SYBASE®

An **SAP** Company

**Administration: Backup, Restore, and Data
Recovery**

SAP Sybase IQ 16.0 SP01

Contents

Data Backup, Recovery, and Archiving

To protect your data, schedule and perform regular backups. You can also use read-only hardware to archive non-modifiable data for easy access.

Data Protection

SAP® Sybase® IQ provides a full set of features that protect you from two types of computer failure, and from database inconsistency.

- A *system failure* occurs when the computer or operating system goes down while there are partially completed transactions. This could occur when the computer is inappropriately turned off or rebooted, when another application causes the operating system to crash, or because of a power failure.
- A *media failure* occurs when the database file, the file system, or the device storing the database file, becomes unusable.

After a system failure, SAP Sybase IQ can usually recover automatically, so that you may not need to restore your database.

After media failure, or if for any reason the data in your database is inconsistent, you must restore your database. To protect your data in all of these situations, make regular backups of your databases. In particular, you should back up your database each time you finish inserting any large quantities of new data into the database.

When failures occur, the recovery mechanism treats transactions properly, as atomic units of work: any incomplete transaction is rolled back and any committed transaction is preserved. This ensures that even in the event of failure, the data in your database remains in a consistent state.

How to Back Up Databases

Use the **BACKUP** command to back up your SAP Sybase IQ database.

Backup includes both the SAP Sybase IQ data (the IQ store) and the underlying SAP® Sybase SQL Anywhere database (the catalog store).

Backup runs concurrently with read and write operations in the database. By contrast, during a restore no other operations are allowed on that database.

You must be connected to a database to back it up. The **BACKUP** command has no way to specify another database.

Types of Data Stores

SAP Sybase IQ data stores consist of one or more files.

They can contain both user data and internal database structures used for startup, recovery, backup, and transaction management. Typically, an SAP Sybase IQ database has the following stores:

- *db-name.db* is the catalog dbspace containing the system tables and stored procedures describing the database and any standard SQL Anywhere database objects you add. It is known as the catalog store, and has the dbspace-name SYSTEM. You can create additional dbspaces in the catalog store.
- *db-name.iq* is the main data dbspace containing the SAP Sybase IQ table data and indexes. It is known as the IQ store, and has the dbspace-name IQ_SYSTEM_MAIN. The dbfile name matches dbspace-name, IQ_SYSTEM_MAIN. You can create multiple dbspaces in the IQ store, and each dbspace can hold multiple dbfiles, including IQ_SYSTEM_MAIN.
- *db-name.iqtmp* is the initial temporary dbspace containing the temporary tables generated by certain queries. It is known as the IQ temporary store and has the dbspace-name IQ_SYSTEM_TEMP. You can add dbfiles to the IQ temporary store.

Any of these stores, and the log files, are possible areas of failure.

Types of Backups

There are four ways to back up SAP Sybase IQ data.

- Database backup
- Operating system-level backup
- Virtual backup
- Archive backup (for log files)

Types of Database Backups

SAP Sybase IQ provides four types of database backups:

- *Full backup* makes a complete copy of the database.
- *Virtual backup* copies all of the database except the table data from the IQ store.
- *Incremental backup* copies all transactions since the last backup of any type.
- *Incremental-since-full backup* copies all transactions since the last full backup.

All these backup types fully back up the catalog store. In most cases, the catalog store is much smaller than the IQ store. If the catalog store is larger than (or nearly as large as) the IQ store, however, incremental backups of IQ are bigger than you may want or expect.

Incremental virtual backup is supported using the **BACKUP** statement.

Temporary store data is not backed up. However, the metadata and any other information needed to recreate the temporary store structure is backed up.

Backing Up the IQ Store and Catalog Store

This procedure summarizes backup steps.

Prerequisites

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

Task

1. Connect to the server using an account with the BACKUP DATABASE system privilege.

   Note: For a multiplex database, you must connect to the coordinator.

2. Run a **BACKUP** command, which backs up the following files:

   - 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

- *Preparing for Backup* on page 5
- *BACKUP Statement* on page 111

Data in Backups

BACKUP backs up committed data only.

Backups begin with a commit and an automatic checkpoint. At this point, the backup program determines what data will be backed up. It backs up the current snapshot version of your database as of the time of this checkpoint. Any data that is not yet committed when this checkpoint occurs is not included in the backup.

A second automatic checkpoint occurs at the end of backup. Any data that is committed while the backup is in progress is included in any subsequent backups.

SAP Sybase IQ backs up only those recoverable database blocks actually in use at the time of backup. Free blocks are not backed up.

SAP Sybase IQ backs up the database files and the catalog information that pertains to the SAP Sybase IQ database to which you are connected. It does not back up the transaction log file. It does not use the transaction log to restore the database.

If for any reason all the commands in the transaction do not process properly, or your database is missing files, the backup fails.

The Transaction Log In Backup, Restore, and Recovery

SAP Sybase IQ uses the transaction log file during recovery from a system failure.

It does not use the transaction log to restore an SAP Sybase IQ database, to recover committed IQ transactions, or to restore the catalog store for an SAP Sybase IQ database. For a full restore, the transaction log must not exist. You must delete this file before starting a full restore.

Distribution of Backup Data

BACKUP always makes a full backup of the catalog store on the first archive device, and then backs up the data from the IQ store in parallel across all of the devices you specify.

Blocks are not evenly distributed across archive media. You may have more on one device than others, depending on the processing speed of individual threads.

Note: The distribution of backup data is important because sets of files must be restored in the order in which they were backed up.

Ensure that your Database is Consistent

Although **BACKUP** does check that all necessary files are present before backing up your database, it does not check internal consistency.

For a more thorough check, you can run the stored procedure **sp_iqcheckdb** before making a backup.

Select Archive Devices

You can back up any SAP Sybase IQ database onto magnetic tape or disk, including WORM devices.

SAP Sybase IQ supports backup and restore using multiple tape drives at near device speeds, or to multiple disks if disk striping is in use. Specify the backup device name in the *archive_device* parameter of the **BACKUP** command.

Disk Backup Requirements

Disk backups must go to a file system; raw disk is not supported as a medium for file system backup. All disks on a redundant array of independent devices (RAID) device are treated as a single device.

Tape Backup Requirements

If you regularly back up large databases, use multiple tape drives. Use Digital Linear Tape (DLT) drives, if they are supported for your platform.

SAP Sybase IQ **BACKUP** can support the following tape drives:

- Digital Linear Tape (DLT) on UNIX systems
- 4 mm Digital Data Storage (DDS)

- 8 mm

SAP Sybase IQ also allows Stacker drives with multiple tapes.

SAP Sybase IQ **BACKUP** does not support jukeboxes or robotic loaders. If you need them, use a third party media manager.

SAP Sybase IQ **BACKUP** does not support fixed-length tape devices on UNIX systems, like Quarter Inch Cartridge (QIC) drives.

Platform-specific Backup Requirements

Be aware of backup requirements for AIX and IBM Linux.

Be aware of the following platform-specific backup requirements:

- Tape devices on AIX systems can be configured for either fixed- or variable-length block mode. See the *Installation and Configuration Guide* for information on how to show and change the block mode. SAP Sybase IQ **BACKUP** does not support fixed-length block mode.
- On IBM Linux on POWER, to back up an IQ database to SCSI tape, you must set the block size of the device to accept variable-length data transfer. Before performing any IQ backups, set the SCSI tape device's default block size. Log in as superuser and run the Linux shell command **mt**, as follows:

```
mt -f /dev/st0 defblksize 0
```

Limit on the Number of Backup Devices

Specify multiple TO clauses in the **BACKUP** statement to parallelize the backup operation.

Use 36 or fewer **TO** clauses in a **BACKUP** command.

This limit affects all versions of SAP Sybase Risk Analytics Platform and SAP Sybase RAP - The Trading Edition™.

Backup Guidelines

Keep backup commands small. Large numbers of devices increase I/O and hardware contention.

- As a practical guideline, use roughly 1 device per core on the machine to saturate CPU usage.
- Use up to 2 devices per core on faster systems.

Preparing for Backup

In order to run **BACKUP**, you must meet the requirements described in the sections that follow.

See also

Obtaining DBA Privileges

You need DBA privileges on a database to run **BACKUP** or **RESTORE**.

You need BACKUP DATABASE system privilege to run BACKUP and SERVER OPERATOR system privilege to run RESTORE.

Rewind Tapes

SAP Sybase IQ does not rewind tapes before using them.

You must ensure the tapes used for backup or restore operations are at the correct starting point before putting them in the tape device.

Tapes are rewound after the backup if you are using a rewinding device. If your tape device automatically rewinds tapes, take care that you do not overwrite any information on the tape.

Retain Old Disk Backups

BACKUP overwrites existing disk files of the same name.

If you need to retain a backup, when you create a new backup either use different file or path names for the archive devices, or move the old backup to another location before starting the backup.

Two Ways to Run BACKUP

You can run **BACKUP** in two ways.

- **Attended –** In attended mode, **BACKUP** assumes that an operator is present, and prompts you to mount the archive media when necessary. With this method, you must run **BACKUP** interactively from the command line.
- **Unattended –** In unattended mode, **BACKUP** assumes that no operator is present, and does not issue prompts. Instead, you must make appropriate estimates of the space required, and set up your devices accordingly. Any error is considered fatal.

In some cases, you can use third party software to create backups. Such products can be particularly useful for unattended backups.

Note: You can run **BACKUP** from a batch script or procedure, as well as from Interactive SQL. You can also automate backups using an event handler. See *Administration: Database > Automate Tasks Using Schedules and Events*.

Estimate Media Capacity

Before you do a backup, be sure that your archive media has sufficient space.

When you estimate available space on disk or tape, keep in mind these rules:

- You need enough room for a full backup of the catalog store, as well as the full or incremental backup of the IQ store. If your catalog store holds SQL Anywhere data in addition to the SAP Sybase IQ system tables, you need room to back up this data as well.

- You do not need to include space for the transaction log, as this log is not backed up.
- For tape backups, the first tape set you specify must be able to hold the full backup of the catalog store, including any non-IQ data in the catalog store. (A tape set consists of one or more backup tapes produced on a given archive device.)
- For stacker devices that hold multiple tape drives, all tapes for a given device must be the same size.

Start a new tape for every backup.

Before starting a backup to disk, SAP Sybase IQ first tests whether there is enough disk file space for the backup. For an operator-attended backup to disk, if there is not enough space, **BACKUP** prompts you to move some files from the disk before it writes any data. The backup does not start until you provide more disk space.

Likewise, if you run out of space during an attended disk backup, **BACKUP** closes all open backup files and waits until it detects that you have cleared some space. Then it restarts with new backup files. You can also stop the backup if you prefer.

By default, you must provide at least 8KB of free disk space before the backup resumes.

Unattended backup cannot prompt you to provide more space. Unless enough space is available, unattended backup fails. **BACKUP** treats size estimates differently for unattended backups.

For an operator-attended backup to tape, **BACKUP** simply begins the backup. If it runs out of room, you must mount additional tapes.

Running Backups

Use the BACKUP statement to run backups.

Concurrency and Backups

You can run backups concurrently with most other database operations.

The exception are:

- No metadata changes can occur while the catalog store is backed up.
- No commands that issue checkpoints or DBCC can be run during backup.

Be aware, however, that transactions that have not committed when you start a backup are not backed up. If a system or media failure occurs during backup, you cannot restore uncommitted transactions.

Once a backup is started, you cannot execute a **CHECKPOINT** command.

Specify Operator Presence

ATTENDED ON or **OFF** controls whether or not human intervention is expected when new tapes or disk files are needed.

The default is **ON**.

For unattended backups to disk, **BACKUP** does not prompt you to add more disk space. If you run out of space, an error occurs and **BACKUP** halts.

For unattended backups to tape, **BACKUP** does not prompt for a new tape to be loaded. The **SIZE** and **STACKER** options determine what happens if you run out of space.

Unattended Backup

With the **ATTENDED OFF** option, you can specify that no operator will be present during a backup.

SAP Sybase IQ supports two unattended backup features:

* The operator does not need to respond to prompts during the backup.
* The archive devices can be stacker drives, which automatically load a set of tapes into a single drive. You can use stacker drives for both attended and unattended backups.

Unattended backup tries to detect all possible reasons for a backup failing except tape media failure, and report any potential errors before attempting the backup, such as available space on disk or tape, and consistent size and block factor.

For unattended backup to disk, SAP Sybase IQ first tests whether there is enough free disk space for the backup, However, it does not pre-allocate the backup files to reserve the space. If another user writes to that disk and as a result there is not enough room for the backup, the backup fails when disk space runs out.

For backup to tape, you must estimate how much data each tape will hold, and specify that number of kilobytes in the **TO** *archive_device* parameter of the **BACKUP** command. The backup program checks information stored internally to see how much room it needs to back up your database. If it determines that there is enough room on the tape, the backup proceeds. However, if you overestimate the amount of space available on the tape(s) and the backup runs out of space, the backup fails at that point.

If you omit the **SIZE** parameter for an unattended backup, the entire backup must fit on one tape.

If you are using a third-party backup product, the vendor information string needs to convey any information needed for the backup, such as the specification of devices, size of files, and stacker drives. See your vendor's documentation for details.

Note: SAP Sybase IQ does not permit unattended restore.

Specify the Type of Backup

FULL | INCREMENTAL | INCREMENTAL SINCE FULL specifies the type of backup.

Choose one:

* **FULL** causes a full backup of both the catalog store and the IQ store. **FULL** is the default action.

For a virtual backup, you can use the **VIRTUAL DECOUPLED | VIRTUAL ENCAPSUATED** options of the **BACKUP** statement.

- **INCREMENTAL** makes a full backup of the catalog store, and then backs up all changes to the IQ store since the last IQ backup of any type.
- **INCREMENTAL SINCE FULL** makes a full backup of the catalog store, and then backs up all changes to the IQ store since the last full IQ backup.

INCREMENTAL and **INCREMENTAL SINCE FULL** virtual backups are supported using the **VIRTUAL DECOUPLED** and **VIRTUAL ENCAPSULATED** options of the **BACKUP** statement.

You may restrict full, incremental-since-full, or incremental backup to the set of read-write files in the databases using the READWRITE FILES ONLY keywords. The read-write dbspaces or files that are backed up must belong to the IQ main store. The backed up files are selected when the backup command checks the read-write status in the catalog.

An IQ backup may back up a set of read-only dbspaces and/or read-only files. The read-only dbspaces or files must belong to the IQ main store. The backed up files are user selected.

Specifying Virtual Backup

The **VIRTUAL DECOUPLED | VIRTUAL ENCAPSUATED** '*shell-command* options specify the type of virtual backup.

The *shell-command* variable of the VIRTUAL ENCAPSULATED parameter allows shell commands to execute a system-level backup as part of the backup operation.

Specifying Archive Devices

The **TO** *archive_device* clause indicates the destination disk file(s) or system tape drive(s) for the backup and controls the number of archive devices.

Backup File Names for Backup to Disk
BACKUP always assigns file names to disk backup files by appending a suffix to the *archive_device* name you specify.

The suffix consists of "." followed by a number that increases by one for each new file. For example, if you specify /iqback/mondayinc as the *archive_device*, the backup files are /iqback/mondayinc.1, /iqback/mondayinc.2, and so on. This convention allows you to store as large a backup as you need, while allowing you control over the file size; see the **SIZE** option for details. Your file system must support long file names to accommodate this convention.

You must make sure that the directory names you specify for the *archive_device* exist. **BACKUP** does not create missing directories. If you try to start a backup in a directory that does not exist, the backup fails.

You should avoid using relative path names to specify the location of disk files. **BACKUP** interprets the path name as relative to the location where the server was started, which you may

---

not be able to identify with certainty when you do a backup. Also, if there is data in other directories along the path, you may not have enough room for the backup.

Positioning Tape Devices

BACKUP does not position tapes for you. You must position the tape appropriately before starting your backup, and be sure that you do not overwrite any of the backup if you use a rewinding tape device. For these reasons, a non-rewinding tape device is preferred. See the operating system documentation for your platform for appropriate naming conventions.

Specifying Tape Devices on UNIX

Here are examples of how you specify non-rewinding tape devices on UNIX platforms:

- On Solaris platforms, insert the letter n for "no rewind" after the device name, for example, `'/dev/rmt/0n'`.
- On IBM AIX platforms, use a decimal point followed by a number that specifies the appropriate compression with rewind setting, for example, `'/dev/rmt0.1'`.
- On HP-UX platforms, use '0m' to specify the default tape mechanism and 'n' for "no rewind," for example, `'/dev/rmt/0mn'`.

Warning! If you misspell a tape device name and write a name that is not a valid tape device on your system, **BACKUP** assumes it is a disk file.

Specifying Tape Devices on Windows

Windows systems do not specify rewind or no rewind devices and only support fixed-length I/O operations to tape devices. SAP Sybase IQ requires variable-length devices. It does additional processing to accommodate fixed-length tape I/O on Windows systems.

While Windows supports tape partitioning, SAP Sybase IQ does not use it, so do not use another application to format tapes for SAP Sybase IQ backup or restore. On Windows, 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.

Specify the Size of Tape Backups

The **SIZE** option of the **TO** clause identifies the maximum size of the backed up data on that stripe, in KB.

If you use the SAP Sybase-provided backup (as opposed to a third party backup product), you should specify **SIZE** for *unattended* tape backups on platforms that do not reliably detect the

---

end-of-tape marker. Note that the value of **SIZE** is per output device. No volume used on the corresponding device can be shorter than this value. Although SAP Sybase IQ does not require you to specify **SIZE** for an *attended* tape backup, it is always best to supply an accurate size estimate.

During backup, if any tape runs out of space and you have not specified **SIZE**, you get an error. If any tape runs out of space before the specified size, you do not get an error immediately; instead, here is what happens:

- For attended backups with **SIZE** and **STACKER** specified, Backup tries to open the next tape.
- For attended backups with **SIZE** specified but not **STACKER**, Backup asks you to put in a new tape.
- For unattended backups with **SIZE** and **STACKER** specified, Backup tries to open the next tape. If there are no volumes available, or if you did not specify **STACKER**, you get an error.

Any additional tapes do not contain the header information needed for a restore, so you must be careful to mount tapes in order during the restore or your database could become inconsistent.

On Windows, there are special requirements for the **SIZE** option on tape devices:

- The value of **SIZE** must be a multiple of 64. Other values are rounded down to a multiple of 64.
- If you do not specify **SIZE** explicitly, it is automatically set to 1.5GB.

Specify the Size of Disk Backups
The **SIZE** option of the **TO** clause identifies the maximum size of the backed up data on that stripe, in KB. Note that the value of **SIZE** is per output device.

If you use the SAP-provided backup, either attended or unattended, specify **SIZE** if any disk file you name as an *archive_device* is larger than the default of 2GB (UNIX) or 1.5GB (Windows).

During backup, when the amount of information written to a given *archive_device* reaches **SIZE**, backup closes the current file and creates another one of the same name with the next ascending number appended to the file name.

For example, if you specify one *archive_device*, a disk file called `janfull`, and you specify SIZE 200000 for a maximum 200MB file, but your backup requires 2GB, then **BACKUP** creates ten 200MB files: `janfull.1`, `janfull.2`,...`janfull.10`. You must ensure that your disk can accommodate this much data before performing the backup.

Specify Stacker Devices
The **STACKER** option of the **TO** clause indicates that you are backing up to an automatically loaded multitape stacker device, and specifies the number of tapes in that device. When

---

ATTENDED is ON and STACKER is specified, BACKUP waits indefinitely for the next tape to be loaded. All tapes in a given stacker device must be the same size.

Specify Devices for Third Party Backups

Note: Do not specify SIZE or STACKER if you are using a third party backup product, as size information is conveyed in the *vendor_specific_information* string.

Other Backup Options

You may want to set a number of other BACKUP command options to customize your backup.

Specifying the Block Factor
BLOCK FACTOR specifies the number of IQ blocks to write to the archive device at one time.

It must be greater than 0, or BACKUP returns an error message. BLOCK FACTOR defaults to 25 on UNIX platforms. On Windows, the default BLOCK FACTOR is based on the block size of your database. For example, if the block size is 512 bytes, BLOCK FACTOR is 120 blocks. If the block size is 32KB, BLOCK FACTOR is 1 block.

