropleaks mode with either a database or dbspace target to reset the allocation map for the entire database or specified dbspace targets. If the target is a dbspace, then the dropleaks operation must also prevent read-write operations on the named dbspace. All dbspaces in the database or dbspace list must be online.

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

The following examples illustrate the use of the **sp_iqcheckdb** procedure.

Example 1

Check the allocation for the entire database:

```
sp_iqcheckdb 'allocation database'
```

Example 2

Perform a detailed check on indexes `i1`, `i2`, and `dbo.t1.i3`. If you do not specify a new mode, **sp_iqcheckdb** applies the same mode to the remaining targets, as shown in the following command:

```
sp_iqcheckdb 'verify index i1 index i2 index dbo.t1.i3'
```

Example 3

You can combine all modes and run multiple checks on a database in a single session. Perform a quick check of partition `p1` in table `t2`, a detailed check of index `i1`, and allocation checking for the entire database using half of the CPUs:

```
sp_iqcheckdb 'check table t2 partition p1 verify index i1
allocation database resources 50'
```

Example 4

Check all indexes of the type **FP** in the database:

```
sp_iqcheckdb 'check indextype FP database'
```

Example 5

Verify the **FP** and **HG** indexes in the table `t1` and the **LF** indexes in the table `t2`:

```
sp_iqcheckdb 'verify indextype FP indextype HG table t1 indextype LF
table t2'
```

Example 6

Check for LVC cell inconsistencies:

```
sp_iqcheckdb 'check index EFG2JKL.ASIQ_IDX_T208_C504_FP'
-------------------------------------
Index Statistics:
** Inconsistent Index: abcd.EFG2JKL.ASIQ_IDX_T208_C504_FP ****** FP
Indexes Checked: 1
** Unowned LVC Cells: 212 ******
```

The **sp_iqcheckdb** LVC cells messages include:

- Unowned LVC cells
- Duplicate LVC cell rows
- Unallocated LVC cell rows

These messages indicate inconsistencies with a `VARCHAR`, `VARBINARY`, `LONG BINARY` (`BLOB`), or `LONG VARCHAR` (`CLOB`) column. Unowned LVC cells represent a small amount of unusable disk space and can safely be ignored. Duplicate and Unallocated LVC cells are serious errors that can be resolved only by dropping the damaged columns.

To drop a damaged column, create a new column from a copy of the old column, then drop the original column and rename the new column to the old column.

Note: LVC is a `VARCHAR` or `VARBINARY` column with a width greater than 255. `LONG BINARY` (`BLOB`) and `LONG VARCHAR` (`CLOB`) also use LVC.

DBCC performance

The execution time of DBCC varies, depending on the size of the database for an entire database check, the number of tables or indexes specified, and the size of the machine.

Checking only a subset of the database (that is, only specified tables, indexes, or index types) requires less time than checking an entire database.

The processing time of **sp_iqcheckdb** dropleaks mode depends on the number of dbspace targets.

This table summarizes the actions and output of the four **sp_iqcheckdb** modes.

Table 13. Actions and Output of sp_iqcheckdb Modes

Mode	Errors Detected	Output	Speed
Allocation	Allocation errors	Allocation statistics only	4TB per hour
Check	Allocation errors Most index errors	All available statistics	60GB per hour
Verify	Allocation errors All index errors	All available statistics	15GB per hour
Dropleaks	Allocation errors	Allocation statistics only	4TB per hour

Output

Depending on the execution mode, **sp_iqcheckdb** output includes summary results, errors, informational statistics, and repair statistics. The output may contain as many as three results sets, if you specify multiple modes in a single session. Error statistics are indicated by asterisks (*****), and appear only if errors are detected.

The output of **sp_iqcheckdb** is also copied to the SAP Sybase IQ message file `.iqmsg`. If the **DBCC_LOG_PROGRESS** option is ON, **sp_iqcheckdb** sends progress messages to the IQ message file, allowing the user to follow the progress of the DBCC operation as it executes.

Output Example

Run **sp_iqcheckdb 'allocation database'** :

```
========================================================================
DBCC Allocation Mode Report
========================================================================
   DBCC Status                                      No Errors Detected
========================================================================
Allocation Summary
========================================================================

   Blocks Total                                      25600
   Blocks in Current Version                         5917
   Blocks in All Versions                            5917
   Blocks in Use                                     5917
   % Blocks in Use                                   23
========================================================================
Allocation Statistics
========================================================================
```

```
   Marked Logical Blocks                          8320
   Marked Physical Blocks                         5917
   Marked Pages                                   520
   Blocks in Freelist                             2071196
   Imaginary Blocks                               2014079
   Highest PBN in Use                             1049285
   Total Free Blocks                              19683
   Usable Free Blocks                             19382
   % Total Space Fragmented                       1
   % Free Space Fragmented                        1
   Max Blocks Per Page                            16
   1  Block Page Count                            165
   3  Block Page Count                            200
   4  Block Page Count                            1
   10 Block Page Count                            1
   16 Block Page Count                            153
   2  Block Hole Count                            1
   3  Block Hole Count                            19
   6  Block Hole Count                            12
   7  Block Hole Count                            1
   10 Block Hole Count                            1
   15 Block Hole Count                            1
   16 Block Hole Count                            1220

Partition Summary
   Database Objects Checked                       2
   Blockmap Identity Count                        2
   Bitmap Count                                   2
========================================================================
Connection Statistics
========================================================================
   Sort Records                                   3260
   Sort Sets                                      2
========================================================================

DBCC Info
========================================================================
   DBCC Work units Dispatched                     197
   DBCC Work units Completed                      197
   DBCC Buffer Quota                              255
   DBCC Per-Thread Buffer Quota                   255
   Max Blockmap ID found                          200
   Max Transaction ID found                       404
```

Note: The report may indicate leaked space. Leaked space is a block that is allocated according to the database free list (an internal allocation map), but DBCC finds that the block is not part of any database object.

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 SAP Sybase IQ, connection status, database version status, and so on.

Syntax

sp_iqconnection [*connhandle*]

Applies to

Simplex and multiplex.

Usage

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.