This parameter also controls the amount of memory used for buffers during the backup, and has a direct impact on backup performance. The effects of the block factor are a function of disk subsystem speed, tape speed, and processor speed. Some systems have better backup performance with a smaller block factor, while others may have better backup performance with a larger one. See your platform operating system documentation for information about your platform's optimal I/O size and block factor.

Specify Error Checking
CRC ON or OFF activates or deactivates 32-bit cyclical redundancy checking on a per block basis.

(BACKUP also uses whatever error detection is available in the hardware.) With CRC ON, the checksums computed on backup are verified during any subsequent RESTORE operation. The default is CRC ON.

Add Comments
WITH COMMENT specifies a string up to 32KB long as part of the header information for the backup archive.

If you omit this option, BACKUP enters a NULL. You can view the comment string by executing a RESTORE DATABASE FROM CATALOG ONLY, or by displaying the backup log, `backup.syb`, that SAP Sybase IQ provides.

Wait for Tape Devices

During backup and restore operations, if SAP Sybase IQ cannot open the archive device (for example, when it needs the media loaded), the server waits for ten seconds and tries again.

The server continues these attempts indefinitely, until either the operation succeeds or is terminated with a Ctrl+C. A message is written to the server `.stderr` file. There is no console notification that the server cannot open the archive device.

Backup and Restore Using Read-Only Hardware

SAP Sybase IQ supports read-only hardware for both backup and restore operations.

The following rules apply:

- SAP Sybase IQ prevents writes to a read-only device during restore because the device may be frozen in read-only mode at the hardware level.
- Virtual backup will not back up or restore the header block of a read-only dbspace or any other block on a read-only dbspace. Since a read-only dbspace is guaranteed never to change, virtual backup and restore need only restore a read-only dbspace after media failure of the read-only dbspace.
- Non-virtual full backup will back up all dbspaces, regardless of mode.
- Non-virtual incremental backup will not back up read-only dbspaces that:
    - Were read-only at the time of the previous backup that the incremental backup depends on,

        and
    - Have not been altered since.

    The contents of such dbspaces are wholly contained by a previous depends-on backup. Read-only dbspaces that have been altered since the time of the depends-on backup are backed up.

Backup Examples

Example topics demonstrate backup options.

Example 1 — Full Backup

This example makes a full, attended backup of the database `iquser` to two tape devices on UNIX. Before running this backup you must position the tapes to the start of where the backup files will be written, and connect to `iquser`. Then issue the following command:

```
BACKUP DATABASE
TO '/dev/rmt/0n'
TO '/dev/rmt/1n'
WITH COMMENT 'Jan 18 full backup of iquser'
```

The catalog store is backed up first, to `/dev/rmt/0n`. The IQ store is backed up next, to both tapes.

Example 2 — Incremental Backup

To make an incremental backup of the same database, this time using only one tape device, issue the command as follows:

```
BACKUP DATABASE
INCREMENTAL
TO '/dev/rmt/0n' SIZE 150
WITH COMMENT 'Jan 30 incremental backup of iquser'
```

Other Examples

An example of how to restore this database from these two backups is provided later in this chapter.

See also

*   *BACKUP Statement* on page 111

Recovery from Errors During Backup

There are two likely reasons for a failed backup: insufficient space, or hardware failure.

Problems with third party software could also cause a failure.

Checking for Backup Space

BACKUP uses the **STACKER** and **SIZE** parameters to determine whether there is enough space for the backup.

*   For disk backups, if it decides that you have not provided enough space, it fails the backup before actually writing any of the data.
*   If it decides that there is enough space to start the backup, but then runs out before it finishes (for example, if your estimate is incorrect, or if a user in another application fills up a lot of disk space while your backup is in progress), an attended backup prompts you to load a new tape, or to free up disk space. An unattended backup fails if it runs out of space.
*   If neither **STACKER** nor **SIZE** is specified, backup proceeds until it completes or until the tape or disk is full. If you run out of space, an attended backup prompts you to load a new tape, or to free up disk space; an unattended backup fails.

Recovery Attempts

If a backup fails, the backup program attempts to recover.

The recovery process is:

*   If backup fails during either the checkpoint at the start of backup or the checkpoint when backup is complete, it performs normal checkpoint recovery.
*   If backup fails between checkpoints, it rolls back the backup.
*   If the system fails at any time between the initial and final checkpoint and you must restore the database, you must do so using an older set of backup tapes or disk files.

- If the system fails during the final checkpoint after a **FULL** backup, you can restore from the backup tapes or files you have just created.

After You Complete a Backup

To move a database or one of its dbspaces, you need to know the name of every dbspace in the database when the backup was made.

SAP Sybase IQ includes a mechanism that verifies an existing SAP Sybase IQ database backup using the **VERIFY** clause of the **RESTORE** SQL statement.

Performing Backups with Non-Sybase Products

SAP Sybase IQ supports backup and restore using a number of third-party products. The package you use must conform to the Adaptive Server Enterprise Backup Interface. Check the documentation for your product to be sure that it supports SAP Sybase databases.

To perform such a backup or restore, you issue the **BACKUP** or **RESTORE** statement as if you were using SAP Sybase IQ to perform the operation, with the following exceptions:

- For each *archive_device*, instead of specifying the actual device name, specify a string in the following format:
  ```
  dll_name::vendor_specific_information
  ```
- Do not specify the **STACKER** or **SIZE** parameters.

The *dll_name* corresponds to a Dynamic Link Library loaded at run time. It can be from 1 to 30 bytes long, and can contain only alphanumeric and underscore characters. The *dll_name* must be the same for each *archive_device*.

The content of *vendor_specific_information* varies by product, and can differ for each *archive_device*. The total string (including *dll_name*:: and vendor information) can be up to 255 bytes long.

The backup program passes vendor information to the third-party program automatically. When you request a third-party backup, it places this information in the backup header file, and writes the header file on the first tape or disk file actually created for each *archive_device* you specify.

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 IQ database in this way, make sure it is certified. See the SAP Sybase Certification Reports for the SAP Sybase IQ product in *Technical Documents*.

Virtual Backups

A virtual backup, sometimes called a NULL backup, backs up all of an IQ database except the IQ store table data.

You must make a separate operating system-level copy of the corresponding IQ store. To restore from a virtual backup, you must first restore the corresponding system-level copy of the IQ store and then proceed with the IQ full restore of the virtual backup.

A virtual backup backs up:

- All IQ catalog data
- All IQ metadata
- All metadata in the IQ store not specific to individual tables. (Includes the freelist, backup and checkpoint information.)

A virtual backup does not back up data or metadata from tables other than those mentioned above.

To make a virtual backup, specify either the **VIRTUAL DECOUPLED** or **VIRTUAL ENCAPSULATED** parameter in the **BACKUP** command when performing a full IQ Backup. The **VIRTUAL** parameters prevent IQ from copying table data and metadata in the IQ store to the backup file.

Types of Virtual Backups

There are two types of virtual backup.

- **Encapsulated virtual backup** – A restore of the system-level backup followed by a restore of the IQ virtual backup results in a fully restored database.
- **Decoupled virtual backup** – A restore of the system-level backup followed by a restore of the IQ virtual backup followed by an incremental-since-full restore results in a fully restored database.

Performing Encapsulated Virtual Backups

For the system-level backup of table data to be consistent with the virtual backup without additional steps, the system-level backup must be made during the backup command and by the backup transaction.

The parameter `VIRTUAL ENCAPSULATED` *'shell-command'* allows arbitrary shell commands to be executed as part of the backup operation to guarantee these semantics. If the shell commands return a non-zero status, the backup operation returns an error. The user must guarantee that the shell commands correctly perform the system-level backup.

Enter a **BACKUP DATABASE** command with the FULL VIRTUAL ENCAPSULATED clause in Interactive SQL.

Use a SQL statement similar to:

```
BACKUP DATABASE FULL VIRTUAL ENCAPSULATED
'dd if=iqdemo.iq of=iqdemo.iq.copy'
TO 'iqdemo.full'
```

Restoring from Encapsulated Virtual Backup

Follow these steps to restore from encapsulated virtual backup.

1. Restore the system-level copy of the IQ store.
2. Perform a full IQ restore from the backup file.
3. Start the IQ database.

Performing Decoupled Virtual Backups

If the system-level backup is done outside the backup transaction, the IQ store backup will not be consistent with the IQ backup file.

However, a non-virtual IQ incremental backup together with the Virtual full backup will represent a consistent database. This is because the IQ incremental backup will copy all IQ store data and metadata that have changed during or since the Virtual full backup. Note that even the automatic commit and checkpoint that are part of the backup command modify the IQ store, making an independent system-level backup inconsistent. Trying to use the database without applying the incremental restore will give unpredictable results.

1. Perform a full IQ backup, using a SQL statement similar to the following:

   ```
   BACKUP DATABASE FULL VIRTUAL DECOUPLED
   TO 'iqdemo.full'
   ```

2. Perform a system-level backup of the IQ store with a shell command:

   ```
   dd if=iqdemo.iq of=iqdemo.iq.copy
   ```

3. Perform a non-virtual incremental IQ backup:

   ```
   BACKUP DATABASE INCREMENTAL SINCE FULL
   TO 'iqdemo.isf'
   ```

Restoring from a Decoupled Virtual Backup

Follow these steps to restore from a decoupled virtual backup.

1. Restore the system-level copy of the IQ store, for example:

   ```
   dd if=iqdemo.copy of=iqdemo.iq
   ```

2. Restore from the IQ full backup file.

   ```
   RESTORE DATABASE iqdemo.db FROM 'iqdemo.full'
   ```

3. Restore from the IQ incremental backup file.

   ```
   RESTORE DATABASE iqdemo.db FROM 'iqdemo.isf'
   ```

4. Start the IQ database.

Virtual Backup with SAN Snapshot or Shadow Hardware

Storage Area Network (SAN) snapshot or shadow hardware provides more flexibility in the backup process by allowing the system-level backup to take place on the shadow copy rather than on the main database.

In place of the system-level copy that is part of the virtual backup, the shadow can instead be separated. A system-level backup can then be performed against the shadow copy of the IQ store. This allows the full backup to complete quickly.

System-Level Backups

The **BACKUP** command is the most reliable method you can use to back up IQ data. If you are careful to follow the correct procedures, you can use system-level backups for an IQ database.

You must follow these procedures when using system-level backups for backing up your IQ database. If you attempt to restore your IQ database files from a system-level backup without these safeguards in place, you are likely to cause data loss or inconsistency, either from activity in the database while the system-level backup occurred, or from missing files.

Shut Down the Database

Shut down your SAP Sybase IQ database before a system-level backup.

Ensure that no one starts the SAP Sybase IQ database until the system-level backup is complete.

Ensuring that the Database is Shut Down

The file protection of the .db file is read-only when the database is shut down cleanly, and set to read/write when the database is in use. If you are writing a script to perform backups, it is a good idea for the script to check the access mode of the file, to be sure that the database is shut down.

To ensure that a database remains shut down, the script can check the size of the .iqmsg file at the start and end of the script to make sure it has not changed. If the database was started while the script was running, the .iqmsg file is larger.

Back Up the Right Files

Back up required files and optional files.

Required Files

You must back up the following files:

- All SYSTEM dbspace files, typically named dbname.db

> **Note:** There may be additional dbspaces in the catalog store, and are listed in
> SYSDBSPACES.

- The transaction log file, which is required for system recovery, typically named
  dbname.log
- The IQ_SYSTEM_MAIN dbspace file, typically named dbname.iq
- Files for any additional dbspaces that have been added to the IQ main store.

Save the lengths of the following files:

- The IQ_SYSTEM_TEMP dbspace file, typically named dbname.iqtmp
- Additional files that have been added to IQ_SYSTEM_TEMP

Backing up the temporary dbspaces is not required. IQ can reconstruct any temporary dbspace provided that it sees a file of the correct length at the time the database starts. Therefore, you may simply keep records of the sizes of the files or raw devices used to hold the temporary dbspaces.

Optional Files

You should back up the ASCII message files such as dbname.iqmsg and the $IQDIR16/logfiles/*.srvlog and $IQDIR15/logfiles/*.stderr files, even though these files are not required for a restore. If problems occur during a restore, the .iqmsg file contains information that proves that the database was shut down before the backup started.

These files may be useful in diagnosing the cause of the database failure you are recovering from. Be sure to make a copy before restoring, for use in later analysis.

If IQ message log wrapping is enabled, you will probably want to back up the .iqmsg file so that all messages are accessible in the event you need them for diagnostic purposes.

If message log archiving is enabled (the **IQMsgMaxSize** server option or the **-iqmsgsz** server startup switch is not equal to zero and the **IQMsgNumFiles** server option or the **-iqmsgnum** server start up switch is not equal to zero), the server automatically backs up the message log archives. The maximum amount of message log that is archived is 128GB, which is sufficient in most cases.

> **Note:** Backing up the message log archives is required before a server restart. After the server restarts, the existing log archives are ignored and a new archive is created when the dbname.iqmsg file is full. To preserve the old archive logs, back up the files before restarting the server.

Keeping Your Backup List Updated

It is critical to add to your system backup specification any dbspaces that are added to the database, whether they are in SYSTEM, IQ_SYSTEM_MAIN, or IQ_SYSTEM_TEMP. If a dbspace is added several months down the road, or after some turnover in your organization, you may miss this step.

To ensure that you are backing up all the files you need, use a script for system-level backups. In the script, before starting the backup, compare a select from *SYSFILE* (for the system

dbspaces) and from *SYSIQFILE* (for the IQ dbspaces) to a list of dbspaces known to be in the system backup specification.

Raw Devices and Symbolic Links

If your database files are on raw devices, be sure your system backup is backing up the raw device contents, not just the name of the device in `/dev/*`.

If symbolic links are used for raw device names, as recommended, be sure the system backup utility follows the symbolic link and backs up the device.

Restoring from a System-Level Backup

If you must restore from a system-level backup, you must ensure that database server is shut down, just as it was during the backup.

When restoring a multiplex database, you must shut down all the secondary servers as well as the write server.

Ensuring that All Files Exist

Before restoring, review the table of contents of the backup to ensure that all files required for IQ are present. The list of files depends on your application.

In the case of the temporary dbspace files, ensure that files or raw devices are present with the correct file names (or symbolic links) and lengths. Contents of temporary dbspace files are irrelevant until the database restarts.

Checking Ownership and Permissions

Ensure that ownership and permission levels do not change during the system-level restore.

Validating Your Database

Backing up a database is useful only if the database is internally consistent.

Backup always makes sure that the database is in a usable state before proceeding. However, validating a database before you perform a backup is a good idea, to ensure that the database you restore is stable. The restore program does not check for inconsistencies in the restored data, since the database may not even exist.

To validate your database, issue the following command:

```
sp_iqcheckdb 'check database'
```

The **sp_iqcheckdb** stored procedure, in conjunction with server startup switches, is the interface to the IQ Database Consistency Checker (DBCC).

DBCC has different verification modes that perform increasing amounts of consistency checking. There are three modes for checking database consistency and one for resetting allocation maps. Each mode checks all database objects, if you specify 'database' as the target

in the **sp_iqcheckdb** command string. Individual tables and indexes can also be specified in the command string. If you specify individual table names, all indexes within those tables are also checked.

The database option **DBCC_LOG_PROGRESS** instructs **sp_iqcheckdb** to send progress messages to the IQ message file. These messages allow you to follow the progress of the **sp_iqcheckdb** procedure as it executes.

You should run **sp_iqcheckdb** before or after backup, and whenever you suspect a problem with the database.

Validating a Multiplex Database
In an IQ multiplex, run **sp_iqcheckdb** only on a write server.

Concurrency Issues for sp_iqcheckdb
When you run **sp_iqcheckdb** on an entire database, **sp_iqcheckdb** reads every database page in use.

This procedure consumes most of the database server's time, so that the I/O is as efficient as possible. Any other concurrent activities on the system run more slowly than usual. The CPU utilization of DBCC can be limited by specifying the **sp_iqcheckdb** parameter **resources resource-percent**, which controls the number of threads with respect to the number of CPUs.

If other users are active when you run **sp_iqcheckdb**, the results you see reflect only what your transaction sees.

Restoring Your Databases

Once you have created a database and made a full backup, you can restore the database when necessary. SAP Sybase IQ restores the database to its state as of the automatic **CHECKPOINT** at the start of the backup.

Before You Restore

Conditions must be met before you can restore a database.

- You must have DBA privileges.
- 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 path name 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 or read-only files in a read-write dbspace except for IQ_SYSTEM_MAIN, the dbspace must be offline. This restriction does not

---

apply to IQ_SYSTEM_MAIN. Selective restore can be used to restore a read-only dbspace, as long as the dbspace is still in the same read-only state.

- To restore all files in a database or from a read-write files only backup, you must be connected to the `utility_db` database. For information on `utility_db` and how to set privileges for using it, see the *Installation and Configuration Guide* for your platform.
- When restoring all the files in the database or from a read-write files only backup, no user can be connected to the database being restored. **RESTORE** exits with an error if there are any active Read Only or Read/Write users of the specified database.

  Use two startup switches to restrict connections:

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

  An alternate way to restrict connections is to specify

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

  just after you start the connection where you are restoring and

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

  on the same connection after restoring. The disadvantage is that this method precludes emergency access from another DBA connection.
- You must restore the database to the appropriate server, and that server must have the archive devices you need. When you use the Sybase-provided restore, you need the same number of archive devices (that is, the disk files or tape drives) as when the backup was created.
- For a full restore, the store files (by default the `.iq` files), the catalog store (by default the `.db` file), and the transaction log (by default the `.log` file) must not exist in the location to which you are restoring. If any of these files exist, you must delete them or move them to a different directory before doing the full restore.

  When a full restore begins, it destroys all old database files and then recreates them. The requirement that you manually delete the store, catalog store, and transaction log files protects you from doing a full restore accidentally.
- For any incremental restore, the catalog store (`.db`) must exist. If it exists, but in a different location than the one you are restoring to, move database files. If it does not exist, you can only do a full restore. (If you do a full restore before any incremental restore, the correct files will be in place.)
- For any incremental restore, the database must not have changed since the last restore.

Restore requires exclusive access to the database and to the server. To gives the DBA greater control over inadvertent opens of the database, start the database server with the **-gd DBA** option set, but do not start the database you are restoring. **RESTORE** automatically starts the database in such a way that no other users can connect to it.

You must restore an entire backup or set of backups, including the full backup and all subsequent incremental backups. Restoring individual files from a backup archive is supported for read-only dbspaces and files alone. However, you can move database files to a new location using the **RENAME** clause of the **RESTORE** command.

Before you restore the database, you can verify the database backup (an existing SAP Sybase IQ version 12.7 or later database backup) using the **VERIFY** clause of the **RESTORE** SQL statement.

Restore Accommodates Dbspace Changes

During a set of incremental restores, **RESTORE** creates and drops dbspaces as needed to match what was done during the period of operation encompassed by the restores.

For example, assume that you make a full backup of a database, then add a dbspace to that database, and then do an incremental backup after adding the dbspace. When you restore from these backups, **RESTORE** creates a file for the new dbspace, at the start of the incremental restore. Similarly, if you drop a dbspace, it is dropped during the restore, although the actual file is not removed.

Note that 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.

Restoring Disk Backup Files

If you back up to disk and then move those files to tape, you must move them back to disk files with the same names as when you created the backup. SAP Sybase IQ cannot restore disk files that are moved to tape directly from tape.

When you restore using the SAP-provided backup and restore, you must specify the same number of archive devices (disk files) for the restore as were used to create the backup.

Restoring Tape Backup Files

When restoring from tape, you must position the tape to the start of the IQ data. **RESTORE** does not reposition the tape for you.

When you restore using the SAP-provided backup and restore, you must use the same number of tape drives for the restore as were used to create the backup(s) you are restoring.

Specifying Files for an Incremental Restore

For an incremental restore, files you restore must match in number and size the files they replace, for both the IQ and catalog stores.

Keeping the Database Unchanged Between Restores

If a user changes the database before you complete a set of incremental restore operations, you cannot restore the remaining incrementals.