Privileges

Requires the DROP CONNECTION, MONITOR or SERVER OPERATOR system privilege. Users without one of these system privileges 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 SAP Sybase IQ side of the SAP Sybase IQ engine. Rows are ordered by ConnCreateTime.

The column MPXServerName stores information related to internode communication (INC), as shown:

Server Where Run	MPXServerName Column Content
Simplex server	NULL (All connections are local/user connections)

Server Where Run	MPXServerName Column Content
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 SAP 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");
```

Column Name	Description
ConnHandle	The ID number of the connection.
Name	The name of the server.
Userid	The user ID for the connection.
LastReqTime	The time at which the last request for the specified connection started.
ReqType	A string for the type of the last request.
IQCmdType	The current command executing on the SAP Sybase IQ side, if any. The command type reflects commands defined at the implementation level of the engine. These commands consist 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 SAP Sybase IQ engine on this connection.
IQCursors	The number of cursors open in the IQ store on this connection.

Column Name	Description
LowestIQCursorState	The IQ cursor state, if any. If multiple cursors exist on the connection, the state that appears is the lowest cursor state of all the cursors; that is, the furthest from completion. Cursor state reflects internal SAP 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 SAP 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 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 SAP Sybase IQ temporary tables are reflected in TempTableSpaceKB.
IQConnID	The ten-digit connection ID included 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 SAP Sybase IQ engine. This might be occasionally useful in determining connection activity. Result sets are returned in buffers of rows and do not increment satoiq_count or iqtosa_count once per row.
iqtosa_count	An internal counter used to display the number of crossings from the IQ side to the SQL Anywhere side of the SAP 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 SAP 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.

Column Name	Description
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.
INCConnName	The name of the underlying INC connection for a user connection. The data type for this column is varchar(255). If **sp_iqconnection** shows an INC connection name for a suspended user connection, that user connection has an associated INC connection that is also suspended.
INCConnSuspended	The value "Y" in this column indicates that the underlying INC connection for a user connection is in a suspended state. The value "N" indicates that the connection is not suspended.

Example

sp_iqconnection

```
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
======== ============ ============ ======== ======= ======== ============
      34           43            2  'local'      ''      244  (NULL)

LSName      INCConnName                 INCConnSuspended
==========  ==================          ================
Finance_LS  'IQ_MPX_SERVER_P54'         'Y'
```

sp_iqdbsize Procedure

Displays the size of the current database.

Syntax

```
sp_iqdbsize ( [ main ] )
```

Applies to

Simplex and multiplex.

Privileges
Requires the ALTER DATABASE system privilege. Users without the ALTER DATABASE system privilege must be granted `EXECUTE` permission to run the stored procedure.

Description
Returns the total size of the database. Also returns the number of pages required to hold the database in memory and the number of IQ pages when the database is compressed (on disk).

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

If run when there are no rows in any RLV-enabled tables, the Physical Blocks, the RLVLogBlocks and RLVLogKBytes columns will contain non-zero entries, and the remaining columns contain zeros. This indicate no row-level versioned tables.

Column Name	Description
Database	The path name of the database file.
Physical Blocks	Total database size in blocks. An IQ database consists of one or more dbspaces. Each dbspace has a fixed size, which is originally specified in units of megabytes. This megabyte quantity is converted to blocks using the IQ page size and the corresponding block size for that IQ page size. The Physical Blocks column reflects the cumulative total of each SAP Sybase IQ dbspace size, represented in blocks.
KBytes	The total size of the database in kilobytes. This value is the total size of the database in blocks (Physical Blocks in the previous **sp_iqdbsize** column) multiplied by the block size. The block size depends on the IQ page size.
Pages	The total number of IQ pages necessary to represent in memory all of the data stored in tables and the metadata for these objects. This value is always greater than or equal to the value of Compressed Pages (the next **sp_iqdbsize** column).
Compressed Pages	The total number of IQ pages necessary to store on disk the data in tables and metadata for these objects. This value is always less than or equal to the value of Pages (the previous **sp_iqdbsize** column), because SAP Sybase IQ compresses pages when the IQ page is written from memory to disk. The **sp_iqdbsize** Compressed Pages column represents the number of compressed pages.
NBlocks	The total size in blocks used to store the data in tables. This value is always less than or equal to the **sp_iqdbsize** Physical Blocks value.

Column Name	Description
Catalog Blocks	The total size in blocks used to store the metadata for tables.
RLVLogBlocks	Number of blocks used for log information for the RLV store.
RLVLogKBytes	Total size of the RLV log, in Kb.

Example
Displays size information for the database `iqdemo`:

```
sp_iqdbsize
```

```
Database

PhysicalBlocks KBytes Pages CompressedPages NBlocks CatalogBlocks
RLVLogBlocks RLVLogKBytes
============== ====== ===== =============== ======= =============
============ ============
/system1/sybase/IQ-16_0/demo/iqdemo.db
          1280    522   688             257    1119              18
```

sp_iqdbspace Procedure

Displays detailed information about each IQ dbspace.

Syntax

sp_iqdbspace [*dbspace-name*]

Applies to
Simplex and multiplex.

Privileges
Requires MANAGE ANY DBSPACE system privilege. Users without MANAGE ANY DBSPACE system privilege must be granted `EXECUTE` permission.

Description
Use the information from **sp_iqdbspace** to determine whether data must be moved, and for data that has been moved, whether the old versions have been deallocated.

Column Name	Description
DBSpaceName	Name of the dbspace as specified in the **CREATE DBSPACE** statement. Dbspace names are always case-insensitive, regardless of the **CREATE DATABASE**...**CASE IGNORE** or **CASE RESPECT** specification.
DBSpaceType	Type of the dbspace (`MAIN`, `SHARED_TEMP`, `TEMPORARY`, or `RLV`).
Writable	T (writable) or F (not writable).

Column Name	Description
Online	T (online) or F (offline).
Usage	Percent of dbspace currently in use by all files in the dbspace.
TotalSize	Total size of all files in the dbspace in the units B (bytes), K (kilobytes), M (megabytes), G (gigabytes), T (terabytes), or P (petabytes).
Reserve	Total reserved space that can be added to all files in the dbspace.
NumFiles	Number of files in the dbspace.
NumRWFiles	Number of read/write files in the dbspace.
Stripingon	F (Off).
StripeSize	Always 1, if disk striping is on.
BlkTypes	Space used by both user data and internal system structures.
OkToDrop	"Y" indicates the dbspace can be dropped; otherwise "N".

Values of the BlkTypes block type identifiers:

Identifier	Block Type
A	Active version
B	Backup structures
C	Checkpoint log
D	Database identity
F	Free list
G	Global free list manager
H	Header blocks of the free list
I	Index advice storage
M	Multiplex CM*
O	Old version
R	RLV free list manager
T	Table use
U	Index use
N	Column use

Identifier	Block Type
X	Drop at checkpoint

*The multiplex commit identity block (actually 128 blocks) exists in all IQ databases, even though it is not used by simplex databases.

Example

Displays information about dbspaces:

```
sp_iqdbspace;
```

Note: The following example shows objects in the `iqdemo` database to better illustrate output. `iqdemo` includes a sample user dbspace named `iq_main` that may not be present in your own databases.

DBSpaceName	DBSpaceType	Writable	Online	Usage	Total Size	Reserve	Num Files	Num RWFiles	Stripingon	Stripe Size	Blk Types	Ok To Drop
IQ_MAIN	MAIN	T	T	55	75M	200 M	1	1	T	1K	1H, 5169A, 190	N
IQ__ SYSTEM_ MAIN	MAIN	T	T	21	300 M	50M	1	1	F	8K	1H, 7648F, 32D, 128 M	N
IQ_SYSTEM_ TEMP	TEMPORARY	T	T	1	100 M	50M	1	1	F	8K	1H, 64F, 32A	N

See also

- *Before You Restore* on page 72
- *Restoring IQ Store Data When Restoring the Database in the Same Location* on page 75
- *Restoring IQ Store Data When Restoring Database in a Different Location* on page 73
- *Synchronizing Servers* on page 24

sp_iqdbspaceinfo Procedure

Displays the size of each object and subobject used in the specified table. Not supported for RLV dbspaces.

Syntax

```
sp_iqdbspaceinfo [ dbspace-name ] [ , owner_name ] [ ,
object_name ] [ , object-type  ]
```

Applies to

Simplex and multiplex.

Privileges

Requires the BACKUP DATABASE, SERVER OPERATOR, or MANAGE ANY DBSPACE system privileges. Users without one of these system privileges must be granted `EXECUTE` permission.

Usage

Parameter	Description
dbspace_name	If specified, **sp_iqdbspaceinfo** displays one line for each table that has any component in the specified dbspace. Otherwise, the procedure shows information for all dbspaces in the database.
owner_name	Owner of the object. If specified, **sp_iqdbspaceinfo** displays output only for tables with the specified owner. If not specified, **sp_iqdbspaceinfo** displays information on tables for all users in the database.
object_name	Name of the table. If not specified, **sp_iqdbspaceinfo** displays information on all tables in the database.
object_type	Valid **table** objects.

All parameters are optional, and any parameter may be supplied independent of another parameter's value.

The **sp_iqdbspaceinfo** stored procedure supports wildcard characters for interpreting *dbspace_name*, *object_name*, and *owner_name*. It shows information for all dbspaces that match the given pattern in the same way the **LIKE** clause matches patterns inside queries.

Description

The procedure returns no results if you specify an RLV dbspace.

sp_iqdbspaceinfo shows the DBA the amount of space used by objects that reside on each dbspace. The DBA can use this information to determine which objects must be relocated before a dbspace can be dropped. The subobject columns display sizes reported in integer quantities followed by the suffix B, K, M, G, T, or P, representing bytes, kilobytes, megabytes, gigabytes, terabytes, and petabytes, respectively.

For tables, **sp_iqdbspaceinfo** displays subobject sizing information for all subobjects (using integer quantities with the suffix B, K, M, G, T, or P) sorted by *dbspace_name*, *object_name*, and *owner_name*.

Table 14. sp_iqdbspaceinfo Columns

Column Name	Description
dbspace_name	Name of the dbspace.
object_type	Type of the object (**table** or **joinindex** only).
owner	Name of the owner of the object.
object_name	Name of the object on the dbspace.
object_id	Global object ID of the object.
id	Table id of the object.
columns	Size of column storage space on the given dbspace.
indexes	Size of index storage space on the given dbspace. Does not use system-generated indexes (for example, HG indexes in unique constraints or FP indexes).
metadata	Size of storage space for metadata objects on the given dbspace.
primary_key	Size of storage space for primary key related objects on the given dbspace.
unique_constraint	Size of storage space for unique constraint-related objects on the given dbspace.
foreign_key	Size of storage space for foreign-key-related objects on the given dbspace.
dbspace_online	Indicates if the dbspace is online (**Y**) or offline (**N**).

If you run sp_iqdbspaceinfo against a server you have started with the -r switch (read-only), you see the error `Msg 13768, Level 14, State 0: SQL Anywhere Error -757: Modifications not permitted for read-only database`. This behavior is expected. The error does not occur on other stored procedures such as

`sp_iqdbspace`, `sp_iqfile`, `sp_iqdbspaceobjectinfo` or `sp_iqobjectinfo`.

Examples

Note: These examples show objects in the `iqdemo` database to better illustrate output. `iqdemo` includes a sample user dbspace named `iq_main` that may not be present in your own databases.

Displays the size of all objects and subobjects in all tables in all dbspaces in the database:

```
sp_iqdbspaceinfo
```

```
dbspace_name    object_type  owner   object_name     object_id  id
  columns
iq_main        table       DBA     emp1            3689     741   96K
iq_main        table       DBA     iq_dummy        3686     740   24K
iq_main        table       DBA     sale            3698     742   96K
iq_main         table        GROUPO  Contacts          3538     732
   288K
iq_main         table        GROUPO  Customers         3515     731
   240K
iq_main        table       GROUPO  Departments     3632     738   72K
iq_main         table        GROUPO  Employees         3641     739
   408K
iq_main         table        GROUPO  FinancialCodes    3612     736
   72K
iq_main        table       GROUPO  FinancialData    3621     737   96K
iq_main         table        GROUPO  Products
 3593     735   272K
iq_main         table        GROUPO  SalesOrderItems   3580     734
   120K
iq_main         table        GROUPO  SalesOrders       3565     733
   144K

indexes  metadata  primary_key  unique_constraint  foreign_key  dbsp
ace_online
0B       1.37M     0B           0B                 0B           Y
0B       464K      0B           0B                 0B           Y
0B       1.22M     0B           0B                 0B           Y
0B       5.45M     24K          0B                 48K          Y
48K      4.63M     24K          0B                 0B           Y
0B       1.78M     24K          0B                 48K          Y
0B       8.03M     24K          0B                 48K          Y
0B       1.53M     24K          0B                 0B           Y
0B       2.19M     24K          0B                 48K          Y
192K     4.67M     24K          0B                 0B           Y
0B       2.7M      24K          0B                 104K         Y
0B       3.35M     24K          0B                 144K         Y
```

Displays the size of all objects and subobjects owned by a specified user in a specified dbspace in the database:

```
sp_iqdbspaceinfo iq_main,GROUPO
```

```
dbspace_name    object_type  owner   object_name     object_id  id
  columns
iq_main         table        GROUPO  Contacts          3538     732
   288K
iq_main         table        GROUPO  Customers         3515     731
   240K
iq_main        table       GROUPO  Departments      3632    738   72K
iq_main         table        GROUPO  Employees         3641     739
   408K
iq_main         table        GROUPO  FinancialCodes    3612     736
   72K
iq_main        table       GROUPO  FinancialData    3621    737   96K
iq_main         table        GROUPO  Products          3593     735
   272K
iq_main         table        GROUPO  SalesOrderItems   3580     734
   120K
iq_main         table        GROUPO  SalesOrders       3565     733
   144K

indexes  metadata  primary_key  unique_constraint  foreign_key  dbsp
ace_online
0B       5.45M     24K          0B                 48K          Y
48K      4.63M     24K          0B                 0B           Y
0B       1.78M     24K          0B                 48K          Y
0B       8.03M     24K          0B                 48K          Y
0B       1.53M     24K          0B                 0B           Y
0B       2.19M     24K          0B                 48K          Y
192K     4.67M     24K          0B                 0B           Y
0B       2.7M      24K          0B                 104K         Y
0B       3.35M     24K          0B                 144K         Y
```

Displays the size of a specified object and its subobjects owned by a specified user in a specified dbspace in the database:

```
sp_iqdbspaceinfo iq_main,GROUPO,Departments
```

```
dbspace_name    object_type  owner   object_name     object_id  id
  columns
iq_main        table       GROUPO  Departments      3632    738   72K

indexes  metadata  primary_key  unique_constraint  foreign_key  dbsp
ace_online
0B       1.78M     24K          0B                 48K          Y
```

sp_iqfile Procedure

Displays detailed information about each dbfile in a dbspace.

Syntax

sp_iqfile [*dbspace-name*]

Applies to

Simplex and multiplex.

Privileges

Requires the MANAGE ANY DBSPACE system privilege. Users without the MANAGE ANY DBSPACE system privilege must be granted `EXECUTE` permission.

Description

sp_iqfile displays the usage, properties, and types of data in each dbfile in a dbspace. You can use this information to determine whether data must be moved, and for data that has been moved, whether the old versions have been deallocated.

Column Name	Description
DBSpaceName	Name of the dbspace as specified in the **CREATE DBSPACE** statement. Dbspace names are always case-insensitive, regardless of the **CREATE DATABASE**...**CASE IGNORE** or **CASE RESPECT** specification.
DBFileName	Logical file name.
Path	Location of the physical file or raw partition.
SegmentType	Type of dbspace (`MAIN`, `TEMPORARY`, or `RLV`).
RWMode	Mode of the dbspace: always read-write (RW).
Online	T (online) or F (offline).
Usage	Percent of dbspace currently in use by this file in the dbspace. When run against a secondary node in a multiplex configuration, this column displays NA.
DBFileSize	Current size of the file or raw partition. For a raw partition, this size value can be less than the physical size.
Reserve	Reserved space that can be added to this file in the dbspace.
StripeSize	Always 1, if disk striping is on.
BlkTypes	Space used by both user data and internal system structures.
FirstBlk	First IQ block number assigned to the file.
LastBlk	Last IQ block number assigned to the file.
OkToDrop	"Y" indicates the file can be dropped; otherwise "N".

Identifier	Block Type
A	Active Version
B	Backup Structures

Identifier	Block Type
C	Checkpoint Log
D	Database Identity
F	Free list
G	Global Free list Manager
H	Header Blocks of the Free List
I	Index Advice Storage
M	Multiplex CM*
O	Old Version
R	RLV Free list manager
T	Table Use
U	Index Use
N	Column Use
X	Drop at Checkpoint

*The multiplex commit identity block (actually 128 blocks) exists in all IQ databases, even though it is not used by simplex databases.

Example

Displays information about the files in the dbspaces:

```
sp_iqfile;
```

```
sp_iqfile;
DBSpaceName,DBFileName,Path,SegmentType,RWMode,Online,
Usage,DBFileSize,Reserve,StripeSize,BlkTypes,FirstBlk,
LastBlk,OkToDrop

'IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','/sun1-c1/users/smith/mpx/m/
mpx_db.iq','MAIN','RW','T','21','
2.92G','0B','1K','1H,76768F,32D,19A,185O,128M,34B,32C'
,1,384000,'N'

'mpx_main1','mpx_main1','/sun1-c1/users/smith/mpx/m/
mpx_main1.iq','MAIN','RW','T','1'
,'100M','0B','1K','1H',1045440,1058239,'N'

'IQ_SHARED_TEMP','sharedfile1_bcp','/sun1-c1/users/smith/mpx/m/
f1','SHARED_TEMP','RO','T','0',
'50M','0B','1K','1H',1,6400,'N'
```

```
'IQ_SHARED_TEMP','sharedfile2_bcp','/sun1-c1/users/smith/mpx/m/
f2','SHARED_TEMP','RO','T','0',
'50M','0B','1K','1H',1045440,1051839,'N'