For example, if you have a set consisting of a full restore and two incrementals, and a user's write transaction commits after the full restore but before you issue the second or third **RESTORE** command, you cannot proceed with the incremental restores. Instead, you must restore the full backup and apply the incrementals again.

If the database has changed since the last restore and you try to do an incremental restore, the following error occurs:

```
Database has changed since the last restore
```

Note: SAP Sybase IQ does not let you do an incremental restore if the database has changed since the previous restore. However, it does not prevent users from making changes. It is the responsibility of the DBA or system operator to ensure that no changes are made to the database until all restores are complete.

Restoring from a Compatible Backup

RESTORE lets you restore database files for SAP Sybase IQ 15.0 and later. Due to changes in the format of the database, you cannot restore from a backup created on a 12.x version.

To move your data from an SAP Sybase IQ 12.x database to SAP Sybase IQ 16.0:

1. Upgrade to 12.6 ESD #11 or later using the migration procedure described in the version 12.6 *Installation and Configuration Guide*.
2. Follow the migration procedure described in the current version *Installation and Configuration Guide*.

RESTORE does not let you restore an SAP Sybase IQ backup to a SQL Anywhere database.

The RESTORE Statement

To restore a database, use the **RESTORE** statement.

You must be connected to the utility_db database as DBA to issue this statement.

You must specify the *db_file* and at least one *archive_device*.

For *db_file* you specify the location of the catalog store file for the database (created with the suffix .db by default). You can specify the full path name or a path name relative to the directory where the database was created. If you specify a new path name, the catalog store and any files created relative to it are moved to that location, except for any files you include in a **RENAME** clause.

Just as for backup, each *archive_device* specifies the API (third party) and, for the Sybase API, the physical tape device or disk file name from which you are restoring. For third-party APIs, the content of the *archive_device* string depends on your vendor. The archive device must not

be a raw disk device. When you restore from disk files using the Sybase API, you must supply the same number of archive devices as were specified when this backup was created.

Warning! If you misspell a tape device name and give a name that is not a valid tape device on your system, **RESTORE** assumes it is a disk file and tries to read from it.

Note: If you are restoring from tape devices on Windows, note that you do not need to redouble the backslashes when you specify tape devices for restore, as you did for backup.

Example 1 — Restoring to the Same Location
This Windows example restores a database to iquser.db. The database is restored from two disk files. All database files are restored to their original locations.
```
RESTORE DATABASE 'iquser.db'
FROM 'c:\\iq\\backup1'
FROM 'c:\\iq\\backup2'
```

Moving Database Files
Move database files to a new location.

If you need to move database files to a new location—for example, if one of your disk drives fails—you use one of the following methods:

- To move the database file that holds the catalog store (by default, the .db file), you simply specify the new name as *db_file*.
- To move or rename the transaction log file, use the dblog transaction log utility. For syntax and details, see the *Utility Guide > dblog Database Administration Utility*.
- To move any other database file, you use the **RENAME** option.

Restoring to a Raw Device
When restoring to a raw device, make sure that the device is large enough to hold the dbspace being restored. IQ **RESTORE** checks the raw device size and returns an error, if the raw device is not large enough to restore the dbspace.

The operating system takes a small amount of space on the raw device and the IQ dbspace occupies the rest. When you restore the dbspace, your raw partition must hold both the IQ dbspace and the space reserved for the operating system.

To restore an IQ main or temporary dbspace to a raw partition, find the raw device size needed for each IQ dbspace from system tables as follows:
```
SELECT segment_type, file_name, block_count,
data_offset, block_size,
(block_count * block_size) + data_offset AS raw_size
FROM SYS.SYSIQFILE, SYS.SYSIQINFO
where segment_type !='Msg' ORDER BY 1,2
```

The segment_type and file_name are informational. Segments of type 'Main' or 'Temp' may be stored on a raw partition, but message files (type 'Msg') may not. The file_name is the name of the dbspace.

The `block_count` column is an integer, the number of blocks used by IQ.

The `data_offset` column is an integer, the number of bytes reserved for the operating system.

The `block_size` column is an integer, the number of bytes per IQ block.

The `raw_size` column is an integer, the minimum size in bytes of a raw device needed to restore this dbspace. You should restore to a raw device that is at least 10MB larger than the original raw device.

Example 2 — Moving the Catalog Store

This example restores the same database as Example 1.

In Example 2, however, you move the catalog store file and any database files that were created relative to it. To do so, you replace the original file name with its new location, `c:\newdir`, as follows:

```
RESTORE DATABASE 'c:\\newdir\\iqnew.db'
FROM 'c:\\iq\\backup1'
FROM 'c:\\iq\\backup2'
```

SAP Sybase IQ moves database files other than the catalog store as follows:

- If you specify a **RENAME** clause, the file is moved to that location.
- If you do not specify a **RENAME** clause, and the file was created using a relative path name, it is restored relative to the new location of the database file. In other words, files originally created relative to the `SYSTEM` dbspace, which holds the catalog store file, are restored relative to the catalog store file. Files originally created relative to the catalog store are restored relative to the catalog store.
- If you do not specify a **RENAME** clause, and the file was created using an absolute path name, the file is restored to its original location.

In other words, if you want to move an entire database, you should specify in a **RENAME** clause the new location for every IQ dbspace in the database—required, temporary, and user-defined. The SYSTEM dbspace is the only one you do not include in a **RENAME** clause.

If you only want to move some of the files, and overwriting the original files is not a problem, then you only need to rename the files you actually want to move.

You specify each dbfile *file_name* as it appears in the SYSIQDBFILE table. You specify *new_dbspace_path* as the new raw partition, or the new full or relative path name, for that dbspace.

You cannot use the **RENAME** option to specify a partial restore.

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 catalog store.

If you are renaming files while restoring both full and incremental backups, be sure you use the dbspace names and paths consistently throughout the set of restores. It is the safest way to ensure that files are renamed correctly.

If a dbspace was added between the full backup and an incremental backup, and you are renaming database files, you need one more **RENAME** clause for the incremental restore than for the full restore. Similarly, if a dbspace was deleted between backups, you need one fewer **RENAME** clause for the restores from any backups that occurred after the dbspace was deleted.

Example 3 — Moving a User dbspace

This example shows how you restore the full and incremental backups in the example shown earlier in this chapter. In this case, media failure has made a UNIX raw partition unusable. The user-defined dbfile on that raw partition, `IQ_USER`, must be moved to a new raw partition, /dev/rdsk/c1t5d2s1. No other database files are affected.

First, you connect to the `utility_db` database. Then you restore the full backup from two tape devices. In this case they are the same two tape devices used to make the backup, but the devices could differ as long as you use the same number of archive devices, the same media type (tape or disk), and the same tape sets in the correct order.

The first **RESTORE** command is:

```
RESTORE DATABASE 'iquser'
FROM '/dev/rmt/0n'
FROM '/dev/rmt/1n'
RENAME IQ_SYSTEM_MAIN TO '/dev/rdsk/c2t0d1s1'
RENAME IQ_SYSTEM_TEMP TO '/dev/rdsk/c2t1d1s1'
RENAME IQ_SYSTEM_MSG TO 'iquser.iqmsg'
RENAME IQ_USER TO '/dev/rdsk/c1t5d2s1'
```

The second **RESTORE** command, to restore the incremental backup, is:

```
RESTORE DATABASE 'iquser'
FROM '/dev/rmt/0n'
RENAME IQ_SYSTEM_MAIN TO '/dev/rdsk/c2t0d1s1'
RENAME IQ_SYSTEM_TEMP TO '/dev/rdsk/c2t1d1s1'
RENAME IQ_SYSTEM_MSG TO 'iquser.iqmsg'
RENAME IQ_USER TO '/dev/rdsk/c1t5d2s1'
```

Note: You could also issue these commands with only the last **RENAME** clause, since only one dbspace is being restored to a new location. Listing all of the files or raw partitions, as shown here, ensures that you know exactly where each will be restored.

Displaying Header Information with CATALOG ONLY Option

The **CATALOG ONLY** option displays the header information for the database, placing it in the `.backup.syb` file.

It does not restore any data, either from the catalog store or the IQ store.

When you specify **CATALOG ONLY** you must include the **FROM archive_device** clause, but omit the **RENAME** clause.

```
RESTORE DATABASE 'iqdemo.db' FROM '/disk1/users/jones/backup/
iqdemo' CATALOG ONLY
```

generates the header:

```
RESTORE, -1988637423.0, bigendian_420111.db,
        ASIQ, '2013-01-11
       06:57:00.000', DBA, Full, Arch, bigendian_420111_backup, ''
BACKUP, 453495113.0, iqdemo.db, ASIQ, '2013-01-23 10:33:00.000',
DBA,
        Full, Arch, /disk1/users/jones/backupiqdemo, ''
RESTORE, 453496081.0, ,  ASIQ , '2013-01-23 10:33:00.000', DBA,
        Full, Arch, /disk1/users/jones/backupiqdemoo,
      ''
```

Adjusting Data Sources and Configuration Files

When you move a database, you may need to modify your data sources, configuration files, and integrated logins to reflect the new location of the database.

Restoring in the Correct Order

When you restore from a full backup, every block in use at the time the backup was made is written to disk. When you restore from an incremental backup, only the blocks that changed between the previous backup (or the previous full backup) and this backup are written to disk.

You must restore full and incremental backups in the correct order, with a separate **RESTORE** command for each backup you are restoring. **RESTORE** ensures that backups are restored in order, and gives the following error if it determines that the order is incorrect:

```
SQL Code:  -1012009
```

```
SQL State: QUA09
```

```
This restore cannot immediately follow the previous restore.
```

To determine the correct order, you need the information about backup files that is stored in the backup log.

Restore backups as follows:

- If your database is inconsistent, or if you are moving any files to a new location, you must restore a **FULL** backup.
- If your most recent backup is a **FULL** backup, or if you need to restore a database to the state that existed before any existing incremental(s) were made, restore the full backup only.
- If you have an **INCREMENTAL_SINCE_FULL** backup that precedes the database failure, first restore from the last **FULL** backup, and then restore the **INCREMENTAL_SINCE_FULL** backup.
- If you do not have an **INCREMENTAL_SINCE_FULL** backup, but you have performed one or more **INCREMENTAL** backups since your last **FULL** backup, first restore the **FULL**

backup, and then restore the **INCREMENTAL** backups in the order in which they were made.

You can also use the advisory stored procedure **sp_iqrestoreaction** to suggest the sequence of restore actions required to attain a stable database set. Always confirm the steps suggested against above rules. The stored procedure also does not factor in moving any database files.

Within a given backup, the order in which you restore tapes is also important. In particular, you need to keep track of the order of tapes in each backup tape set, that is, the set of tapes produced in a given backup on a given archive device:

- You must restore the tape set that contains the backup of the catalog store first, and it must be on the first archive device.
- Within each set, you must restore tapes in the order in which they were created.
- You cannot interleave sets; each set must be restored before you can restore another set.
- After the first set, the order in which sets are restored does not matter, as long as it is correct within each set.

Use the same number of drives to restore as were used to produce the backup, so that you do not accidentally interleave tapes from different sets.

Example
Assume that you are restoring a full backup, in which you used three archive devices, and thus produced three tape sets, A, B, and C.

The contents of each set, and the restore order, are as follows:

- **Set A** – Tapes A_1, A_2, and A_3. Tapes A_1 and A_2 contain the catalog store. This set must be restored first, and must be in the first device.
- **Set B** – Tapes B_1 and B_2. These must be restored as a set, after Set A, and either before or after Set C. They can be in either the second or third device.
- **Set C** – Tapes C_1, C_2, and C_3. These must be restored as a set, after Set A, and either before or after Set B. They can be in either the second or third device.

The Restore program checks that tapes within each set are in the correct order on a single device. If not, you get an error, and the restore does not proceed until you supply the correct tape. Except for the set with the catalog store, it does not matter which set you put on a given device.

Note: You must ensure that the catalog store tape set is restored first. The Restore program does not check this.

Although these rules also apply to disk files, you are not likely to back up to multiple files on a given disk device.

Reconnecting After You Restore

SAP Sybase IQ requires the DBF parameter and database file name in order to connect to a database under certain circumstances.

This situation occurs when you use Interactive SQL and you have restored that database from backup while connected to utility_db.

For example, include the DBF parameter as follows:

```
CONNECT USING 'uid=DBA;pwd=sql;dbf=node1/users/fiona/mydb.db;
links=tcpip{host=serv1;port=1234};eng=serv1_iqdemo'
```

The following syntax returns a specified database not found error:

```
CONNECT DATABASE mydb USER DBA IDENTIFIED BY SQL
```

To avoid the error, enter a **START DATABASE** command while connected to utility_db, for example:

```
START DATABASE mydb
```

Use this method when connecting via Interactive SQL.

Renaming the Transaction Log after you Restore

When you rename or move all other files in the database, you should also do the same for the log file.

To move or rename the log file, use the Transaction Log utility (**dblog**). You should run this utility:

- After using **RESTORE** with a new database name
- After using **RESTORE** with the **RENAME** option

Note: The database server must not be running on that database when the transaction log file name is changed.

You can also use **dblog** to rename the transaction log, even if you have not restored the database, given certain restrictions. You can access the Transaction Log utility from the system command line, using the **dblog** command-line utility. See *Utility Guide > dblog Database Administration Utility*.

Validating the Database After You Restore

To ensure that tapes have been restored in the correct order, you should run the stored procedure **sp_iqcheckdb** after you finish restoring your database.

If you are restoring a set of incremental backups, it is safest to run **sp_iqcheckdb** after restoring each backup. To save time, however, you may prefer to run **sp_iqcheckdb** only after restoring the last incremental.

Restore Requires Exclusive Write Access

Once **RESTORE** starts, no other users are allowed to access the specified database.

If you restore from a full backup and then from one or more incremental backups, you should ensure that no users are modifying the database between the restores. The modifications are permitted, but you cannot perform any more incremental restores. Instead, you must start the entire restore again.

This restriction extends to any incremental restores you may need if your system crashes during recovery. If you need to recover from a system or media failure that occurs during a restore, you must do one of the following:

• Continue the original sequence of full and incremental restore operations, or
• Perform a full restore, followed by any incremental restores needed to fully recover your database.

The default database server startup setting **-gd DBA** makes the SERVER OPERATOR system privilege a requirement for starting up a database. When a user with the SERVER OPERATOR system privilege runs **RESTORE**, the command automatically starts the database, gets the information it needs for the restore, and then stops the database. At the end of the restore, the command starts the database, issues a checkpoint, and stops it again. This procedure ensures that the DBA has exclusive write access throughout a restore.

When all incremental restores are complete, a user with the SERVER OPERATOR system privilege issues the **START DATABASE** command again to allow other users access to the database.

To restore a multiplex database, see *Administration: Multiplex*.

Displaying Header Information

You can display the contents of the header file by using the **RESTORE** statement with the **CATALOG ONLY** option and no **FILE** clauses.

A **RESTORE** with **CATALOG ONLY** produces the information in the same format as the backup log entry for an actual **RESTORE**.

You can get more information about the backup archive using the command-line utility **db_backupheader**, which accepts the file path corresponding to the first backup archive. The utility reads the backup archive file. It does not connect to the database.

The backup archive information includes:

• backup information
• database information at the time of the backup
• dbspace information for each dbspace in the database
• dbfile information for each dbfile in the dbspaces

Recovery from Errors During Restore

If an incremental restore fails early in the operation, the database is still usable (assuming it existed and was consistent before the restore).

If a full restore fails, you do not have a usable database.

If a failure occurs after a certain point in the operation, the restore program marks the database as inconsistent. In this case recovery is only possible by means of a **FULL RESTORE**. If you were performing a **FULL RESTORE** when the failure occurred, you may need to go back to the previous **FULL BACKUP**.

Verifying a Database Backup

SAP Sybase IQ includes a mechanism that verifies an existing SAP Sybase IQ version 12.7 or later database backup using the **VERIFY** clause of the **RESTORE** SQL statement.

The verification process directs the server to validate the specified SAP Sybase IQ database backup archives for a full, incremental, incremental since full, or virtual backup, and to check the specified archive for the same errors a restore process checks, but performs no write operations. All status and error messages are written to the server log file.

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

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.

Verification of an Incremental Backup
Use the **RESTORE… VERIFY COMPATIBLE** clause to check the compatibility of an incremental archive with the existing database files. If the database files do not exist on the system on which you invoke **RESTORE…VERIFY COMPATIBLE**, an error is returned. If you specify **COMPATIBLE** while verifying a full backup, the keyword is ignored; no compatibility checks need to be made while restoring a full backup.

If you specify **RESTORE VERIFY** without **COMPATIBLE** for an incremental restore, SAP Sybase IQ does not look for any dbspaces and does not perform any compatibility checks. No warning is reported, even if the files do not exist. The compatibility check is performed only when you include the **COMPATIBLE** clause.

If you specify **RESTORE VERIFY COMPATIBLE** for an incremental restore, and the IQ catalog store or any of the SAP Sybase IQ dbspaces do not exist, the compatibility check cannot be made; an error is reported and the operation fails.

During validation of an incremental backup, the **RESTORE VERIFY COMPATIBLE** process opens the SAP Sybase IQ dbspaces in read-only mode to perform consistency checking. No dbspaces are modified.

In incremental restores, if the database has been modified or the particular incremental archive is not the correct archive for the database, **RESTORE VERIFY COMPATIBLE** reports the error `Database has changed since last restore.` (SQLCODE -1012008, SQLSTATE QUA08) or `This restore cannot immediately follow the previous restore.` (SQLCODE -1012009, SQLSTATE QUA09).

Verification Progress Reporting

The **RESTORE VERIFY** process verifies every stripe specified in the command. As the verification process checks stripes and their corresponding files, it reports progress in terms of the number of IQ blocks verified. After every 5000 IQ blocks verified, a message is written to the server log file:

```
5000/100000 (5%) Blocks verified
```

A final message is written to the server log file when 100% of the IQ blocks are verified.

The messages `RESTORE VERIFY Started`, the number of IQ blocks to be verified, and `RESTORE VERIFY successfully completed` are also written in the server log, when the verify action starts and completes, respectively. For example:

```
I. 11/17 06:45:24. VERIFY RESTORE Started
I. 11/17 06:45:24. Total number of IQ blocks to be verified: 764
I. 11/17 06:45:24. Total number of IQ blocks verified: 764/764 ( 100
% )
I. 11/17 06:45:24. VERIFY RESTORE Successfully Complete
```

Verification Error Reporting

If the verification process finds errors after which it can continue, the process continues checking the archive and logs information for the errors detected.

The errors for which the verification can continue are:

- `Header of block to be restored appears to be corrupted. (SQLCODE -1012011l, SQLSTATE QUA11)`
- `Media data appears corrupted (bad checksum). (SQLCODE -1012012, SQLSTATE QUA12)`
- `Media meta data appears corrupted (boundary record). (SQLCODE -1012013, SQLSTATE QUA13)`
- `Media meta data appear corrupted (multiple begin boundary records). (SQLCODE -1012014, SQLSTATE QUA14`

If any of these errors are found and the verification process can continue to the end of the archive, SAP Sybase IQ reports this error:

```
The verification of the provided archive has failed. Please
check the server log for details of the errors thrown during
verify.
```

If any error pertaining to **RESTORE** is found other than the errors above, the error that occurred is reported, and the verification process stops.

See also
• *RESTORE DATABASE Statement* on page 117

Getting Information about Backups and Restores

SAP Sybase IQ provides a backup log, `.backup.syb`, to help you manage your backup media.

This log is not used to create the backup or to restore the database; however, information describing the backup or restore is recorded in this file during both Backup and Restore.

Note: To display only the information about a particular backup, you can run **RESTORE** with the **CATALOG ONLY** option. This option displays the header file for a backup from the media rather than from the file, so that the DBA can identify what is on the tape or file.

Locating the Backup Log

The `.backup.syb` file is in ASCII text format.

Its location depends on the setting of environment variables at the time the server is started:

• On UNIX, the server tries to place it in the following locations, in this order:
    • The directory specified by the IQLOGDIR16 environment variable.
    • The directory specified by the HOME environment variable.
    • The home directory as obtained from account information.
    • The current directory (where the server was started).
    If the file is placed in the home directory, it is prefixed with a "." in order to make it a hidden file. If the file is placed in the current directory, it is not prefixed.
• On Windows, the server tries to place it in the following locations, in this order:
    • The directory specified by the IQLOGDIR16 environment variable.
    • The directory that holds the server executable files.

Content of the Backup Log

For every backup or restore you perform, the backup log contains a comma-separated list of fields.

The backup log provides:

- Operation (Backup or Restore)
- Version
- Database name
- Database type (SAP Sybase IQ or SQL Anywhere)
- Date and time of backup or restore
- Creator user ID
- Type of backup/restore: Full, Incremental, or Incremental_since_full, or Database File Only (for SQL Anywhere databases only)
- Method: Archive (for SAP Sybase IQ or SQL Anywhere databases) or Image (SQL Anywhere databases only)
- Location
- Comment (if entered on the **BACKUP** command), enclosed in single quotes. If the comment includes quotes, they appear as two consecutive single quotes.

Here is a sample backup log.

```
BACKUP, 2.0, all_types.db, ASIQ, '2009-01-31 16:25:00.000', DBA,
Full, Arch, TED_FULL00, ''
```

```
BACKUP, 2.0, all_types.db, ASIQ, '2009-01-31 16:53:00.000', DBA,
Incr, Arch, TED_X_bkup_inc, ''
```

```
RESTORE, 2.0, all_types.db, ASIQ, '2009-01-31 16:25:00.000', DBA,
Full, Arch, TED_FULL00, ''
```

```
RESTORE, 2.0, all_types.db, ASIQ, '2009-01-31 16:53:00.000', DBA,
Incr, Arch, TED_X_bkup_inc, ''
```

```
BACKUP, 2.0, all_types.db, ASIQ, '2009-01-31 20:07:00.000', DBA,
InSF, Arch, A_partial2_yes_sf, ''
```

```
BACKUP, 2.0, all_types.db, ASIQ, '2009-01-31 20:07:00.000', DBA,
InSF, Arch, A_partial2_yes_sf, ''
```

Maintaining the Backup Log

It's a good idea to clean up the backup log after you purge backup media. Use a text editor to do so.

Be careful with your edits: once **BACKUP** or **RESTORE** records information in this file, it does not check its accuracy.

There is only one backup log on a server. The server must be able to read and write this file. The system administrator may want to limit access to this file by other users. If you are running more than one database server on a system, you should set the IQLOGDIR15 environment variable differently for each server, to produce separate backup logs.

Warning! Do not edit the backup log while a backup or restore is taking place. If you are modifying the file while **BACKUP** or **RESTORE** is writing to it, you may invalidate the information in the file.

Recording Dbspace Names

In the event that you ever need to use the **RENAME** option of **RESTORE** to move a database or one of its dbspaces, you need to know the name of every dbspace in the database.

The dbspace names are in the SYSFILE table of every database, but you do not have this table available when you are restoring. Run **db_backupheader** on the first backup archive file path to view this information. Alternatively, you may execute the **sp_iqdbspace** and **sp_iqfile** stored procedures or issue the following statement any time you back up your database:

```
SELECT dbf.dbfile_name, f.*
FROM SYSFILE f, SYSDBFILE dbf
WHERE f.file_id=dbf.dbfile_id
```

Keep the results of this query some place other than the disk where the database resides, so that you have a complete list of dbspace names if you need them.

You can also run the following script in Interactive SQL. This script produces an output file that contains the set of rename clauses you use, if you do not actually change the location of any files. You can substitute any new file locations, and use the resulting file in your **RESTORE** statement.

Note: Because the database may not exist when you need to restore, you may want to run this script after you back up your database.

```
-- Get dbspace and IQ file names and add
-- rename syntax including quotation marks

select 'rename' as 'restore ... rename' ,
dbf.dbfile_name as 'IQ file' , 'to' as 'to' ,
'''' + f.file_name + '''' as 'file_path'
from SYSFILE f, SYSDBFILE dbf
where f.store_type=2 and f.file_id=dbf.dbfile_id

 -- Send output to a file in proper format
 -- without delimiters or extra quotation marks

output to restore.tst delimited by ' ' quote '';

 -- This produces a restore.tst file like the following:
 -- rename IQ_SYSTEM_MAIN to '/dev/rdsk/c2t0d1s7'
 -- rename IQ_SYSTEM_TEMP to '/dev/rdsk/c2t1d1s7'
 -- rename IQ_SYSTEM_MSG to 'all_types.iqmsg'
```

Determining Your Data Backup and Recovery Strategy

To develop an effective strategy for backing up your system, you need to determine the best combination of full, incremental, and incremental-since-full backups for your site, and then set up a schedule for performing backups.

Consider the performance implications of various backup options, and how they affect your ability to restore quickly in the event of a database failure.

Scheduling Routine Backups

Make a full backup of each database just after you create it, to provide a base point, and perform full and incremental backups on a fixed schedule thereafter. It is especially important to back up your database after any large number of changes.

Your backup plan depends on:

- The load on your system
- The size of your database
- The number of changes made to the data
- The relative importance of faster backups and faster recovery

Determining the Type of Backup

When you decide whether to do a full, incremental, or incremental_since_full backup, you need to balance the time it takes to create the backup with the time it would take to restore.

You also should consider media requirements. A given incremental backup is relatively quick and takes a relatively small amount of space on tape or disk. Full backups are relatively slow and require a lot of space.

Incremental_since_full is somewhere in between. It starts out as equivalent to incremental, but as the database changes and the number of backups since a full backup increases, incremental_since_full can become as time-consuming and media-consuming as a full backup, or worse.

In general, the opposite is true for restore operations. For example, if you need to restore from a very old full backup and a dozen or more incrementals, the restore may take longer and the backup may use up more space than a new full backup.

The obvious advantage of incremental backups is that it is much faster and takes less space to back up only the data that has changed since the last backup, or even since the last full backup, than to back up your entire database. The disadvantage of relying too heavily on incremental backups is that any eventual restore takes longer.

For example, once you have a full backup of your database, in theory you could perform only incremental backups thereafter. You would not want to do this, however, because any future recovery would be intolerably slow, and would require more tape or disk space than doing a

full backup periodically. Remember that other users can have read and write access while you do backups, but no one else can use the database while you are restoring it. You might find yourself needing to restore dozens of incremental backups, with your system unavailable to users throughout the process.

A much better approach is a mix of incremental and full backups.

The greater the volume of your database changes, the more important it is do a backup, and the smaller the advantage of incremental backups. For example, if you update your database nightly with changes that affect 10 percent or more of the data, you may want to do an incremental_since_full backup each night, and a full backup once a week. On the other hand, if your changes tend to be few, a full backup once a month with incrementals in between might be fine.

Designating Backup and Restore Responsibilities

Backing up or restoring an SAP Sybase IQ database requires BACKUP DATABASE system privilege.

Improving Performance for Backup and Restore

The overall time it takes to complete a backup or restore a database depends largely on the strategy you choose for mixing full and incremental restores.

Several other factors also affect the speed of backup and restore operations: the number of archive devices, data verification, the memory available for the backup, and size of the IQ and catalog stores.

Increasing the Number of Archive Devices

The **TO** clause in the **BACKUP** statement controls the number of archive devices.

Eliminating Data Verification

You can also improve the speed of backup and restore operations by setting **CRC OFF** in the **BACKUP** command.

This setting deactivates cyclical redundancy checking. With **CRC ON**, numbers computed on backup are verified during any subsequent restore operation, affecting performance of both commands. The default is **CRC ON**. If you turn off this checking, remember that you are giving up a greater assurance of accurate data in exchange for faster performance.

Spooling Backup Data

You may find that it is faster and more efficient to create backups on disk, and then spool them onto tape for archival storage. If you choose this approach, you need to unspool the data onto disk before restoring it.

Increasing Memory Used During Backup

The amount of memory used for buffers during backup directly affects backup speed, primarily for tape backups. The **BLOCK FACTOR** parameter of the **BACKUP** command controls the amount of memory used. If your backups are slow, you may want to increase the value of **BLOCK FACTOR** for faster backups.

The effect of **BLOCK FACTOR** depends on your operating system, and on the block size specified when the database was created. The default IQ page size of 128KB for newly created databases results in a default block size of 8192 bytes.

Set **BLOCK FACTOR** to at least 25 (the UNIX default). With this combination, **BACKUP** is able to buffer data ideally for most UNIX tape drives, with enough data in memory that drives are kept busy constantly throughout the backup.

On Windows, the default **BLOCK FACTOR** is computed based on the database block size. This value usually achieves maximum throughput on Windows. Because of the way Windows handles tape devices, you may not be able to achieve faster backups by increasing the **BLOCK FACTOR**.

Balancing System Load

SAP Sybase IQ allows you to perform backups concurrently with all other read/write operations, except those that affect the structure of the database.

It is still a good idea to schedule backups during times of low system use, however, to make the best possible use of system resources—disk, memory, and CPU cycles.

Controlling the Size of the Catalog Store

an SAP Sybase IQ database consists of an IQ store and an underlying catalog store.

BACKUP makes a full backup of the catalog store at the start of every backup, both full and incremental. Ordinarily the catalog store is quite small, containing only the system tables, metadata, and other information SAP Sybase IQ needs to manage your database. However, it is possible to create non-IQ tables in the catalog store. You can improve IQ backup performance by keeping any non-IQ data in a separate SQL Anywhere-only database, rather than in the catalog store.

Backup copies only the latest committed version of the database. Other version pages used by open transactions are not backed up.

Archiving Data with Read-Only Hardware

Recent regulations such as the Health Insurance Portability and Accountability Act (HIPAA) and Sarbanes-Oxley Act specify rigid rules for data retention and compliance, requiring data archived in a immutable, easily accessible form.

Data volumes may extend into the several terabyte range, while data retention periods range from a few years to decades.

WORM disk storage solutions evolved to address these requirements. WORM (write once, read many) storage began as optical disk technology allowing only one permanent write of each storage location. WORM disk arrays are known as *read-only hardware* in SAP Sybase IQ. Read-only hardware functionality is provided by low-cost disk array hardware with a WORM protection layer added. The protection layer allows normal read-write use of the disks until the data is "frozen".

When data is frozen, the user specifies an indefinite or fixed retention period. The disks may be frozen at the volume or file level. Once frozen, data may not be modified, and the retention period may be extended, but never decreased.

Read-only hardware functionality is not limited to WORM disk array hardware; you may also remove write privilege from a raw device or file system file after the dbspace is altered read-only.

Using Read-Only Hardware

This example describes read-only hardware operations in a typical scenario.

Note: Read-only hardware functionality does not require WORM disk array hardware. You may remove write privilege from a raw device or file system file after the dbspace is altered read-only.

1. *Create an Archive*

   For this example, consider an IQ database consisting of a single catalog store dbspace named db.db with three main IQ dbspaces: A, B and C.

2. *Create New Dbspaces*

   You may create new dbspaces after creating the archive.

3. *Examine Archived Data*

   Suppose that you need to examine the archived database as of time t0.

4. *Update the Working Archive*

   If years have passed since time t0, you may upgrade the db.db0.working as long as ALTER DATABASE UPGRADE modifies no objects in the IQ main store.

5. *Create More Archives*

Create a new archive at time t1 as follows.

Create an Archive

For this example, consider an IQ database consisting of a single catalog store dbspace named `db.db` with three main IQ dbspaces: `A`, `B` and `C`.

1. At time `t0`, alter all three main dbspaces read-only.
2. Copy `db.db` to `db.db0`, either by shutting the database down and copying `db.db` or using **dbbackup** to make a copy while the database is still running.
3. Freeze dbspaces `A`, `B` and `C` at the hardware level. Store `db.db0` in an immutable form, perhaps by storing it in a file system file on the WORM device and freezing it.

At this point, the database has been archived as of time `t0` in an immutable form.

Create New Dbspaces

You may create new dbspaces after creating the archive.

1. Create two new main dbspaces `D` and `E`.
2. Continue using the database `db.db` as a production database.

    The database objects (tables, indexes, etc.) that existed as of time `t0` may have changed so that `db.db` does not equal `db.db0`. The database `db.db` continues to read data from dbspaces `A`, `B` and `C` as long as the tables that existed at time `t0` exist and as long as they contain some unmodified rows of data that existed as of time `t0`. Even if or when this is no longer true, `db.db` will continue to open `A`, `B` and `C` unless they are dropped from `db.db`, which is only allowed if they are empty from `db.db`'s point of view.

Examine Archived Data

Suppose that you need to examine the archived database as of time `t0`.

1. Copy the archived read-only `db.db0` to a read-write file `db.db0.working`.
2. Start `db.db0.working`. Note that as long as the server name `db.db0.working` does not conflict with the production system `db.db`, there is no need to stop the production system. `db.db0.working` will open `A`, `B`, `C`, and `D` in read-only mode. This will not interfere with `db.db`'s use of these files on UNIX, although Windows returns a sharing violation.

    Note that the catalog file `db.db0.working` is open in read-write mode.
3. Create a user `inv` for an investigator who wishes to examine the archived database.
4. Grant `inv` RESOURCE permission to create views, stored procedures, global or local temporary tables or any other structures necessary for the investigation.

    `db.db0` as well as `A`,`B` and `C` remain unchanged.

Update the Working Archive

If years have passed since time `t0`, you may upgrade the `db.db0.working` as long as **ALTER DATABASE UPGRADE** modifies no objects in the IQ main store.

1. The temporary dbspaces that existed as of time `t0` are not required to start `db.db0.working`. Use the server startup switch **-iqnotemp** to start `db.db0.working`

2. Drop and create new temporary dbspaces or use the temporary space created by the **-iqnotemp** parameter.

Create More Archives

Create a new archive at time `t1` as follows.

1. Alter dbspaces `D` and `E` read-only.

2. Copy `db.db` to `db.db1`.

3. Freeze `D` and `E`.

4. Save `db.db1` in an immutable form.

5. Create new main dbspace(s), for example, `F` and `G`.

6. Continue to use the production system `db.db`.

To use the archived databases `db.db0` or `db.db1`, or even both simultaneously, copy `db.db0` and/or `db.db1` to a working file and start a server. Create an archive and follow this procedure to create any number of archived versions of `db.db`.

System Recovery and Database Repair

Learn about normal SAP Sybase IQ server recovery, special recovery modes, how to verify database consistency, and how to repair database inconsistencies.

When you restart the database server, SAP Sybase IQ attempts to recover automatically. If the server cannot recover and restart, especially after a system failure or power outage, the database may be inconsistent.

Recovery and Repair Overview

If your SAP Sybase IQ server or database has problems restarting, use this information to diagnose database startup problems, verify the consistency of databases, and repair databases.

If you can restart the server after a failure, verify your database using the **sp_iqcheckdb** stored procedure, preferably before allowing users to connect

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

Examining the Server Log and IQ Message Log

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

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

Normal Recovery

During system recovery, any uncommitted transactions are rolled back and any disk space used for old versions (snapshots of database pages that were being used by transactions that did not commit) returns to the pool of available space.

After normal recovery, the database then contains only the most recently committed version of each permanent table, unless it is a multiplex database. A multiplex database contains all versions accessible to secondary servers.

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

Database Verification

Use **sp_iqcheckdb** for database verification.

Check the consistency of your database as soon as possible after the server restarts following an abnormal termination, such as a power failure, and before performing a backup of the database.

You can use the **sp_iqcheckdb** stored procedure to detect and repair database consistency problems.

The sp_iqcheckdb Stored Procedure

The SAP Sybase IQ Database Consistency Checker (DBCC) performs database verification. The **sp_iqcheckdb** stored procedure, in conjunction with server startup options, is the interface to DBCC.

You select the different modes of check and repair by specifying an **sp_iqcheckdb** command string. **sp_iqcheckdb** reads every database page and checks the consistency of the database, unless you specify otherwise in the command string.

Note: On a secondary server **sp_iqcheckdb** does not check the free list. It performs all other checks.

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

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

DBCC Performance
The execution time of DBCC varies according to the size of the database for an entire database check, the number of tables or indexes specified, and the size of the machine. Checking only a subset of the database, i.e., only specified tables, indexes, or index types, requires less time than checking an entire database.

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

sp_iqcheckdb Check Mode
In check mode, **sp_iqcheckdb** performs an internal consistency check on all IQ indexes and checks that each database block has been allocated correctly. All available database statistics

are reported. This mode reads all data pages and can detect all types of allocation problems and most types of index inconsistencies. Check mode should run considerably faster than verify mode for most databases.

When to run in check mode:

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

Examples of check mode:

Table 1. sp_iqcheckdb Check Mode Examples

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

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

When to run in verify mode:

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

Examples of verify mode:

Table 2. sp_iqcheckdb Verify Mode Examples

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

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

sp_iqcheckdb Allocation Mode

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

When to run in allocation mode:

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

Examples of allocation mode:

Table 3. sp_iqcheckdb Allocation Mode Examples

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

If some partitions of the table are offline, you can specify a partition target to check only part of a table.

You can combine all modes and run multiple checks on a database in a single session. In the following example, **sp_iqcheckdb** performs 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'
```

Allocation mode options are only allowed with the DBCC command 'allocation database'.

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

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

The **resetclocks** option corrects the values of internal database versioning clocks, in the event that these clocks are slow. Do not use the **resetclocks** option for any other purpose unless you contact Technical Support.

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

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

sp_iqcheckdb 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.

In the following example, the first statement resets allocation maps for the entire database, and the second statement resets allocation maps for the dbspace dbsp1.