'IQ_SYSTEM_TEMP','IQ_SYSTEM_TEMP','/sun1-c1/users/smithmpx/m/
mpx_db.iqtmp','TEMPORARY','RW',
'T','1','2.92G','0B','1K','1H,64F,33A',1,384000,'N'
```

sp_iqindexinfo Procedure

Displays the number of blocks used per index per main dbspace for a given object. If the object resides on several dbspaces, **sp_iqindexinfo** returns the space used in all dbspaces, as shown in the example.

Syntax

```
sp_iqindexinfo '{ database
| [ table table-name | index index-name ] [...] }
[ resources resource-percent ]'
```

Applies to

Simplex and multiplex.

Privileges

Requires MANAGE ANY DBSPACE system privilege. Users without MANAGE ANY DBSPACE system privilege must be granted `EXECUTE` permission to run the stored procedure.

Usage

You can request index information for the entire database, or you can specify any number of table or index parameters. If a table name is specified, **sp_iqindexinfo** returns information on all indexes in the table. If an index name is specified, only the information on that index is returned.

If the specified *table-name* or *index-name* is ambiguous or the object cannot be found, an error is returned.

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

resource-percent must be an integer greater than 0. The resources percentage allows you to limit the CPU utilization of the **sp_iqindexinfo** procedure by specifying the percent of total CPUs to use.

Description

sp_iqindexinfo shows the DBA on which dbspaces a given object resides. The DBA can use this information to determine which dbspaces must be given **relocate** mode to relocate the object.

The results of **sp_iqindexinfo** are from the point of view of the version seen by the transaction running the command. Blocks used by other versions are not shown.

Table 15. sp_iqindexinfo columns

Column name	Description
Object	Table or index name
Dbspace_name	Name of the dbspace
ObjSize	Size of data for this object on this dbspace
DBSpPct	Percent of dbspace used by this object
MinBlk	First block used by this object on this dbspace
MaxBlk	Last block used by this object on this dbspace; useful for determining which objects must be relocated before the dbspace is resized to a smaller size

Examples

Displays information about indexes in the Departments table:

```
sp_iqindexinfo 'table GROUPO.Departments';
```

Object	Dbspace-Name	ObjSize	DBSpPct	MinBlk	MaxBlk
GROUPO.Departments	iq_main	288K	1	1,045,496.00	1,048,891.00
GROUPO.Departments.ASIQ_IDX_T779_C1_FP	iq_main	176K	1	1,047,197.00	1,047,328.00
GROUPO.Departments.ASIQ_IDX_T779_C2_FP	iq_main	160K	1	1,047,213.00	1,047,324.00
GROUPO.Departments.ASIQ_IDX_T779_C3_FP	iq_main	184K	1	1,047,229.00	1,047,317.00

Object	Dbspace-Name	ObjSize	DBSpPct	MinBlk	MaxBlk
GROUPO.Departments.ASIQ_IDX_T779_C3_HG	iq_main	440K	1	1,048,421.00	1,048,796.00
GROUPO.Departments.ASIQ_IDX_T779_I4_HG	iq_main	288K	1	1,047,261.00	1,047,306.00

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
```

Applies to
Multiplex only.

Privileges
No special privileges are required to execute the procedure.

Description

Table 16. Column Descriptions

Column Name	Description
DiagMsgID	Uniquely identifies a diagnostic message
Description	Diagnostic message describing the issue found with DQP configuration

Table 17. 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	Logical server policy option dqp_enabled is set to 0
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
10	IQ_SHARED_TEMP dbspace is dynamically offline

Example
Sample output from the **sp_iqmpxcheckdqpconfig** procedure:

```
diagmsgid          description
3                Logical server policy option dqp_enabled is set to 0
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]
```

Applies to
Multiplex only.

Privileges
Requires MANAGE MULTIPLEX system privilege. Users without the MANAGE MULTIPLEX system privilege must be granted EXECUTE permission.

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
Show 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
          0000000000000000000000800350101000000c380000000000000
          0100000000000000000000000RFID:0100050100000000130000 0
          0000000001000000000000100RBID:010005010000000013000
```

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
```

Applies to

Multiplex only.

Privileges

Must have the MANAGE MULTIPLEX system privilege. Users without the MANAGE MULTIPLEX system privilege must be granted `EXECUTE` permission to run the stored procedure.

Description

sp_iqmpxfilestatus returns:

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
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_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 only the current node.

Syntax

```
sp_iqmpxincconnpoolinfo
```

Applies to
Multiplex only.

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

Privileges
MANAGE MULTIPLEX system privilege required. Users without the MANAGE MULTIPLEX system privilege must be granted `EXECUTE` permission.

Description
The **sp_iqmpxincconnpoolinfo** procedure returns:

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
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_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
```

Applies to
Multiplex only.

Privileges
MANAGE MULTIPLEX system privilege required. Users without the MANAGE MULTIPLEX system privilege must be granted `EXECUTE` permission.

Description
The **sp_iqmpxincheartbeatinfo** procedure returns:

Column Name	Data Type	Description	Values
server_id	unsigned int	Identifier for the server	
server_name	char(128)	Name of the server	

Column Name	Data Type	Description	Values
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.	

sp_iqmpxincheartbeatinfo Examples
Sample output of **sp_iqmpxincheartbeatinfo**.

```
server_id,server_name,last_positive_hb,
time_not_responding,time_until_timeout
2,'r2_dbsrv90210',2012-11-17
15:48:42.0,00:00:00,00:00:00
3,'w3_dbsrv90210',2012-11-17
15:48:42.0,00:00:00,00:00:00
```

If the elapsed time exceeds 24 hours, SAP 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 2013 11:57AM,11:59PM,11:59PM
3,'w4_mpx_cr_srv',Jan 14 2013
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_iqmpxincstatistics Procedure

Displays a snapshot of the aggregate statistics of internode communication (INC) status since server startup as of the moment of execution.

Syntax
sp_iqmpxincstatistics

Applies to
Multiplex only.

Privileges
Requires the MANAGE ANY STATISTICS system privilege. Users without MANAGE ANY STATISTICS system privilege must be granted `EXECUTE` permission.

Description
Returns:

Table 18. sp_iqmpxincstatistics Columns

Column Name	Data Type	Description
stat_name	char(128)	INC statistics name
stat_value	unsigned integer	INC statistics value

Table 19. Valid stat_name Values

Value	Description
NumSuspendedINC	Number of suspended INC connections since server startup
NumResumedINC	Number of resumed INC connections since server startup
NumDroppedSuspendedINC	Number of dropped INC connections that have been suspended (on coordinator only)
NumSuspendedTxnRollbackINC	Number of rolled back global DML transactions due to INC failure (on writer only)

Example
Shows one suspended and one resumed transaction:

```
sp_iqmpxincstatistics
stat_name                   stat_value
NumSuspendedINC             1
NumResumedINC               1
NumSuspendedTXNRollBackINC  0
```

sp_iqmpxinfo Procedure

Returns a row for every node in the multiplex. Can be run from any multiplex node.

Syntax

```
sp_iqmpxinfo
```

Applies to
Multiplex only.

Privileges
MANAGE MULTIPLEX system privilege required.

Description

The **sp_iqmpxinfo** procedure returns:

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.
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

- *Designated Failover Node* on page 27
- *Designating a Failover Node* on page 21
- *Checking Server Status in Interactive SQL* on page 18

sp_iqmpxsuspendedconninfo Procedure

Shows details about currently suspended connections and transactions on the coordinator node.

Syntax

sp_iqmpxsuspendedconninfo

Applies to

Multiplex only.

Privileges

Requires any of the following system privileges to see all suspended connections in the database:

- DROP CONNECTION system privilege
- MONITOR system privilege
- SERVER OPERATOR system privilege

No system privileges are required to see a user's own suspended connections. This procedure is owned by user DBO.

Description

Returns:

Table 20. sp_iqmpxsuspendedconninfo Columns

Column Name	Data Type	Description
ConnName	char(128)	Connection name
ConnHandle	unsigned integer	Connection identifier
GlobalTxnID	unsigned bigint	Global transaction identifier of active transaction on this connection
MPXServerName	char(128)	Name of the multiplex server where the INC connection originates
TimeInSuspendedState	integer	Total time, in seconds, spent by the connection in suspended state
SuspendTimeout	integer	Suspend timeout, in seconds (2*MPX_LIVENESS_TIMEOUT)

Example

sp_iqmpxsuspendedconninfo

```
ConnName              ConnHandle      GlobalTxnId
====================  =============   =============
'IQ_MPX_SERVER_P54'             14          112753

MPXServerName         TimeInSuspendedState
===================   ========================
'HP1_12356_IQ_mpx2'                       37

SuspendTimeout
===============
            360
```

sp_iqmpxvalidate Procedure

Checks multiplex configuration for inconsistencies.

Syntax

call dbo.sp_iqmpxvalidate ('show_msgs')

Applies to

Multiplex only.

Privileges

None.

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

Applies to

Multiplex only.

Privileges

No specific system privilege is required.

Description

Table 21. sp_iqmpxversioninfo Columns

Column	Data Type	Description
CatalogID	unsigned bigint	Catalog version on this server
VersionID	unsigned bigint	Latest version available on this server
OAVID	unsigned bigint	Oldest active version on this server
ServerType	char(1)	Type of server: “C” (Coordinator), “W” (Write Server) or “Q” (Query Server)

Column	Data Type	Description
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, **sp_iqsharedtempdistrib** displays shared temp space distribution for all nodes. If run from a secondary node, displays shared temp space usage for only 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
```

Applies to

Multiplex only.

Privileges

Requires the MANAGE ANY DBSPACE system privilege. Users without the MANAGE ANY DBSPACE system privilege must be granted `EXECUTE` permission.

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 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

Displays the number of blocks used by each object in the current database and the name of the dbspace in which the object is located.

Syntax

```
sp_iqspaceinfo [ `main
| [table table-name | index index-name] [...] `]
```

Applies to

Simplex and multiplex.

Privileges

Requires the MANAGE ANY DBSPACE system privilege. Users without the MANAGE ANY DBSPACE system privilege must be granted `EXECUTE` permission.

Description

For the current database, displays the object name, number of blocks used by each object, and the name of the dbspace. **sp_iqspaceinfo** requires no parameters.

The information returned by **sp_iqspaceinfo** is helpful in managing dbspaces.

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

If you supply no parameter, you must have at least one user-created object, such as a table, to receive results.

Example

This output is from the **sp_iqspaceinfo** stored procedure run on the `iqdemo` database. Output for some tables and indexes are removed from this example.

```
              Name                                  NBlocks     dbspace_name

Contacts                                           19         IQ_SYSTEM_MAIN
SalesOrderItems.DBA.ASIQ_IDX_T205_C5_FP             56       IQ_SYSTEM_MAIN
Contacts.DBA.ASIQ_IDX_T206_C10_FP                   55       IQ_SYSTEM_MAIN
Contacts.DBA.ASIQ_IDX_T206_C1_FP                    61       IQ_SYSTEM_MAIN
...
Contacts.DBA.ASIQ_IDX_T206_C9_FP                    55       IQ_SYSTEM_MAIN
Contacts.DBA.ASIQ_IDX_T206_I11_HG                   19       IQ_SYSTEM_MAIN
Customers                                          20         IQ_SYSTEM_MAIN
Customers.DBA.ASIQ_IDX_T207_C1_FP                   61       IQ_SYSTEM_MAIN
Customers.DBA.ASIQ_IDX_T207_C2_FP                   55       IQ_SYSTEM_MAIN
...
Customers.DBA.ASIQ_IDX_T207_I10_HG                  19       IQ_SYSTEM_MAIN
...
```

sp_iqspaceused Procedure

Shows information about space available and space used in the IQ store, IQ temporary store, RLV 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,
               out rlvLogKB          unsigned bigint,
               out rlvLogKBUsed      unsigned bigint)
```

Applies to

Simplex and multiplex.

Privileges

Requires the ALTER DATABASE, MANAGE ANY DBSPACE, or MONITOR system privileges. Users without one of these system privileges must be granted `EXECUTE` permission.

Usage

sp_iqspaceused returns several values as unsigned bigint out parameters. This system stored procedure can be called by user-defined stored procedures to determine the amount of main, temporary, and RLV 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.

If run on a multiplex database, this procedure applies to the server on which it runs. Also returns space used on IQ_SHARED_TEMP.

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 number of kilobytes of total IQ temporary store space in use by the database.
shTempTotalKB	The total IQ global shared temporary store space, in kilobytes.
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.
rlvLogKB	The total RLV store space, in kilobytes.
rlvLogKBUsed	The number of kilobytes of RLV store space in use by the database.

Example

sp_iqspaceused requires seven output parameters. Create a user-defined stored procedure **myspace** that declares the seven output parameters, then calls **sp_iqspaceused**:

```
create or replace 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 rvlt unsigned bigint;
    declare rvlu unsigned bigint;
    call sp_iqspaceused(mt,mu,tt,tu,gt,gu,lt,lu, rvlt, rvlu);
    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,
    if gu=0 then 0 else gu*100/gt endif as globalshtempPerCent,
    cast(lt/1024 as unsigned bigint) as shTempLocalMB,
    cast(lu/1024 as unsigned bigint) as shTempLocalKBUsed,
    if lt=0 then 0 else lu*100/lt endif as localshtempPerCent,
    cast(rvlt/1024 as unsigned bigint) as rlvLogKB,
    cast(rvlu/1024 as unsigned bigint) as rlvLogKBUsed;
end
```

To display the output of **sp_iqspaceused**, execute **myspace**:

```
myspace
```

sp_iqstatus Procedure

Displays a variety of SAP Sybase IQ status information about the current database.

Syntax

sp_iqstatus

Applies to

Simplex and multiplex.

Privileges

Requires the ALTER DATABASE, MANAGE ANY DBSPACE, MONITOR, or SERVER OPERATOR system privilege. Users without one of these system privileges must be granted `EXECUTE` permission.

Description

Shows status information about the current database, including the database name, creation date, page size, number of dbspace segments, block usage, buffer usage, I/O, backup information, and so on.

sp_iqstatus displays an out-of-space status for main and temporary stores. If a store runs into an out-of-space condition, **sp_iqstatus** shows `Y` in the store's out-of-space status display value.

Memory used by the row-level versioning (RLV) store can be monitored with **sp_iqstatus**. The **RLV memory limit** row displays the memory limit as specified by the `-iqrlvmem` server

option, or the sa_server_option `rlv_memory_mb`. The `RLV memory used` row displays the amount of memory used by the RLV store.

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

To display space that can be reclaimed by dropping connections, use **sp_iqstatus** and add the results from the two returned rows:

```
(DBA)> select * from sp_iqstatus() where name like '%Versions:%'
Execution time: 6.25 seconds
Name                 Value
---------------------------
Other Versions: 2 = 1968Mb
Active Txn Versions: 1 = C:2175Mb/D:2850Mb

(First 2 rows)
```

The above example output shows that one active write transaction created 2175MB and destroyed 2850 MB of data. The total data consumed in transactions and not yet released is 4818MB, or 1968MB + 2850MB = 4818MB.

sp_iqstatus omits blocks that will be deallocated at the next checkpoint. These blocks do however, appear in **sp_iqdbspace** output as type X.

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 see which versions are being used and the amount of space that can be recovered by releasing versions.

Example

Note: This example includes a sample user dbspace named `iq_main` that may not be present in your own databases.

The following output is from the **sp_iqstatus** stored procedure:

```
SAP Sybase IQ (TM)               Copyright (c) 1992-2013 by Sybase,
Inc.
                                        All rights reserved.
Version:                                16.0.0.160/120507/D/ELAN/
Sun_x64/OS 5.10/
                                        64bit/2012-05-07 17:36:36
Time Now:                               2013-05-16 09:53:13.590
Build Time:                             2013-05-07 17:36:36
File Format:                            23 on 03/18/1999
Server mode:                         IQ Multiplex Coordinator Server
Catalog Format:                         2
Stored Procedure Revision:              1
Page Size:                              131072/8192blksz/16bpp
Number of Main DB Files :               3
Main Store Out Of Space:                N
Number of Shared Temp DB Files:         0
```

```
Shared Temp Store Out Of Space:          N
Number of Local Temp DB Files :          1
Local Temp Store Out Of Space:           N
DB Blocks: 1-640000                      IQ_SYSTEM_MAIN
DB Blocks: 1045440-130101439             iqmain1
DB Blocks: 2090880-2346879               iqmain2
Local Temp Blocks: 1-384000              IQ_SYSTEM_TEMP
Create Time:                             2013-05-08 15:54:15.549
Update Time:                             2013-05-16 09:53:00.077
Local Temp Blocks:  1-1600               IQ_SYSTEM_TEMP
Create Time:                             2013-05-08 15:54:15.549
Update Time:                             2013-05-16 09:53:00.077
Main IQ Buffers:                         510, 64Mb
Temporary IQ Buffers:                    510, 64Mb
Main IQ Blocks Used:                  157379 of 1126400, 13%=1229Mb,
Max Block#: 2128363
Shared Temporary IQ Blocks Used:        0 of 0, 0%=0Mb, Max Block#: 0
Local Temporary IQ Blocks Used:          81 of 358400, 0%=0Mb, Max
Block#: 81
Main Reserved Blocks Available:          25600 of 25600, 100%=200Mb
Shared Temporary Reserved Blocks Available:  0 of 0, 0%=0Mb
Local Temporary Reserved Blocks Available:   25600 of 25600,
100%=200Mb
IQ Dynamic Memory:                       Current: 178mb, Max: 178mb
Main IQ Buffers:                         Used: 99, Locked: 0
Temporary IQ Buffers:                    Used: 5, Locked: 0
Main IQ I/O:                          I: L60904/P29 O: C5463/D11343/
P9486
                                         D:5450 C:51.3
Temporary IQ I/O:                     I: L12526/P0 O: C165/D319/P157
D:160 C:100.0
Other Versions:                          6 = 0Mb
Active Txn Versions:                     0 = C:0Mb/D:0Mb
Last Full Backup ID:                     0
Last Full Backup Time:
Last Backup ID:
Last Backup Type:                        None
Last Backup Time:
DB Updated:                              1
Blocks in next ISF Backup:               0 Blocks: =0Mb
Blocks in next ISI Backup:               0 Blocks: =0Mb
Main Tlvlog Size:                     Pages: 1, Recs: 193, Replays:
0/0
DB File Encryption Status:               OFF
```

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

- `I:` Input
- `L:` Logical pages read (“Finds”)
- `P:` Physical pages read
- `O:` Output
- `C:` Pages created

- `D:` Pages dirtied
- `P:` Physically written
- `D:` Pages destroyed
- `C:` Compression ratio

sp_iqtransaction Procedure

Shows information about transactions and versions.

Syntax