```
sp_iqcheckdb 'dropleaks database'
sp_iqcheckdb 'dropleaks dbspace dbsp1'
```

Note: Use **sp_iqrebuildindex** to repair index errors.

See also
- *sp_iqcheckdb Procedure* on page 123

sp_iqcheckdb Output

The output of **sp_iqcheckdb** consists of an extensive list of statistics and any errors reported by DBCC.

Only non-zero values are displayed. Lines containing errors are flagged with asterisks (*****). Note that if you encounter errors, some of the statistics reported by DBCC may be inaccurate.

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

```
sp_iqcheckdb ># file_name
```

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

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

The following is sample progress log output of the command sp_iqcheckdb 'check database'

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

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

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

The following DBCC output indicates a Future Version error:

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

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

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

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

Stat	Value	Flags
======================	=================================	=====
DBCC Verify Mode Report		
======================	=================================	=====
DBCC Status	No Errors Detected	
DBCC Work units		
Dispatched	75	
DBCC Work units		
Completed	75	

```
=====================|================================|=====
Index Summary        |                                |
=====================|================================|=====
   Verified Index Count |86                           |
=====================|================================|=====
Allocation Summary   |                                |
=====================|================================|=====
   Blocks Total      |8192                            |
   Blocks in Current |                                |
   Version           |4855                            |
 |
   Blocks in All     |                                |
   Versions          |4855                            |
   Blocks in Use     |4855                            |
   % Blocks in Use   |59                              |
=====================|================================|=====
Allocation Statistics |                               |
=====================|================================|=====
   DB Extent Count   |1                               |
   Blocks Created in |                                |
   Current TXN       |211                             |
   Blocks To Drop in |                                |
   Current TXN       |212                             |
   Marked Logical    |                                |
   Blocks            |8240                            |
   Marked Physical   |                                |
   Blocks            |4855                            |
   Marked Pages      |515                             |
   Blocks in Freelist |126422                         |
   Imaginary Blocks  |121567                          |
   Highest PBN in Use |5473                           |
   Total Free Blocks |3337                            |
   Usable Free Blocks |3223                           |
   % Total Space     |                                |
        Fragmented   |1                               |
   % Free Space      |                                |
        Fragmented   |3                               |
   Max Blocks Per Page |16                            |
   1  Block Page Count |104                           |
   3  Block Page Count |153                           |
   ...               |                                |
   16 Block Hole Count |199                           |
=====================|================================|=====
Index Statistics     |                                |
=====================|================================|=====
   ...
   Verified Index    |fin_data.DBA.ASIQ_IDX_T209_C3_HG |
   Verified Index    |fin_data.DBA.ASIQ_IDX_T209_C4_FP |
   Verified Index    |product.DBA.ASIQ_IDX_T210_C1_FP  |
   ...
   Verified Index    |employee.DBA.ASIQ_IDX_T212_C20_FP |
   Verified Index    |iq_dummy.DBA.ASIQ_IDX_T213_C1_FP |
   FP Indexes Checked |68                             |
   HNG Indexes Checked |1                             |
   HG Indexes Checked |17                             |
```

```
======================|=================================|=====
    ...
```

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

Resource Issues Running sp_iqcheckdb

sp_iqcheckdb reports resource issues encountered while executing.

Messages describing resource issues are reported in the **sp_iqcheckdb** output or in the .iqmsg file.

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

Buffer cache sizes are set permanently using the database option TEMP_CACHE_MEMORY_MB. Use the server startup switches **-iqmc** and **-iqtc** to override the buffer cache size values set using the database options.

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

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

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

The database option **DBCC_PINNABLE_CACHE_PERCENT** can be used to tune DBCC buffer usage. The default of **DBCC_PINNABLE_CACHE_PERCENT** is to use 50% of cache. See *Reference: Statements and Options*.

Database Repair

Allocation problems can be repaired by running **sp_iqcheckdb** in **dropleaks** mode.

If DBCC detects index inconsistencies while attempting allocation repair, an error is generated and allocation problems are not fixed.

Analysis of Index Errors

Use **sp_iqcheckdb** to analyze index inconsistencies.

Sample of Output with Inconsistent Index

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

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

In this example, DBCC reports an inconsistent HNG index. Because the corresponding FP index checks are good, the FP index can be used with **sp_iqrebuildindex** to repair the damaged HNG index.

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

```
      Stat                         Value                      Flags
=====================|==============================|=====
DBCC Verify Mode Report |                           |
=====================|==============================|=====
** DBCC Status        |Errors Detected              |*****
   DBCC Work units    |                             |
   Dispatched         |75                           |
   DBCC Work units    |                             |
   Completed          |75                           |
=====================|==============================|=====
Index Summary         |                             |
=====================|==============================|=====
** Inconsistent Index |                             |
   Count              |1                            |*****
   Verified Index     |                             |
   Count              |85                           |
=====================|==============================|=====
Index Statistics      |                             |
=====================|==============================|=====
** Inconsistent Index |contact.DBA.idx01_HNG        |*****
   ...
   Verified Index     |fin_data.DBA.ASIQ_IDX_T209_C3_HG  |
   Verified Index     |fin_data.DBA.ASIQ_IDX_T209_C4_FP  |
   ...
   Verified Index     |employee.DBA.ASIQ_IDX_T212_C19_FP |
   Verified Index     |employee.DBA.ASIQ_IDX_T212_C20_FP |
```

```
   Verified Index        |iq_dummy.DBA.ASIQ_IDX_T213_C1_FP  |
** Extra Index RIDs       |5                                 |*****
   FP Indexes Checked     |68                                |
   HNG Indexes Checked    |1                                 |
   HG Indexes Checked     |17                                |
                          |                                  |
```

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

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

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

DBCC Index Errors
The DBCC output contains messages related to problems with indexes.

Table 4. DBCC Index Errors

DBCC message	Description/action
Inconsistent Index Count	The number of indexes that DBCC found to have inconsistencies.
Inconsistent Index	The name of an index that DBCC found to be inconsistent.

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

Index Error Repair

Use the **sp_iqrebuildindex** procedure to repair an index, then run **sp_iqcheckdb** in verify mode to check for index inconsistencies.

If an index is still inconsistent, drop and recreate the index, and then rebuild the index.

Note: The **sp_iqrebuildindex** procedure cannot repair FP indexes. SAP Sybase IQ has no functionality to repair FP indexes.

Analysis of Allocation Problems

Use **sp_iqcheckdb** to analyze allocation problems.

The database maintains an allocation map, also known as a free list, which tracks the blocks that are in use by database objects.

DBCC detects three types of allocation problems:

- Leaked blocks—A leaked block is a block that is allocated according to the database allocation map, but is found not to be part of any database objects. DBCC can recover leaked blocks.
- Unallocated blocks—An unallocated block is a block that is not allocated according to the database allocation map, but is found to be in use by a database object. DBCC can recover unallocated blocks.
- Multiply-owned blocks—A multiply-owned block is a block that is in use by more than one database object. At least one of the structures involved contains inconsistent data. DBCC cannot repair this type of allocation problem. If you encounter this type of error, run DBCC again, specifying a list of indexes, until you identify the indexes that share the block. These indexes must then all be dropped to eliminate the multiply-owned block.

Sample of Leaked Space Output

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

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

```
          Stat                         Value                   Flags
===========================|===========================|=====
DBCC Allocation Mode Report   |                           |
===========================|===========================|=====
** DBCC Status                |Errors Detected            |*****
   DBCC Work units Dispatched |164                        |
   DBCC Work units Completed  |164                        |
                              |                           |
===========================|===========================|=====
Allocation Summary            |                           |
===========================|===========================|=====
```

```
   Blocks Total              |8192                           |
   Blocks in Current Version |4785                           |
   Blocks in All Versions    |4785                           |
   Blocks in Use             |4801                           |
   % Blocks in Use           |58                             |
** Blocks Leaked             |16                             |*****
                             |                               |
=============================|===============================|=====
Allocation Statistics        |                               |
=============================|===============================|=====
   ...
** 1st Unowned PBN           |1994                           |*****
   ...
=============================|===============================|=====
```

If one or more dbspaces are offline, use the following syntax to show allocation problems for a particular dbspace:

```
sp_iqcheckdb 'allocation dbspace dbspace-name'
```

DBCC Allocation Errors

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

Messages in the DBCC output related to allocation problems are listed in the following table.

Table 5. DBCC Allocation Errors

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

If the Allocation Summary lines indicate no problem, but the Index Summary section reports a value for "Inconsistent Index Count," then this indicates one or more inconsistent indexes.

Repairing Allocation Problems using DBCC

Use **sp_iqcheckdb dropleaks** to repair database allocation problems.

Note: This procedure uses the **-gd** and **-gm** switches to restrict database access. For a more restrictive method, start in forced recovery mode.

1. Start the server.

   For example:

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

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

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

2. Run the stored procedure **sp_iqcheckdb** in dropleaks mode:

   ```
   sp_iqcheckdb 'dropleaks database'
   ```

   If one or more dbspaces are offline, you can repair allocation problems for a dbspace alone by running:

   ```
   sp_iqcheckdb 'dropleaks dbspace dbspace-name'
   ```

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

3. Stop the server after **sp_iqcheckdb** finishes. To stop the server, use **stop_iq** on UNIX or the shutdown button in the console window on Windows.

After allocation problems are repaired, allocation statistics appear in the DBCC output with no errors.

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

For example:

```
sp_iqcheckdb 'dropleaks dbspace mydbspace';
checkpoint;
```

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

DBCC reports statistics that do not show in this abbreviated output.

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

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

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

Forced Recovery Mode

Forced database recovery differs from normal database recovery.

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

Before Forced Recovery

Restricting database access provides greater control over inadvertent database opens during forced recovery.

Use two server startup switches to restrict access:

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

An alternate way to restrict connections is to specify

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

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

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

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

Starting a Server in Forced Recovery Mode

Forced recovery allows the server to start if the allocation map is inconsistent.

If you cannot start a server or database in a multiplex, forced recovery may be needed. Use forced recovery only when normal database recovery fails to restore the database to a running

state, and only if you see `s_buf` or free list errors during recovery. Never use forced recovery in response to SQL Anywhere errors, such as SA transaction log replay failure.

If you have followed documented recovery procedures and SAP Sybase Technical Support recommends forced recovery, follow these steps:

1. Shut down all secondary nodes using **stop_iq**.
2. Start the server with the **-iqfrec** and **-iqmpx_sn 1** flags:

   ```
   start_iq -n my_server -x 'tcpip(port=7934}'
   -gd dba -gm 1 -iqmpx_sn 1 -iqfrec
   my_db /database/my_db.db
   ```
3. Connect to the server and run:

   ```
   sp_iqcheckdb 'dropleaks database'
   checkpoint
   ```
4. Correct errors and rerun **sp_iqcheckdb**. Repeat until no errors result.
5. Shut down and restart the server normally (without the flags in Step 2).

If you cannot start your server in forced recovery mode, contact SAP Sybase Technical Support.

Using Forced Recovery without a Follow On sp_iqcheckdb

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

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

Recovering Leaked Space

Use the **sp_iqcheckdb** stored procedure in **dropleaks** mode to recover leaked storage space within the specified database.

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

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

You can recover leaked storage and force recovery either at the same time or separately. To recover leaked space within a database without doing a forced recovery, repair allocation problems using DBCC. To recover leaked space within a database after doing a forced recovery, recover leaked space using this procedure.

If repairing allocation problems using DBCC fails to recover leaked storage, then use this procedure.

Note: This procedure uses the **-gd** and **-gm** switches to restrict database access. For a more restrictive method, start the server in forced recovery mode.

1. Start the server with the `-iqfrec` option in the **start_iq** command.

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

   You specify the database name twice in a row, once to specify it as the database you are starting, and once to specify it as the database undergoing forced recovery. The **-iqfrec** option requires the database name.

2. Connect to the database you are recovering.

3. Run the stored procedure **sp_iqcheckdb** in dropleaks mode.

   ```
   sp_iqcheckdb 'dropleaks database'
   ```

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

   If inconsistency is found, drop inconsistent indexes, tables, or columns. Then run **sp_iqcheckdb** again to recover leaked space.

4. Issue a checkpoint.

5. Stop the server using your usual method.

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

Recovering Multiplex Databases

Multiplex databases have special recovery requirements.

Before recovering a multiplex database, see *Administration: Multiplex*.

Problems Reported by DBCC

Messages are reported for problems that DBCC cannot repair.

Table 6. Messages for Problems DBCC Cannot Repair

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

Index Problems that DBCC Cannot Repair

Use these suggestions to repair inconsistent indexes.

If DBCC detects a problem with an index, the name of the index is reported with the type of problem. Use **sp_iqrebuildindex** to repair a non-FP index. FP indexes cannot be repaired. Analyze index errors for indexes reported as "Inconsistent Index," when **sp_iqcheckdb** is run in default or check mode.

Depending on the type of problem, use **DROP INDEX**, **ALTER TABLE DROP COLUMN**, **DROP TABLE**, or the **FORCE_DROP** option to resolve the problem.

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

Dropping Inconsistent Indexes, Tables, or Columns

Use these suggestions to resolve issues with unrepairable indexes, columns, or tables.

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

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

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

The FORCE_DROP database option is not allowed on a secondary node. If a force drop is attempted on a secondary node, an error is returned. FORCE_DROP is a temporary option, so that the value of the option does not get propagated to secondary nodes at synchronization.

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

The following procedure uses the **-gd** and **-gm** switches to restrict database access. The **-gd** switch only limits users who can start or stop databases on a running server. For a more restrictive method, start the server in forced recovery mode.

1. Restart the server.

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

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

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

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

2. Set the temporary option FORCE_DROP to ON.

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

3. Drop all inconsistent objects.

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

4. Restart the server.

   To recover the leaked space and update the allocation map to the correct state, start the server.

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

5. Run sp_iqcheckdb.

   ```
   sp_iqcheckdb 'dropleaks database';
   ```

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

DBCC Error Messages

These are the most important messages in the DBCC output.

Table 7. DBCC Error Messages

DBCC message	Description/action
Inconsistent Index Count	The number of indexes that DBCC found to have inconsistencies.
Inconsistent Index	The name of an index that DBCC found to be inconsistent.
Extra Index RIDs Missing Index RIDs Duplicate Index RIDs	The total number of rows that are inconsistent for all inconsistent indexes.
Bitmap Verify Errors	The total number of inconsistent bitmaps in all database objects.
FP Lookup Table Inconsistencies	An unrepairable error, where the 1-byte or 2-byte FP is internally inconsistent.
Non-Completed Index Count	The number of indexes that could not be verified, because an exception occurred while checking.
Non-Completed Index	The name of an index that was not verified because an exception occurred while checking. If the exception is a future version, out of memory, or out of buffers error, commit the DBCC connection and re-run DBCC.
HG Missing Groups HG Extra Groups HG Extra Keys HG Missing Keys B-Tree Invalid Item Count B-Tree Invalid Item Count G-Array Empty Page Errors G-Array Bad Group Type Errors G-Array Out of Order Group Errors	High Group index specific errors.

DBCC message	Description/action
VDO Incorrect First Available Fields VDO Incorrect Next Available Fields VDO Incorrect Used Count Fields VDO Incorrect In-use Bitvec VDO Incorrect In-use Bitmap VDO Incorrect Partial Bitmap VDO Incorrect Deleted Bitmaps	Unrepairable errors that can cause entire tables to be inaccessible. You must force drop the inconsistent table to resolve these errors.
Block Count Mismatch	This count accompanies other allocation errors.
Blocks Leaked 1st Unowned PBN	Blocks that were found not to be in use by any database object. Use dropleaks mode to repair.
Blocks with Multiple Owners 1st Multiple Owner PBN	Blocks in use by more than one database object. Drop the object that is reported as inconsistent.
Unallocated Blocks in Use 1st Unallocated PBN	Blocks in use by a database object, but not marked as in use. Use dropleaks mode to repair.
Freelist Updated	Indicates successful allocation repair.
Freelist Not Updated	Indicates errors detected during allocation repair and the allocation repair was not successful.
Invalid Blockmap Unique ID Generator Blockmap Unique ID Generator Updated Invalid Transaction ID Counter Transaction ID Generator Updated	Errors and repair messages specific to the DBCC **resetclocks** option.
DBCC Future Version Errors	DBCC could not open the table, because DDL was performed on it. Commit the DBCC connection and re-run DBCC.
DBCC Locked Table Access Conflict	DBCC tried to open a table that another connection has locked. To ensure complete DBCC processing, make sure that no other users have locked tables in the database.
DBCC Out of Buffers Errors	The size of the IQ main cache is too small. Either increase the main cache size or run DBCC on individual objects.
DBCC Out of Memory Errors	There is insufficient system memory to complete the DBCC operation.
DBCC Meta-data Errors Blockmap Invalid Chunksize Error Count Blockmap Compression Bit Error Count Blockmap Invalid Block Number Error Count	An internal page mapping structure is inconsistent and the object needs to be dropped.
DBCC Page Read Errors	An I/O error occurred while trying to read an object. Perform hardware diagnostics.

DBCC message	Description/action
DBCC Inconsistent Disk Block Headers DBCC Decompress Errors	The storage for the object is inconsistent and the object needs to be dropped.
DBCC Unknown Exceptions	An exception of a type unknown to DBCC occurred. Check the IQ message file for details.
Unowned LVC cells Duplicate LVC cell rows Unallocated LVC cell rows	Messages indicate inconsistencies with a VARCHAR or 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 only be resolved 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 alter rename the new column to the old column. LVC is a VARCHAR column with a width greater than 255. CLOB also uses LVC.
Hash Pid: '%pid' is corrupt, count mismatch. Missing RIDs in RID mgr Hash Partition corruption, RID range mismatch	An error in the hash or hash-range load or insertion assigned row IDs incorrectly from different hash partitions. Unload and reload the table.

Troubleshooting Hints

SAP Sybase IQ provides many resources for addressing problems.

Sources of Online Support

If you cannot resolve a problem using documentation, see the SAP Sybase IQ online support Web site, MySybase.

MySybase lets you search closed support cases, software bulletins, and resolved and known problems, using a view customized for your needs. You can even open a Technical Support case online.

You can use MySybase from most Internet browsers. Point your Web browser to *MySybase* for information on how to sign up for and use this free service. For additional useful SAP Sybase Web sites, see the *Release Bulletin*.

Solutions for Specific Conditions

More information may be needed to diagnose and resolve certain issues. You can use diagnostic tools to diagnose various conditions

Decision Flow for Server Recovery and Database Repair

You may experience trouble starting a server or database, or connecting to or verifying a database.

1. Does the server start?

   If yes, go to step 2.

   If no, see *Server Operational Issues.* If you cannot start the server after following these suggestions in this section, see *Starting a Server in Forced Recovery Mode* and start the server in forced recovery mode.

   If the server does not start in forced recovery mode, call Technical Support. You may need to restore the database from backup.

2. Can you connect to the database?

   If you cannot connect to the database, see *Database Connection Issues* for troubleshooting suggestions.

   If you can connect to the database and you previously started the server in forced recovery, see *Analysis of Allocation Problems* for information on verifying database allocation and recovering leaked blocks.

If you can connect to the database, but suspect the database may be inconsistent, see *Database Verification* for information on checking the consistency of your database.

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

   If you previously started the server with forced recovery or you suspect database inconsistency, run DBCC checks to validate the database. See *Database Verification* for information on checking both index consistency and database allocation.

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

   If **sp_iqcheckdb** reports errors in the Index Summary and Index Statistics sections of the results, see *Index Error Repair* for the procedure to repair index problems using DBCC.

   If **sp_iqcheckdb** reports errors in the Allocation Summary and Allocation Statistics sections of the results, see *Repairing Allocation Problems using DBCC* for the procedure to repair allocation problems using DBCC.

Server Operational Issues

Issues that may affect server operation include startup, shutdown, unresponsiveness, and abnormal termination.

SAP Sybase IQ Will Not Start

If there is a problem starting the server, **start_iq** returns a non zero value.

If you did not specify a log file after the **-o** switch on startup, SAP Sybase IQ writes the error to the first one of the following that is defined:

* `$IQDIR16/logfiles/<servername>.nnnn.stderr`
* `$IQDIR16/logfiles/<servername>.nnnn.srvlog`
* The Systems applications log file

There are several possible causes.

Transaction Log File Does Not Match the Database

Messages appear in the server log file (`.srvlog`) and in the window where you are starting the server:

```
Starting database "dbname" (/dbdir/dbname.db)
at Fri Apr 27 2009 10:53 Transaction log: dbname.log
Error: Cannot open transaction log file
-- Can't use log file "dbname.log" since the database
file has been used more recently
Cannot open transaction log file
-- Can't use log file "dbname.log" since the database
file has been used more recently
Database server stopped at Fri Apr 27 2009 10:53
```

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

If you cannot find the correct transaction log and restoring from backup is not an option, perform an emergency recovery without a transaction log.

Server Cannot Find the Transaction Log

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

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

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

If no other option for starting the server is available, you may be able to start the server using the emergency recovery **-f** option. Contact SAP Sybase Technical Support for assistance, if necessary.

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

Server Name Is Not Unique on Your Network

If multiple servers on your system have the same name, messages appear in the server log file (`*.srvlog` or the name specified in the **-o** startup option) when you attempt to start the server using **start_iq**.

```
DBSPAWN ERROR: -85
Communication error
```

If you see these errors in the server log file and the server does not start, try to start the server using the **iqsrv16** command. The **iqsrv16** command returns a more specific error message:

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

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

Log File Has Illegal Name

If you specified a separate request-level logging file, but the file name is an illegal identifier, errors result on server startup.

```
Naming conflict: "iqdemo" --
aborting
```

```
Database naming conflict --
aborting startup
```

These errors may indicate a space in the file path specified on the **-zo** option.

Specify the **-zo** option again and enclose any file name that contains a space within quotation marks.

Server Port Number Is Not Unique on the Machine
If an SAP Sybase IQ server is running and you attempt to start another SAP Sybase IQ server on the same machine using the same port number, messages appear in the server log file (`*.srvlog`).

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

Server failed to start
```

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

You see these messages in the server log file when you start the server without specifying a port number:

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

Server Started with an Incorrect Path
When you start a new multiplex server, the database file path must match the database file path specified when creating that server.

If you use the wrong path, server startup fails, and writes these messages in the server log file (`*.srvlog`):

```
E. 08/18 07:22:19. MPX: server myserver
has been started with an incorrect catalog path
(expected path: /work/IQ-16_0/demo/mympx/iqdemo.db).
-- (st_database.cxx 7883)
I. 08/18 07:22:19. Database server shutdown due
to startup error
```

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

If you see these messages, restart the server with the expected path. If you plan to use UNIX soft (symbolic) links for server paths, you must create the soft link before you run **CREATE MULTIPLEX SERVER**.

Environment Variables Not Set Correctly
If your database configuration file parameters differ from those used by **start_iq**, make sure the correct parameters are used to start the server.

You Cannot Run start_iq
If you cannot run the **start_iq** command and you normally use a configuration file or other command line switches, try starting the server using only **start_iq** with the server name and database name.

If the server starts with this simple command, then the problem is probably caused by one or more of the switches or parameters entered on the command line or in the configuration file. Try to isolate which parameter or switch is preventing the server from starting.

If the server does not start with the most basic **start_iq** command, try starting the iqdemo demo database using your configuration file and command line switches. If the server starts with the iqdemo database, there may be a problem with your database.

If you still cannot run the **start_iq** command, use the **iqsrv16** command.

Note: Use **iqsrv16** only for troubleshooting server start up errors. Always use **start_iq** to start SAP Sybase IQ servers.

Before running **iqsrv16**, you must perform the following tasks (which **start_iq** normally does for you):

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

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

    Note: unlimit affects soft limits only. You must change any hard limits by setting kernel parameters.
*   Set all server options appropriately for your platform. See the *Installation and Configuration Guide* guide.
*   Add the path $SYBASE/OCS-15_0/lib to the environment to load the engine and required libraries before you invoke **iqsrv16**. Put this path in the environment only during testing, as follows:
    On AIX:

```
% setenv LIBPATH "${LIBPATH}:{SYBASE}/OCS-15_0/lib"
```

On other UNIX/LINUX platforms:

```
% setenv LD_LIBRARY_PATH "${LD_LIBRARY_PATH}:${SYBASE}/OCS-15_0/
lib"
```

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

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

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

The syntax for **iqsrv16** is:

```
iqsrv16 -n server-name -gm number
[ other-server-switches ] [ database-file [ database-switches ] ]
```

Note: On the **iqsrv16** command line, the last option specified takes precedence, so to override your configuration file, list any options you want to change after the configuration file name. For example:

```
iqsrv16 @iqdemo.cfg -x 'tcpip{port=1870}' iqdemo
```

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

If the server fails to start when you run the **iqsrv16** command, then attempt to start again using the **iqsrv16** utility with minimal switches and parameters. For example:

```
iqsrv16 -n <servername> <dbname>.db -c 32m
-gd all -gl all
```

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

When you start the server with the **iqsrv16** command, it does not run in the background, and messages do not automatically go to the server log. However, if you include the **-o** file name server switch, messages are sent to the named file in addition to the server window.

SAP Sybase IQ Stops Processing or Stops Responding

You can detect the cause of server unresponsiveness by looking in the SAP Sybase IQ message file.

Possible Causes

The most common causes of server unresponsiveness include:

- Insufficient disk space
- Insufficient room in main or temp buffer cache

Action

If your server seems to be prone to unresponsiveness, either while processing or during shutdown, use the **start_iq** command line option **-z** and the SAP Sybase IQ database option QUERY_PLAN = 'ON' to log useful information in the SAP Sybase IQ message (.iqmsg) and server log (.srvlog) files.

In addition to logging this information, there are other steps you can take to determine the cause of the problem:

- Check both the SAP Sybase IQ message file and the server log file for You have run out of space... messages. If you have run out of IQ main store or IQ temporary store, add the appropriate dbspace with the **CREATE DBSPACE** command. Setting the database options MAIN_RESERVED_DBSPACE_MB and TEMP_RESERVED_DB_SPACE_MB to large enough values to handle running out of space during a DDL **COMMIT** or **CHECKPOINT** is also important. A few hundred MB should be enough, but you can set these options higher for a large database.
- Determine if the SAP Sybase IQ server process (iqsrv16) is consuming CPU cycles by monitoring the CPU usage for a few minutes at the operating system level. Record this information. If the CPU usage changes, then the SAP Sybase IQ server process should be processing normally.
  If the SAP Sybase IQ server CPU usage is normal, you can examine what the server is doing, that is, what statement the server is currently executing.
- If there are no out of space indications, use Interactive SQL on a new or existing connection to gather the following information, in the specified order.

Table 8. Information to Gather for Server Unresponsiveness

Command	Informational purpose
SELECT db_name()	Database name
CHECKPOINT	Checkpoint can succeed
sa_conn_properties ># sa_conn_properties.out	Connection information
sa_conn_info ># sa_conn_info.out	Connection information
sa_db_properties ># sa_db_properties.out	Database property information
sa_eng_properties ># sa_eng_properties.out	Server property information
sp_iqstatus ># sp_iqstatus.out	Database status information
sp_iqconnection ># sp_iqconnection.out	Connection information
sp_iqtransaction ># sp_iqtransaction.out	Transaction information

If you cannot resolve the issue, contact SAP Sybase Technical Support for assistance. They can use the information you have just gathered to help diagnose the problem.

- When the server is unresponsive, generate a stack trace for each SAP Sybase IQ thread by creating a file named `DumpAllThreads` or `dumpallthreads` in the `$IQDIR16/logfiles` directory (the `%ALLUSERSPROFILE\%\SybaseIQ\logfiles` folder on Windows 64 platforms, `C:\ProgramData\SybaseIQ\logfiles` for Vista 64).

  Starting SAP Sybase IQ as recommended, using the Program Manager or **start_iq** command, sets the *IQDIR16* variable automatically. If the *IQDIR16* variable is not set, create the `DumpAllThreads` file in the directory in which `iqsrv16` was started.

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

  This stack trace information can be used by SAP Sybase Technical Support to help diagnose the problem.

- If you can connect to the database, run the **IQ UTILITIES** buffer cache monitor on the main and temp (private) buffer caches for ten minutes with a ten-second interval:
  1. Connect to the database or use the existing connection.
  2. `CREATE TABLE #dummy_monitor(c1 INT);`
  3. `IQ UTILITIES MAIN INTO #dummy_monitor START MONITOR '-append -debug -interval 10 -file_suffix iqdbgmon';`
  4. `IQ UTILITIES PRIVATE INTO #dummy_monitor START MONITOR '-append -debug -interval 10 -file_suffix iqdbgmon';`

     Let the process run for 10 minutes, then stop the buffer cache monitor:
  5. `IQ UTILITIES MAIN INTO #dummy_monitor STOP MONITOR;`
  6. `IQ UTILITIES PRIVATE INTO #dummy_monitor STOP MONITOR;`

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

  The long-term solution for avoiding deadlocks due to resource depletion is one or a combination of:
  - Restricting the number of users on the server by reducing the value of the **-gm** server startup option
  - Adding another secondary server to a multiplex
  - Increasing the processing capacity of the hardware by adding CPUs

System Failure/SAP Sybase IQ Failure

You can detect the cause of system/SAP Sybase IQ failure by looking in the SAP Sybase IQ message file.

Possible Causes

Various.

Actions

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

Server Fails to Shut Down

To shut down the server, run the **dbstop** utility or **stop_iq**, type `q` in the server window on UNIX, or click **Shutdown** on the server window on Windows.

Possible Causes

Various.

Actions

Perform these actions if the server fails to shut down.

On UNIX systems:

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

   ```
   ps -aAdeflcj|egrep "PPID|iqsrv16"
   ```

   ```
   /usr/ucb/ps -awwwlx|egrep "PPID|iqsrv16"
   ```

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

   ```
   kill -6 pid
   ```

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

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

```
kill -9 pid
```

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

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

On Windows systems:

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

2. In the Processes tab, select **iqsrv16.exe**, then click the **End Process** button to stop the database server.

3. If necessary, restart Windows.

Database Connection Issues

You may encounter issues while attempting to connect to a database.

Cannot Connect to a Database

You may experience problems connecting to a database.

Possible Causes

- Data source is not defined, or is defined incorrectly. Try connecting again with the correct user ID and password.

  A data source is a set of connection parameters, stored in the registry (on Windows) or in a file (Windows and UNIX).

- An incorrect user name or password is specified.

- User may not have permission to use the database.

- You are connecting over TDS (for example, using jConnect) and the user ID or password is longer than 30 bytes. You see:

```
Invalid user ID or password
CT-LIBRARY error:
ct_connect(): protocol specific layer:
external error: The attempt to connect to the server failed.
```

- You provide an incorrect database file name. Try connecting again with the correct database file name.

  You must supply the **DBF** parameter and the database file name to connect when you use Interactive SQL and you have restored the database from backup while connected to `utility_db`.

- Database files may be missing. The files `dbname.db`, `dbname.iq`, and `dbname.iqmsg` (where `dbname` is the name of the database) must all exist.
- A limit on the number of connections or other DBA-defined login restrictions may be exceeded.
- You have run out of disk space. Check the SAP Sybase IQ message file for messages related to disk space.
- The server name specified is incorrect. Check the name of the server and try connecting again with the correct server name.
- The server machine name or address has changed.
- When connecting from a client for the first time and the server name is not specified, providing the wrong port number can cause a failure to connect to the database. The error messages returned are:

```
Could not connect to the database.
Database server not found.
```

  When connecting from Interactive SQL, ensure that the name in the Server Name field is spelled correctly, that the network options on the network tab are correct, and that the database server has been started. Either provide the server name when connecting, or use the correct port number. To determine the server name and the number of the port on which the server is listening, run **stop_iq** (UNIX), which displays this information.
- Port number may be out of correct range or in use by another process.
- If you receive the either the message:

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

  or:

```
Database server not running.
```

  when trying to start the client, the client cannot find the database server on the network. The connection string may be incorrect or the server name cache may contain incorrect or old connection information. For example, if the server is started with a different port number, even if the client application specifies the new port number at connect time, the connection information is still taken from the server name cache.
- You specified a character set in the CharSet connection parameter and tried to connect to a server that does not support that character set.

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

Action

If you suspect that you cannot connect because there is a problem with the database, look in the `dbname.iqmsg` file to determine where the problem occurred.

`Open Database Completed` indicates that the database opened without error and the problem is related to the clients connecting. If the message does not appear, then the database may have failed while opening or recovering.

Resource Issues

Resource issues may include insufficient disk space, insufficient number of threads, thread stack overflow, and unused system resources.

Insufficient Disk Space

The SAP Sybase IQ server does not wait for additional space on an out-of-dbspace condition, but rolls back either the entire transaction or rolls back to a savepoint.

If there is not enough temporary or main dbspace available for a buffer or dbspace allocation request, the statement making the request rolls back.

At this point, the DBA can add more space to a dbspace using the **ALTER DBSPACE** or the **ALTER FILE** command. (You may choose to add files instead of dbspaces. A single dbspace can have multiple dbfiles.)

Warning! If SAP Sybase IQ holds certain system locks or is performing a checkpoint when you run out of disk space, you may not be able to add disk space. It is important for you to recognize when you are low on disk space, and to add a new dbspace before you run out of space.

Actions

- Check recent messages in the SAP Sybase IQ message log (`dbname.iqmsg`). An `out of space` message indicates that you must add another dbspace. The message in the SAP Sybase IQ message file indicates which dbspace has run out of space. If the problem occurs while you are inserting data, you probably need more room in the IQ main store. If the problem occurs during queries with large sort-merge joins, you probably need more room in the IQ temporary store.
  Check the SAP Sybase IQ message log for the following messages:
  - If a buffer or dbspace allocation request fails because there is no space in the dbspace, this error message is logged in the `dbname.iqmsg` message file:

    ```
    You have run out of space in %2 DBSpace. %1

    [EMSG_OUT_OF_DBSPACE: SQL Code -1009170L,
    SQL State QSB66, Sybase Error Code 20223]
    ```

    where %2 is the name of the dbspace.
  - If the entire transaction is rolled back on an out-of-dbspace condition, you see:

    ```
    %1 -- Transaction rolled back"

    [IQ_TRANSACTION_ROLLBACK: SQL Code -1285L,
    SQL State 40W09, Sybase Error Code 2973]
    ```

    where %1 is the error that caused the transaction to roll back, when encountered by the server during a critical operation.

- If a buffer allocation request finds a dirty buffer, but the buffer manager cannot flush the buffer due to an out-of-space condition, you see this message, and the current statement rolls back:

```
%2: All buffer cache pages are in use, ask your
DBA to increase the size of the buffer cache. %1

[EMSG_BUFMAN_ALLSLOTSLOCKED: SQL Code -1009031L,
SQL State QSA31, Sybase Error Code 20052]
```

  where %2 is the particular buffer cache throwing the exception.
- Try to connect to the database from a new connection. If this works, the database server is running, even though the query is waiting. Run **sp_iqstatus** to get more information.
- If you cannot connect to the database, check if SAP Sybase IQ is in an unusable state by monitoring the CPU usage for that processor. If the CPU usage does not change over a small time interval, then SAP Sybase IQ is probably not operational. If the CPU usage does change, SAP Sybase IQ is operational.
- Check the **sp_iqstatus** output for:

```
Main IQ Blocks Used:,10188 of 12288,
82%, Max Block#: 134840

Temporary IQ Blocks Used:,163 of 6144,
2%, Max Block#: 97
```

  If the percentage of blocks used is in the nineties, add more disk space with the **CREATE DBSPACE** command. In this example, 82% of the Main IQ Blocks and 2% of the Temporary IQ Blocks are used, indicating that more space will soon be needed in the IQ main store.
- If out-of-space conditions occur, or **sp_iqstatus** shows a high percentage of main blocks in use on a multiplex server, run **sp_iqversionuse** to find out which versions are being used and the amount of space that can be recovered by releasing versions.

Running out of Space During Checkpointing
Start in forced recovery mode and add space as soon as possible.

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

Effect of Checkpoints on Out-of-Disk Space Conditions
If SAP Sybase IQ has already run out of space when a checkpoint is requested, the **checkpoint** command fails with an error.

```
You have run out of space during the CHECKPOINT operation.
```

```
[EMSG_IQSTORE_OUTOFSPACE_CHECKPOINT:'QSB33', 1009133].
```

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

Adding Space If You Cannot Connect to a Server

If you run out of space during an operation and cannot add space because you cannot connect to the server, add space using the **CREATE DBSPACE** command.

1. Shut down the server using any of these methods:
   - On any platform, run **dbstop**.
   - On Windows, click the correct server icon on the Windows task bar to display the SAP Sybase IQ window, then click the **Shutdown** button.
   - On UNIX, run **stop_iq** or type q in the window where the server was started.
2. Restart the engine with the **start_iq** command.
3. Connect to the database.
4. Use the **CREATE DBSPACE** command to add space.
5. Re-run the operation that originally failed due to insufficient space.

Managing Dbspace Size

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

To minimize catalog file growth:

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

If the catalog store cannot extend one of its files (.tmp, .db, or .iqmsg), SAP Sybase IQ returns A dbspace has reached its maximum file size. To prevent this problem:

- Periodically monitor space usage.
- Verify that there are no operating system file size limits (such as Sun Solaris **ulimit**) where the .tmp, .db, or .iqmsg files are located. The .db and .tmp files are typically in the main SAP Sybase IQ database directory. The .tmp file is located under $IQTMP16/ <servername>/tmp, or if $IQTMP16 is not set, under /tmp/.SQLAnywhere/ <servername>/tmp.

Adding the Wrong Type of Space

If the temporary dbspace runs out of space and you omit the **TEMPORARY** keyword from the **CREATE DBSPACE** command, you cannot create a temporary dbspace.

Instead, add the file in the existing temporary dbspace as IQ_SYSTEM_TEMP.

Fragmentation

SAP Sybase IQ provides control over fragmentation by taking advantage of even the smallest unused spaces.

However, fragmentation can still occur. If your database runs out of space, even though Mem Usage listed by **sp_iqstatus** or the `.iqmsg` file shows that the Main IQ Blocks Used value is less than 100%, it usually indicates that your database is fragmented,

Freeing Space

When a connection is out of space, you cannot free space by dropping tables or indexes in another connection; the out-of-space transaction sees those objects in its snapshot version.

Reserving Space for the Future

SAP Sybase IQ automatically reserves the minimum of 200MB and 50 percent of the size of the last dbspace.

To ensure that you have enough room to add new dbspaces if you run out of space in the future, set the database options MAIN_RESERVED_DBSPACE_MB and TEMP_RESERVED_DBSPACE_MB to values that are large enough to handle running out of space during a **COMMIT** or **CHECKPOINT**.

Monitoring Disk Space Usage

You can use an event handler to monitor disk space usage and notify you when available space is running low.

The first example in this section is especially useful for monitoring space during loads. You can enable the event handler before you start the load, and disable it after the load completes.

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

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

-- This event handler runs every minute. The event handler uses
-- sp_iqspaceused to sample the space usage. If the space is
-- more than 95 percent full, a file that contains the date and
-- time is created in the directory where iqsrv16 is
-- running. The file contents are then mailed to the database
-- administrator and the file is removed.
-- This event can be enabled before a load and be used
-- to monitor disk space usage during loading. The event can
-- then be disabled after the load.

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

begin
```

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

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

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

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

end
```

The following code creates a timer-based event that monitors space usage to help avoid unexpected rollbacks, which may occur in out-of-space situations on operations without privileges. The DBSpaceLogger event is created in the sample iqdemo database.

```
CREATE EVENT DBSpaceLogger
SCHEDULE START TIME '00:00:01' EVERY 300 SECONDS
HANDLER
BEGIN
DECLARE DBSpaceName VARCHAR(128);
DECLARE Usage SMALLINT;
DECLARE cursor_1 CURSOR FOR
SELECT DBSpaceName, Usage
FROM sp_iqdbspace()
WHERE Usage > 0
ORDER BY Usage
FOR READ ONLY;

OPEN cursor_1;
idx1: LOOP
FETCH cursor_1 INTO DBSpaceName, Usage;
IF SQLCODE <> 0 THEN LEAVE idx1 END IF;
IF Usage >= 70 AND Usage < 80 THEN
call dbo.sp_iqlogtoiqmsg('Information: DBSpace' +
DBSpaceName + '''s usage is more than 70%');
ELSEIF Usage >= 80 AND Usage < 90 THEN
call dbo.sp_iqlogtoiqmsg('Warning: DBSpace ' +
DBSpaceName + '''s usage is more than 80%');
ELSEIF Usage >= 90 AND Usage < 100 THEN
call dbo.sp_iqlogtoiqmsg('Critical Warning: DBSpace
' + DBSpaceName + '''s usage is more than 90%');
END IF;
END LOOP;
```

```
CLOSE cursor_1;
END;
```

Insufficient Threads
The required number of server threads may not be available for your query.

Possible Cause
A client message similar to `Not enough server threads available for this query [-1010011] ['QXA11']` indicates that the query requires additional kernel threads for the IQ store.

Actions

- Wait for another query to finish and release the threads it is using. Then resubmit your query.
- Run **sp_iqconnection**. The column `IQThreads` contains the number of IQ threads currently assigned to the connection. This column can help you determine which connections are using the most resources. Some threads may be assigned but idle.
- If the condition persists, you may need to restart the server and specify additional IQ threads. Use the **-iqmt** server startup switch to increase the number of processing threads that SAP Sybase IQ can use.

  The default is 60 threads per CPU for the first 4 CPUs, and 50 threads per CPU for the remainder, with 3 more for system use, plus threads needed for database connections and background tasks. For example, on a system with 12 CPUs and 10 connections: `60*4 + 50*(numCPUs - 4) + numConnections + 6 = 656`. The minimum value is `numConnections + 3`. The total number of server threads cannot exceed 4096 on 64-bit platforms, or 2048 on 32-bit platforms.
- If the server runs out of threads, or if sufficient threads are not available to a connection during a restore, you may see the error `Ran out of threads. Start up server with more threads. (SQLCODE -1012024)`. The **RESTORE** command attempts to allocate a "team" of threads for the restore operation. SAP Sybase IQ attempts to allocate at least one thread per backup device, plus two threads per CPU, plus one thread to the team. Make sure you have allocated enough threads on both a per-connection and per-team basis, as well as to the server. Use the `MAX_IQ_THREADS_PER_CONNECTION` and `MAX_IQ_THREADS_PER_TEAM` database options.

Stack Overflow
You may experience problems if the thread stack overflows.

`AbortIfEndofStack` in the stack trace file (`stktrc-YYYYMMDD-HHNNSS_#.iq`), indicates that the thread stack has overflowed.