```
sp_iqtransaction
```

Applies to

Simplex and multiplex.

Privileges

Requires the MONITOR system privilege. Users without the MONITOR system privilege must be granted `EXECUTE` permission to run the stored procedure.

Description

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

sp_iqtransaction output does not include connections without transactions in progress. To include 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.

Column Name	Description
Name	The name of the application.
Userid	The user ID for the connection.
TxnID	The transaction ID of this transaction control block. The transaction ID is assigned during **begin transaction**. It appears in the `.iqmsg` file by the BeginTxn, CmtTxn, and PostCmtTxn messages, and is the same as the Txn ID Seq that is logged when the database is opened.
CmtID	The ID assigned by the transaction manager when the transaction commits. For active transactions, the CmtID is zero.

Column Name	Description
VersionID	For simplex and multiplex nodes, a value of 0 indicates that the transaction is unversioned, and the VersionID has not been assigned. For the multiplex coordinator, the VersionID is assigned after the transaction establishes table locks. Multiplex secondary servers receive the VersionID from the coordinator. The VersionID is used internally by the SAP 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 SAP Sybase IQ implementation details and is subject to change in the future. Currently, transaction states are `NONE`, `ACTIVE`, `ROLLING_BACK`, `ROLLED_BACK`, `COMMITTING`, `COMMITTED`, and `APPLIED`. `NONE`, `ROLLING_BACK`, `ROLLED_BACK`, `COMMITTING` and `APPLIED` are transient states with a very small life span. `ACTIVE` indicates that the transaction is active. `COMMITTED` indicates that the transaction has completed and is waiting to be `APPLIED`, at which point a version that is invisible to any transaction is subject to garbage collection. Once the transaction state is `ROLLED_BACK`, `COMMITTED`, or `APPLIED`, ceases to own any locks other than those held by open cursors.
ConnHandle	The ID number of the connection.
IQConnID	The ten-digit connection ID that is included 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, 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 SAP 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 SAP 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	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 global transaction ID associated with the current transaction, 0 (zero) if none.
VersioningType	The snapshot versioning type of the transaction; either table-level (the default), or row-level. Row-level snapshot versioning (RLV) applies only to RLV-enabled tables. Once a transaction is started, this value cannot change.
Blocking	Indicates if connection blocking is enabled (True) or disabled (False). You set connection blocking using the **BLOCKING** database option. If true, the transaction blocks, meaning it waits for a conflicting lock to release before it attempts to retry the lock request.

Column Name	Description
BlockingTimeout	Indicates the time, in milliseconds, a transaction waits for a locking conflict to clear. You set the timeout threshold using the **BLOCKING_TIMEOUT** database option. A value of 0 (default) indicates that the transaction waits indefinitely.

Example

Example **sp_iqtransaction** output:

```
Name   Userid  TxnID  CmtID VersionID    State    ConnHandle  IQConnID
====== ====== ====== ====== =========  ========== ===========
========
red2      DBA  10058  10700     10058  Active     419740283       14


MainTableKBCr      MainTableKBDr    TempTableKBCr TempTableKBDr
============= ================== ================ =============
           0                  0            65824             0


TempWorkSpaceKB TxnCreateTime               CursorCount SpCount
SpNumber
==============
========================     =========== ======= ========
       0        2013-03-26 13:17:27.612            1       3        2


MPXServerName  GlobalTxnID   VersioningType  Blocking
BlockingTimeout
=============  ===========   ==============  ========
===============
        (NULL)          0     Row-level       True
0
```

sp_iqversionuse Procedure

Displays version usage for the IQ main store.

Syntax

```
sp_iqversionuse
```

Applies to

Simplex and multiplex.

Privileges

Requires the MONITOR system privilege. Users without the MONITOR system privilege must be granted `EXECUTE` permission to run the stored procedure.

Description

The **sp_iqversionuse** system stored procedure helps troubleshoot situations where the databases uses excessive storage space due to multiple table versions.

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

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.

The amount of space is expressed as a range because the actual amount typically depends on which other versions are released. The actual amount of space released can be anywhere between the values of MinKBRelease and MaxKBRelease. The oldest version always has MinKBRelease equal to MaxKBRelease.

The WasReported column is used in a multiplex setting. WasReported indicates whether version usage information has been sent from the secondary server to the coordinator. WasReported is 0 initially on a coordinator for new versions. WasReported changes to 1 once the database server replicates version usage information back to the coordinator.

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.

Table 22. sp_iqversionuse Columns

Column Name	Description
VersionID	In simplex databases, the VersionID is displayed as zero. 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 SAP Sybase IQ in-memory catalog and the SAP Sybase IQ transaction manager to uniquely identify a database version to all nodes within a multiplex database.
Server	The server to which users of this version are connected
IQConnID	The connection ID using this version
WasReported	Indicates whether the server has received usage information for this version

Column Name	Description
MinKBRelease	The minimum amount of space returned once this version is no longer in use
MaxKBRelease	The maximum amount of space returned once this version is no longer in use

Examples

Sample output from the **sp_iqversionuse** system procedure:

```
VersionID Server         IQConnID WasReported
========= ======         ======== ===========
        0 ab2ab_iqdemo          9           0

MinKBRelease  MaxKBRelease
============  ============
           0             0
```

The following examples show multiplex output. 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.

Output 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

Output 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.

Backup Utility (dbbackup)

The **dbbackup** utility truncates the database name to 70 characters and creates a target file with a truncated name. SAP 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 SAP Sybase IQ multiplex servers at startup.

Table 23. 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. 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.

Startup switch	Values	Description
-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) , mipcmaxt (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))` .
-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 Technical Support. If the value is not set, the default maximum pool size is `MAX (number of cores, mipcmint)` .

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.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W