Possible Causes

- To avoid this problem, restart SAP Sybase IQ with the server parameter **-iqtss** set to 300 on 32-bit operating systems, or 500 on 64-bit operating systems. The server startup switch **-iqtss** specifies thread stack size in KB. If this is inadequate, raise the value of **-iqtss** by 72 until the problem is solved.
- If possible, identify the command that caused the error and forward it to Technical Support.

Unused Semaphores and Shared Memory Left After Abnormal Exit

Abnormal exits may leave unused semaphores and shared memory.

AIX, HP-UX, and Linux platforms use semaphores for communication between clients and servers on the same computer. Each client allocates one semaphore, as does each server. A client signals the server's semaphore when it has placed a packet for the server to read, and vice versa. The number of semaphores needed for a given system depends on how many local client applications connect via shared memory to the local server. If a client needs to allocate multiple semaphores for multiplex connections to one or more servers, it attempts to allocate all semaphores in the semaphore group.

Possible Causes

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

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

Action

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

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

For example:

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

Insufficient Buffers

If the resource manager determines that there is not enough cache to complete an operation, the operation is not started, and `Insufficient buffers` is returned.

Use one of these suggestions to resolve the issue:

- Increase the buffer cache size and re-run the operation.
- Reschedule the operation to a time when the server is not as busy and more buffer cache might be available.
- In a multiplex environment, move the workload to another node.

Processing Issues

Processing issues may be related to loads, queries, indexes, and table access.

Backups and Symbolic Links (UNIX Only)

In backups involving symbolic links, SAP Sybase IQ may create dbspaces in a directory other than the one desired.

For example, suppose that you create dbspaces in the following files:

```
-rw-r--r--   1 fiona   sybase    122880000 Feb 26 18:27 iqdemo.db
-rw-r--r--   1 fiona   sybase    122880000 Feb 26 18:27 iqdemo.iq1
-rw-r--r--   1 fiona   sybase    122880000 Feb 26 18:27 iqdemo.iq2
-rw-r--r--   1 fiona   sybase    122880000 Feb 26 18:27 iqdemo.iq3
-rw-r--r--   1 fiona   sybase    122880000 Feb 26 18:27 iqdemo.iqtmp
-rw-r--r--   1 fiona   sybase    122880000 Feb 26 18:27 iqdemo.iqmsg
```

If you create the following links first, the dbspaces will be created in the directories (or on the raw partitions) to which the links point:

```
lrwxrwxrwx   1 fiona   sybase          14 Feb 26 17:48 iqdemo.iq1 ->
 LINKS/iqdemo.iq1

lrwxrwxrwx   1 fiona   sybase          14 Feb 26 17:48 iqdemo.iq2 ->
 LINKS/iqdemo.iq2

lrwxrwxrwx   1 fiona   sybase          18 Feb 26 17:48 iqdemo.iq3 ->
 /dev/rdsk/c2t6d0s0
```

When you back up the files and restore them with the **CATALOG ONLY** option, you don't see anything telling you that these files were links; in fact, this information is not saved.

SAP Sybase IQ saves these files as though they were actually present in the directory where the symbolic links reside. When you do the restore, the files are recreated in the directories or on the raw partitions named by the database name. Whether or not the links exist at restore time, they are never used again. The database is restored to its original location.

Too Many Indexes on Table

Issues may occur when a table has too many indexes.

Possible Cause

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

Action

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

Unexpectedly Long Loads or Queries

Long loads or queries may cause issues.

Possible Causes

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

Action

Monitor paging to determine if thrashing is a problem.

- To monitor IQ paging, run the IQ buffer cache monitor.
- To monitor operating system paging, use the UNIX **vmstat** utility or other platform-specific tools, or the Windows Performance Monitor.

Reset your buffer sizes as needed.

You can also limit the amount of thrashing during a query execution that involves hash algorithms. Adjusting the HASH_THRASHING_PERCENT database option controls the percentage of hard disk I/Os allowed before the statement is rolled back and an error is returned.

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

Queries involving hash algorithms that executed in earlier versions of SAP Sybase IQ may now be rolled back when the default HASH_THRASHING_PERCENT limit is reached, and you may see either of these messages:

```
Hash insert thrashing detected.
```

```
Hash find thrashing detected. (SQLState QFA43, SQLCode -1001047)
```

To provide the query with the resources required for execution, perform one or more of these actions:

- Relax the paging restriction by increasing the value of HASH_THRASHING_PERCENT.
- Increase the size of the temporary cache (DBA only). Increasing the size of the temporary cache reduces the size of the main cache.

- Attempt to identify and alleviate why SAP Sybase IQ is incorrectly estimating one or more hash sizes for this statement.
- Decrease the value of the database option HASH_PINNABLE_CACHE_PERCENT.

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

Load Fails on Number of Unique Values

The number of unique values in a query may cause issues.

Possible Cause

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

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

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

Action

Replace the **LF** index with an **HG** index. Issue a **DROP INDEX** statement to drop the **LF** index identified in the error message. For example:

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

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

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

Cannot Write to a Locked Table

Locked tables may cause issues.

Possible Causes

The following error message is reported when writing to an object to which another user already has write access. Cannot open the requested object for write in the current transaction (TxnID1). Another user has write access in transaction TxnID2.

Action

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

The error message also includes the transaction ID of the user who is attempting to write (TxnID1) and the transaction ID of the user who is currently writing (TxnID2). For more

detailed information about the transaction that has locked the table, run the **sp_iqtransaction** stored procedure.

Managing Write Lock Contention on a Table

High contention for write locks on a table used by multiple users can impact processing, if most of the transactions can obtain the lock.

This sample stored procedure shows one method of managing contention for a write lock on a table. This procedure does not eliminate the write lock contention on the table, but does manage the contention, so that transactions can obtain the write lock.

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

You can modify this code to perform other similar tasks.

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

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

To prevent a critical update operation from failing, you may reserve write locks on all required tables in advance. For example, the following example reserves write locks on the tables `SalesOrders`, `Customers`, and `SalesOrderItems`, which are required for a hypothetical update:

```
BEGIN
WHILE TRUE LOOP
    LOCK TABLE SalesOrders, SalesOrderItems, Customers IN WRITE MODE
WAIT '30:00:00';
    If SQLCODE indicates that lock could not be acquired
    then
        SET status_msg = 'lock for required tables
        not yet acquired – retrying';
        Message to client status_msg;
    ELSE
        BREAK;
    ENDIF;
END LOOP; // Locks on SalesOrders, SalesOrderItems, Customers are
acquired
Update table SalesOrders …;
INSERT INTO SalesOrderItems …;
LOAD INTO Customers …;
COMMIT;
END;
```

Checkpoint Hints

The default values for checkpoint time and recovery time are sufficient and in most cases do not need to be changed.

The time between checkpoints defaults to 60 minutes.

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

For details on **start_iq** database options, see the *Utility Guide*.

Troubleshooting Network Communications

Network software involves several different components, increasing the likelihood of issues requiring troubleshooting.

The primary source of assistance in network troubleshooting is the documentation and technical support for your network communications software, as provided by your network communications software vendor. However, you can follow best practices and use diagnostic tools to obtain information on various conditions.

Using Compatible Protocols

If you have more than one protocol stack installed on the client or server computer, ensure that the client and the database server are using the same protocol.

The −x command line switch for the server selects a list of protocols for the server to use, and the CommLinks connection parameter does the same for the client application.

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

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

For more information about the start_iq database startup utility −x switch, see the *Utility Guide*.

Using Current Drivers

Ensure that you have the latest version of the NDIS or ODI driver for your network adapter, as appropriate.

You should be able to obtain current network adapter drivers from the manufacturer or supplier of the adapter card.

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

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

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

Powering Down Your Computer Between Restarts

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

Diagnosing the Protocol Stack Layer by Layer

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

One way to isolate network communication problems is to work up the protocol stack, testing whether each level of communication is working properly.

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

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

Once you have verified that the data link layer is working, next verify other applications using the same network and transport layers as SAP Sybase IQ are working.

Testing a TCP/IP Protocol Stack

If you are running under TCP/IP, there are several applications you can use to test the compatibility of the client computer and server computer TCP/IP protocol stack.

Using Ping to Test the IP Layer

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

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

```
ping 191.72.109.12
```

Wait to see if the packets are routed. If they are, you see output similar to:

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

If the ping works, it indicates that the computer can route packets to itself. This is reasonable assurance that the IP layer is set up correctly. Ask someone else running TCP/IP for his or her IP address, and try pinging that computer.

Before proceeding with additional diagnostics, ensure that you can ping the computer running the database server from the client computer.

Using Telnet to Test the TCP/IP Stack

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

There are several applications commonly provided with TCP/IP implementations that can be used for this purpose. To use the **telnet** command to test the TCP/IP stack:

1. Start a Telnet server process (or *daemon*) on one machine. Check your TCP/IP software documentation for instructions. For a typical command line Telnet program, enter this at the command prompt:

```
telnetd
```

2. Start the Telnet client process on the other machine, and see if you get a connection. Again, check your TCP/IP software documentation for instructions. Typically, enter an instruction similar to:

```
telnet server_name
```

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

Establishing a Telnet connection between these two machines indicates that the protocol stack is stable and the client and server should be able to communicate using the TCP/IP link. If you cannot establish a Telnet connection, there is a problem. Before proceeding with additional diagnostics, ensure that your TCP/IP protocol stack is working correctly.

Diagnosing Wiring Problems

Faulty network wiring or connectors can cause problems that are difficult to isolate.

Try re-creating problems on a similar machine with the same configuration. If a problem occurs on only one machine, it may indicate a wiring or hardware problem.

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

Checking Common Network Communications Problems

Familiarize yourself with common network communications problems and their solutions.

"Unable to start — server not found" Message

The message Unable to start — server not found when trying to start the client, indicates that the client cannot find the database server on the network.

- The network configuration parameters of your network driver on the client machine may be different from those on the server machine. For example, two Ethernet adapter cards should use a common frame type. For Novell NetWare, the frame type is specified in the net.cfg file. In Windows, look for the frame type setting in the Control Panel Network Settings.
- Under the TCP/IP protocol, clients search for database servers by broadcasting a request. Such broadcasts typically do not pass through gateways, so any database server on a machine in another (sub)network, is not found. In this case, you must supply the host name of the machine on which the server is running using the -x server startup command line option. This is required to connect to NetWare servers over TCP.

- Your network drivers or wiring is not installed properly.
- The network configuration parameters of your network driver may be incompatible with SAP Sybase IQ multiuser support.

"Unable to initialize any communication links" Message

The message `Unable to initialize any communication links`, indicates that no link can be established.

The probable cause is that your network drivers have not been installed. The server and the client try to start communication links using all available protocols, unless you have specified otherwise using the `-x` server startup option. Check your network documentation to find out how to install the driver you need to use.

Diagnostic Tools

Several tools help you diagnose various conditions.

Restoring to a New Temporary File Topology

If temporary dbfiles cannot be opened or are damaged, you can restore the database to a different temporary file topology.

1. Start the utility server so it ignores all temporary IQ file definitions in the backed up database during the restore:

```
start_iq -n utility_startup_svr -c 32m
-x 'tcpip{port=1234}' -iqnotemp
```

2. Restore the database:

```
RESTORE DATABASE 'iqdemo'
FROM '/system1/IQ16/IQ-16_0/demo/backup/iqmain'
```

3. Restart the restored database using the **-iqnotemp** flag:
4. Drop all the files in IQ_SYSTEM_TEMP:

```
ALTER DBSPACE IQ_SYSTEM_TEMP DROP FILE ALL
```

5. Restart the server without the **-iqnotemp** flag:
6. Add new temporary dbfiles to IQ_SYSTEM_TEMP:

The sp_iqstatus Stored Procedure

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

Note: The following example shows output from the `iqdemo` sample database. The sample user dbspace `iq_main` may not be present in your own user-created databases.

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

Sybase IQ (TM)	Copyright (c) 1992-2013 by Sybase, Inc. All rights reserved.
Version:	16.0.0.6552/110812/P/GA/Sun_Sparc/OS 5.10/64bit/2012-08-12 03:08:39
Time Now:	2012-09-13 10:33:19.979
Build Time:	2012-08-13 03:08:39
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:	2
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: 1-12800	IQ_SYSTEM_MAIN
DB Blocks: 1045440-1058239	iq_main
Local Temp Blocks: 1-3200	IQ_SYSTEM_TEMP
Create Time:	2013-08-17 11:31:03.313
Update Time:	2013-09-12 10:32:00.077
Main IQ Buffers:	510, 64Mb
Temporary IQ Buffers:	510, 64Mb
Main IQ Blocks Used:	8076 of 19200, 42%=63Mb, Max Block#: 1051107
Shared Temporary IQ Blocks Used:	0 of 0, 0%=0Mb, Max Block#: 0
Local Temporary IQ Blocks Used:	113 of 1600, 7%=0Mb, Max Block#: 834
Main Reserved Blocks Available:	6400 of 6400, 100%=50Mb
Shared Temporary Reserved Blocks Available:	0 of 0, 0%=0Mb
Local Temporary Reserved Blocks Available:	1600 of 1600, 100%=12Mb

IQ Dynamic Memory:	Current: 150mb, Max: 150mb
Main IQ Buffers:'	Used: 509, Locked: 0
Temporary IQ Buffers:'	Used: 8, Locked: 0
Main IQ I/O:'	I: L184357/P71 O: C18370/D25255/P20297 D: 5613 C:51.8
Temporary IQ I/O:'	I: L248471/P0 O: C22502/D25269/P4896 D: 22494 C:59.3
Other Versions:'	2 = 0Mb
Active Txn Versions:'	0 = C:0Mb/D:0Mb
Last Full Backup ID:'	0
Last Full Backup Time:'	
Last Backup ID:	0
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: 2, Recs: 413, Replays: 0/0
DB File Encryption Status:	OFF

Key to `Main IQ I/O` and `Temporary IQ I/O` output codes:

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

Check the following information:

- The lines `Main IQ Blocks Used` and `Temporary IQ Blocks Used` tell what portion of your dbspaces is in use. If the percentage of blocks in use (the middle statistic on these lines) is in the high nineties, add a dbspace.
- The `Main IQ Blocks Used` and `Temporary IQ Blocks Used` are calculated based on the line `DB Blocks` (Total Main IQ Blocks) minus `Main Reserved Blocks Available` and the line `Temp Blocks` (Total Temp IQ Blocks) minus `Temporary Reserved Blocks Available`, since the Reserved Blocks cannot be used for user operations.
- The lines `Main IQ Buffers` and `Temporary IQ Buffers` tell you the current sizes of your main and temp buffer caches.
- `Other Versions` shows other db versions and the total space consumed. These versions will eventually be dropped when they are no longer referenced or referencable by active transactions.
- `Active Txn Versions` shows the number of active write transactions and the amount of data they have created and destroyed. If these transactions commit, the "destroyed" data becomes an old version and eventually be dropped. If they roll back, the "created" data is freed.
- `Main Reserved Blocks Available` and `Temporary Reserved Blocks Available` show the amount of available reserved space.
- The lines `Main IQ I/O` and `Temporary IQ I/O` display I/O status in the same format as in the IQ message log.

Interpreting Notification Messages

By default, SAP Sybase IQ displays information about your database during insert and load operations in the IQ message log (`.iqmsg` file).

The statistics in these messages indicate when you need to perform maintenance and optimization tasks, such as adding more dbspaces. The messages also report on the progress of the load.

At the start of the insert is a description of the operation, such as:

```
In table 'tab2', the full width insert
of 2 columns will begin at record 1.
I. 02/11 13:28:14. 0000000002 Insert Started:
I. 02/11 13:28:14. 0000000002 tab2
I. 02/11 13:28:14. 0000000227 [20895]: Insert Pass 1
completed in 0 seconds.
I. 02/11 13:28:14. 0000000227 [20895]: Insert Pass 2
completed in 0 seconds.
I. 02/11 13:28:14. 0000000227 [20834]:
    1 records were inserted into 'tab2'.
```

Each time SAP Sybase IQ inserts the number of records specified in the **NOTIFY** load option, the server sends a message such as:

```
2010-05-27 13:03:49 0000000002
[20897]: 100000 Records, 2 Seconds
```

The first line shows how many rows SAP Sybase IQ has read so far and the number of seconds taken since the last notification message to read these additional rows. Even if SAP Sybase IQ reads the same number of rows each time, the amount of time varies depending on the data read (for example, how many data conversions are required). Reported time intervals smaller than 1 second are usually reported as "0 Secs".

Memory Message

The memory message displays information about memory usage of the SAP Sybase IQ server.

This line in the IQ message log (.iqmsg file) displays memory usage information:

```
Mem:  469mb/M470
```

Table 9. Memory Usage Message

Item	Description
Mem: # mb	Current memory, in megabytes, being used by this SAP Sybase IQ server.
M# mb	The maximum number of megabytes used by this SAP Sybase IQ server since it was started.

IQ Main Store Blocks Message

The IQ main store blocks message displays information about use of the blocks and buffers in the IQ main store.

This line in the IQ message log (.iqmsg file) describes the permanent IQ main store:

```
Main      Blks: U63137/6%, Buffers: U12578/L7
```

Table 10. IQ Main Store Blocks Message

Item	Description
U#	Number of blocks in use.
#%	Percentage of database filled.
Buffers: U#	Number of buffers in use. Normally this is 100% because the buffer manager leaves buffers in memory until they are needed for some other data. In general, the buffers used and buffers locked numbers are meaningless, because IQ uses buffers as aggressively and efficiently as it can.
	This value grows to the maximum number of buffers that fit in the main buffer cache. The number increments whenever a buffer is allocated, but decrements only when a buffer is destroyed, not when it is unlocked or flushed. Objects in the temporary cache release their buffers when they are finished, but in the main cache, IQ may or may not destroy the buffers because as long as a buffer is unlocked, it is available for reuse, whether it is empty, contains data, or contains destroyed data.

Item	Description
L#	Number of locked buffers. A locked buffer is in use and cannot be removed from the cache. IQ locks buffers of some objects, such as hash objects, to keep them in memory. It locks buffers of other objects, such as sorts, depending on the work-load and what it considers a fair share for that object.
	This number increments whenever you request a buffer. If you exceed the maximum while running a script, the command that exceeds fails and subsequent commands may complete incorrectly.
	Buffer locks do not use any memory. A locked buffer has a flag set in the in-memory structure and the flag exists whether or not the buffer is locked.

It is important that you recognize when the server is low on disk space and that you add a new dbspace before the server runs out of space. For an example of using an event handler to monitor disk space usage and to notify you when available space is low during a load, see *Monitoring Disk Space Usage*.

IQ Temporary Store Blocks Message

The IQ temporary store blocks message displays information about use of the blocks and buffers in the IQ temporary store.

This line in the IQ message log (`.iqmsg` file) describes the Temporary IQ Store:

```
Temporary Blks: U273/0%, Buffers: U1987/L1960
```

Table 11. Temporary IQ Store Blocks Message

Item	Description
U#	Number of blocks in use.
#%	Percentage of database filled.
Buffers: U#	Number of buffers in use. Normally, this is 100% because the buffer manager leaves buffers in memory until they are needed for some other data. In general, the buffers used and buffers locked numbers are meaningless, because IQ uses buffers as aggressively and efficiently as it can.
	Objects in the temporary cache release their buffers when they are finished.

Item	Description
L#	Number of locked buffers. A locked buffer is in use and cannot be removed from the cache. IQ locks buffers of some objects, such as hash objects, to keep them in memory. It locks buffers of other objects, such as sorts, depending on the workload and what it considers a fair share for that object.
	This number increments when you request a buffer. If you exceed the maximum while running a script, the command that exceeds it fails and subsequent commands may complete incorrectly.
	Buffer locks do not use any memory. A locked buffer has a flag set in the in-memory structure and the flag exists whether or not the buffer is locked.

It is important that you recognize when the server is low on disk space and adding a new dbspace before the server runs out of space is important. For an example of using an event handler to monitor disk space usage and to notify you when available space is low during a load, see *Monitoring disk space usage*.

Main Buffer Cache Activity Message

The main buffer cache activity message displays information about the IQ main store buffer cache.

This line in the IQ message log (`.iqmsg` file) displays information about the IQ main store buffer cache:

```
Main      I: L331224/P22 O: D25967/P7805 C:D0
```

Table 12. IQ Main Store Buffer Cache Message

Item	Description
Main: I: L#	Number of logical file reads.
P#	Number of physical file reads.
O: D#	Number of times a buffer was destroyed.
P#	Number of physical writes.
C: D#	Buffer manager data compression ratio. This is the total number of bytes eligible for compression, minus the number of bytes used after compression, divided by total number of bytes eligible for compression times 100. In other words, how much data has been compressed (what percentage it is of its uncompressed size). The larger the number, the better. Only certain data blocks are eligible for compression. Eligible blocks include indexes, (90-95% of a database) and sort sets. The sort reflects only data compression techniques used by the buffer manager. Other data compression may take place before data reaches the buffer manager, so the total data compression may be higher.

In general, assuming the buffer cache is full, you should have between 10 and 1000 logical reads per physical read. A lower value indicates excessive thrashing in the buffer manager.

More than 1000 times larger can indicate that you may have allocated too much memory to your buffer cache.

Temporary Buffer Cache Message

The temporary buffer cache activity message displays information about the IQ temporary store buffer cache.

This line in the IQ message log (`.iqmsg` file) displays information about the IQ temporary store buffer cache:

```
Temporary I: L25240/P8 O: D4749/P0 C:D0
```

Table 13. Temporary IQ Store Buffer Cache Message

Item	Description
Temporary: I: L#	Number of logical file reads.
P#	Number of physical file reads.
O: D#	Number of times a buffer was destroyed.
P#	Number of physical writes.
C: D#	Buffer manager data compression ratio. This is the total number of bytes eligible for compression, minus number of bytes used after compression, divided by the total number of bytes eligible for compression, times 100. In other words, how much data has been compressed (what percentage it is of its uncompressed size). The larger the number, the better. Only certain data blocks are eligible for compression. Eligible blocks include indexes, (90-95% of a database) and sort sets. The ratio reflects only data compression techniques used by the buffer manager. Other data compression may take place before data reaches the buffer manager, so the total data compression may be higher.

In general, assuming the buffer cache is full, you should have between 10 and 1000 logical reads per physical read. A lower value indicates excessive thrashing in the buffer manager. More than 1000 times larger can indicate that you may be overallocating memory to your buffer cache.

User Name, Connection Handle, and Connection ID

After the temporary buffer cache message, the connection handle, connection ID (SA connID), and user name are logged in the `.iqmsg` file once per database connection.

These lines in the IQ message log (`.iqmsg` file) display connection information:

```
2010-05-12 09:34:42 0000000002 Txn 173
2010-05-12 09:34:42 0000000002 Connect: 1550990889. SA connID: 1.
User: DBA.
```

The connection handle is the value shown by the **sa_conn_info** stored procedure.

Note: To correlate connection information in the **-zr** log file with that in the .iqmsg file, see *Correlating connection information between the .srvlog and .iqmsg files.*

The sp_iqcheckdb Stored Procedure

If you suspect problems in your database, try running the stored procedure **sp_iqcheckdb**.

This procedure reads every database page from disk into memory and performs various consistency checks. However, depending on the size of your database, the check can take a long time to run.

sp_iqdbstatistics displays the database statistics collected by the most recent execution of **sp_iqcheckdb**.

Checking Database and Server Startup Option Values

When diagnosing server startup, resource, or processing issues, you may need to check the current values of database options and server startup options.

For the connected user, the **sp_iqcheckoptions** stored procedure displays a list of the current value and the default value of database options that have been changed from the default. **sp_iqcheckoptions** also lists server startup options that have been changed from the default values.

When a DBA executes **sp_iqcheckoptions** he or she sees all options that are set on a permanent basis for all roles and users, and sees temporary options set for DBA. Non-DBA users see their own temporary options. All users see non-default server startup options.

The **sp_iqcheckoptions** stored procedure requires no parameters. In Interactive SQL, execute:

```
sp_iqcheckoptions
```

The system table DBO.SYSOPTIONDEFAULTS contains all of the names and default values of the SAP Sybase IQ and SQL Anywhere options. You can query this table, to see all option default values.

Finding the Currently Executing Statement

When diagnosing a problem, you may want to know what statement was executing when the problem occurred.

The **sp_iqcontext** stored procedure lists currently running statements, and identifies the user and connection that issued the statement. You can use this utility together with information provided by **sp_iqconnection**, the .iqmsg log, and the **-zr** server request log (.srvlog), as well as stack traces, to determine what was happening when a problem occurred.

To match .iqmsg log and the **-zr** server request log entries using connection information, correlate connection information between the .srvlog and .iqmsg files.

Logging Server Requests

To isolate some types of problems, especially problems with queries, log server requests.

You can enable request-level logging by either:

- Setting the **-zr** command line option when you start the server, or
- Calling the **sa_server_option** stored procedure, which overrides the current setting of the **-zr**.

Server requests are logged in `*.srvlog`. The **-zr** server startup option enables request-level logging of operations and sets the type of requests to log (SQL | HOSTVARS | PLAN | PROCEDURES | TRIGGERS | OTHER | BLOCKS | REPLACE | ALL | NONE). The **-zo** option redirects request-level logging information to a file separate from the regular log file. The **-zs** limits the size of this file.

Note: If the size of the query text being written to the log exceeds the specified limit, the query text is not truncated and is logged in its entirety.

Once the database server is started, you can adjust the request log settings to log more or less information using **sa_server_option**. These commands enable request logging of a limited set of requests and redirect the output to the file `sqllog.txt`:

```
call sa_server_option('RequestLogging','SQL');
    call sa_server_option('RequestLogFile',
                          'sqllog.txt')
```

To disable request-level logging, use:

```
call sa_server_option('RequestLogging','NO');
```

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

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

To match `.iqmsg` log and the **-zr** server request log (`.srvlog`) entries using connection information, correlate connection information between the `.srvlog` and `.iqmsg` files.

Note: SAP Sybase IQ version 15.1 modified the request log. Instead of fixed-format line prefixes, common information began being recorded as comma-delimited text. Where possible, times are recorded as "=" (meaning the same as the previous line) or +nnn (meaning nnn milliseconds after the previous line). Consequently, request logs are much smaller now than in versions earlier than SAP Sybase IQ 15.1.

In addition, more information is recorded in the request log. For queries, the information recorded is isolation level, number of rows fetched, and cursor type. For **INSERT**, **UPDATE**, and **DELETE** statements, the information recorded is number of rows affected and number of triggers fired.

Optionally, you can also choose to log statements executed within procedures and triggers.

You can select to record the short form of query plans in the request log. If procedure logging is enabled, plans for procedure statements are also recorded.

The following output shows an excerpt from the request log, when the server is started with the
-zr all option. In this example, the user connects to the iqdemo database and executes
sp_iqstatus.

There are several comma-separated fields in each line, and the first field indicates the time.
Periodically, a full timestamp is output in the form:

```
MMdd hhmmss.sss
0523 095954.807,[,1000000001,sp_iq_mpx_init,16,iq utilities status 1
```

For lines after this line, for example, "+13,C,1,UID=DBA", the offset is from the previous
line. In this case, "+13" means that about 13 milliseconds have passed since the last line. In
some cases, "=" means approximately 0 milliseconds have elapsed since the last line.

Here is the excerpt from the request log:

```
0523 095954.807,[,1000000001,sp_iq_mpx_init,16,iq
utilities status 1
+2,],1000000001,sp_iq_mpx_init,16
+1,[,1000000001,sp_iq_mpx_init,62,message STRING('IQ
Server ',@@servername,'.') to console
+2,],1000000001,sp_iq_mpx_init,62
taj% pg iqdemo.sqllog
0523 095954.807,[,1000000001,sp_iq_mpx_init,16,iq
utilities status 1
+2,],1000000001,sp_iq_mpx_init,16
+1,[,1000000001,sp_iq_mpx_init,62,message STRING('IQ
Server ',@@servername,'.') to console
+2,],1000000001,sp_iq_mpx_init,62
0523 100510.344,<,1,CONNECT
+13,C,1,UID=DBA
+83,>,1,CONNECT,1
+1,<,1,PREPARE,SELECT @@version, if 'A'<>'a' then 1
else 0 endif, isnull(property('IsIQ'),'NO'),
isnull(connection_property('odbc_distinguish_char_and_
varchar'),'Off'),
isnull(connection_property('odbc_describe_binary_as_va
rbinary'),'Off'), connection_property('charset'),
db_property('charset')
+1,>,1,PREPARE,65536
=,<,1,EXEC,65536
+79,P,1,[S]DUMMY<seq>
=,>,1,EXEC
+1,<,1,DROP_STMT,65536
=,>,1,DROP_STMT
=,<,1,PREPARE,SET TEMPORARY OPTION time_format =
'hh:nn:ss';SET TEMPORARY OPTION timestamp_format =
'yyyy-mm-dd hh:nn:ss.ssssss';SET TEMPORARY OPTION
date_format = 'yyyy-mm-dd';SET TEMPORARY OPTION
date_order = 'ymd';SET TEMPORARY OPTION isolation_level
= 0;
+1,>,1,PREPARE,65537
+1,<,1,EXEC,65537
=,[,1,*batch*,1,set temporary option time_format =
'hh:nn:ss'
```

```
+11,],1,*batch*,1
=,[,1,*batch*,1,set temporary option timestamp_format =
'yyyy-mm-dd hh:nn:ss.ssssss'
+11,],1,*batch*,1
+1,[,1,*batch*,1,set temporary option date_format =
'yyyy-mm-dd'
+11,],1,*batch*,1
=,[,1,*batch*,1,set temporary option date_order = 'ymd'+11,],
1,*batch*,1
=,[,1,*batch*,1,set temporary option isolation_level = 0
+11,],1,*batch*,1
=,>,1,EXEC
```

Request Log File Analysis

Use the stored procedures **sa_get_request_profile** and **sa_get_request_times** to read the **-zr** log file and summarize the results.

sa_get_request_profile analyzes the request log to determine the execution times of similar statements, and summarizes the results in the global temporary table satmp_request_profile. For example:

```
call sa_get_request_profile('/sys1/users/jones/iqreqs1_zr.log');
select * from satmp_request_profile;
```

sa_get_request_times also analyzes the request log to determine statement execution times and summarizes the results in the global temporary table satmp_request_time. For example:

```
call sa_get_request_times('/sys1/users/jones/iqreqs1_zr.log');
select * from satmp_request_time;
```

For more information about request-level logging, see the start_iq **-zo** switch in *Utility Guide > start_iq Database Server Startup Utility*, and the **sa_server_option** system procedure in *Reference: Building Blocks, Tables, and Procedures*.

Connection for Collecting Diagnostic Information

The database option DEDICATED_TASK lets the DBA dedicate a request handling task to handle requests from a single connection.

This pre-established connection allows you to gather information about the state of the database server if it becomes otherwise unresponsive. See the **DEDICATED_TASK** option in *Reference: Statements and Options*.

Diagnosing Communications Issues

If your server returns a communication error on startup, you may want to set the **-z** command line option when you start the server.

This switch provides diagnostic information on communications links at server startup. Information is logged to standard output from where the server started and in the srvlog file.

Reporting Problems to Technical Support

If you cannot resolve a problem using the manuals or online help, the designated person should contact SAP Technical Support or the SAP subsidiary in your area.

Each SAP installation that has purchased a support contract has one or more designated people who are authorized to contact SAP Technical Support.

Technical Support needs information about your SAP Sybase IQ environment to resolve your problem.

If you have issues with Sybase Control Center, see the see the Sybase Control Center for SAP Sybase IQ online help in SCC or at *http://sybooks.sybase.com/sybooks/sybooks.xhtml? prodID=10680*.

Collecting Diagnostic Information Using getiqinfo

SAP Sybase IQ includes a script for collecting information that SAP Technical Support needs to diagnose problems.

The **getiqinfo** script collects information about the operating system environment, the SAP Sybase IQ environment, and log files.

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

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

Before You Run getiqinfo

Collect information before running the **getiqinfo** script.

Before running the script, know the:

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

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

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

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

Running the getiqinfo Script

On UNIX platforms, **getiqinfo** is a shell script. On Windows platforms, `getiqinfo.bat` is a batch script in the `IQ-16_0\bin64` directory. The **getiqinfo** script is installed with IQ Server; it is unavailable with the Network Client.

The steps vary for UNIX and Windows platforms.

1. Start the script according to your platform:

   - At the UNIX command prompt, in the `IQ-16_0/bin64` directory, type:

     ```
     getiqinfo.sh
     ```

   - In Windows, select **Start** > **Run** > `<install_path>`
     `\IQ-16_0\bin64\getiqinfo.bat`.

2. As you are prompted, enter:

   - The directory of the database file. This is also the default location of the `.iqmsg` file, and the `stktrc*.iq` file on UNIX.
   - The base name of the database file (the file name without the `.db` suffix). This is also the default base name of the `.iqmsg` file.
   - Other directories to search for these files.
   - SAP Sybase IQ engine name (server name) and port number for this database server.
   - User ID and password of a user granted one of:
     - DROP CONNECTION system privilege
     - MONITOR system privilege
     - SERVER OPERATOR system privilege
   - The full path to the configuration file used to start the SAP Sybase IQ server, if one was used.
   - The full path to the output file in the **-zo** server option, if one was specified.

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

Information Collected by getiqinfo

The **getiqinfo** script collects information.

- Type of hardware, amount of memory, CPU type, speed, number of CPUs
- Operating system (for example, Sun Solaris 2.10)
- Swap space size
- SAP Sybase IQ version and EBF level, and Anywhere version
- Stack trace file for the date and time this problem occurred, named `stktrc-YYYYMMDD-HHMMSS_#.iq`, in the directory where you started the database server. (UNIX and Linux platforms only)
- Command or query that produced the error

- Message log file, named `dbname.iqmsg`, located, by default, in the directory where you started the database server
- Query plan (recorded in `.iqmsg` file; see the note below)
- Server logs
  - For UNIX, `IQ-16_0/logfiles/<servername>.000n.stderr` and `IQ-16_0/logfiles/<servername>.000n.srvlog`
  - On Windows platforms, if needed, you must restart the server and manually collect a copy of the console window
- Startup and connection option settings, from the configuration file (by default, `dbname.cfg`)
- Database option settings and output from **sa_conn_properties** (if the server is still running)

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

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

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

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

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

The query plan is in the message log file. The default values for these options are QUERY_PLAN = ON and QUERY_DETAIL = OFF.

If you have performance problems, set the following option:

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

Setting this option enables Technical Support to see which steps in the query processing used the time.

Correlating Connection Information Between the .srvlog and .iqmsg Files

Technical Support may ask you to set the **-zr** option on the **start_iq** command in your configuration file.

This server startup option sets the request logging level to track statements sent to the server. Parameters are ALL, NONE, or SQL. The option produces a log file named for the server with the suffix `.srvlog`.

In the *SAP Sybase IQ* message file `.iqmsg`, each connection to the server is identified by a connection handle. The `.iqmsg` message file records the errors, warnings, and tracing information for each connection.

To correlate the connection identifiers in the `.srvlog` and `.iqmsg` files to find relevant information.

1. In the `.iqmsg` file, locate a connection of interest. For example:

```
Connect:  SA connHandle: 1000000061
```

These lines show the `.iqmsg` log file contents for this connection:

```
16:14:59. 0000000062 Connect:  SA connHandle: 1000000061
          SA connID: 31   IQ connID: 0000000062  User: DBA
03/17 16:15:00. 0000000062 Cmt 12064
03/17 16:15:00. 0000000062 PostCmt 0
03/17 16:15:00. 0000000000 Disconnect:  SA connHandle: 1000000061
                SA connID: 31   IQ connID: 0000000062  User: DBA
```

2. Isolate all of the lines for the connection by searching the `.srvlog` file for the number that follows "SA connHandle" in the `.iqmsg` file.

For example, search the `.srvlog` file for "1000000061":

```
16:14:59. [,1000000061,sp_iqdbspace,48,select
str_replace(dbspaceName,'"',null) into dbspaceName_literal
03/17 16:14:59. P,1000000061,[S][0]DUMMY<seq>
03/17 16:14:59. ],1000000061,sp_iqdbspace,48
03/17 16:14:59. P,1000000061,[1]ISYSIQDBFILE<seq> JNL
dbf<ISYSDBFILE>
JNL ISYSDBSPACE<ISYSDBSPACE>
03/17 16:14:59. [,1000000061,sp_iqdbspace,58,execute immediate
with
quotes on  'iq utilities main into iq_dbspace_temp dbspace info
' || dbspaceName
03/17 16:14:59. P,1000000061,[S]INSERT ROWS
03/17 16:14:59. P,1000000061,[S]INSERT ROWS
03/17 16:14:59. P,1000000061,[S]INSERT ROWS
03/17 16:14:59. P,1000000061,[S]INSERT ROWS03/17 16:14:59. P,
1000000061,[S]INSERT ROWS
03/17 16:14:59. ],1000000061,sp_iqdbspace,58
03/17 16:14:59. [,1000000061,sp_iqdbspace,60,select
d.dbspace_name as DBSpaceName,   min(SegType) as DBSpaceType,...
03/17 16:15:00. ],1000000061,sp_iqdbspace,60
```

```
03/17 16:15:00. P,1000000061,Work[ Sort[ GrByH[ dbf<seq> JNL
ISYSIQDBSPACE<ISYSIQDBSPACE> JNL ISYSDBSPACE<ISYSDBSPACE> JH*
iq_dbspace_temp<seq> ] ] ] : ISYSIQPARTITIONCOLUMN<seq> :
idx<seq> : tab<seq>
03/17 16:15:00. [,1000000061,sp_iqdbspace,105,drop table
dbo.iq_dbspace_temp
03/17 16:15:00. ],1000000061,sp_iqdbspace,105
03/17 16:15:00. P,1000000061,[1]Work[ Sort[ sp_iqdbspace<call> ] ]
```

The connection handle in this example is 1000000061.

Support Web Site

If you cannot resolve a problem, you may find additional help on the SAP Sybase IQ online support Web site, MySybase.

MySybase lets you search through closed support cases, latest software bulletins, and resolved and known problems, using a view customized for your needs. You can even open a Technical Support case online.

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

Checklist: Information for Technical Support

You can run the **getiqinfo** script to collect information.

Information Requested	Value
SAP Sybase IQ version (for example 16.0 GA or SP number)	
sp_iqlmconfig output	
Type of hardware	
Amount of memory	
Number of CPUs	
Operating system name and version (for example, Microsoft Windows 2008 Service Pack 1)	
Operating system patch level	
Front-end tool used (for example, Business Objects Crystal Reports)	
Connectivity protocol used (for example, ODBC, JDBC, TDS)	
Open Client version	
Configuration type (single node or multiplex)	

Information Requested	Value
Message log file (`dbname.iqmsg`)	
Server log files (`server.nnnn.srvlog` and `server.nnnn.stderr`)	
Stack trace file (`stktrc-YYYYMMDD-HHNNSS_#.iq`)	
Command or query that produced the error	
Start up option settings	
Connect option settings	
Database option settings	
Schema and indexes for the database	
sp_iqstatus output	
Query plan: set options (Query_Plan, Query_Detail, Query_Plan_After_Run, Query_Plan_As_Html, Query_Plan_As_Html_Directory, Query_Timing), re-run command or query	
Screen snapshot of the problem, if possible.	

Backup Reference

Certain SQL statements have special syntax to support backup and restore operations.

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 ]

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
```

Parameters

- **TO** – 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.

- **WITH COMMENT** – 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.

- **READWRITE FILES ONLY** – restricts FULL, INCREMENTAL, and INCREMENTAL SINCE FULL backups 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 clause 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** – activates 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** – applies only when backing up to a tape device. If ATTENDED ON clause (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** *integer* – specify the number of blocks to write at one time. The 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 | INCREMENTAL | INCREMENTAL SINCE FULL** –

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

- **VIRTUAL DECOUPLED** – 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** – 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.

- **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.

Table 14. 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

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.
- **STACKER** – 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 IQwaits 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.

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 backsup 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.

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'
```

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. 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 clause 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
- *Preparing for Backup* on page 5
- *Backing Up the IQ Store and Catalog Store* on page 3

RESTORE DATABASE Statement

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

Syntax

Syntax 1

```
RESTORE DATABASE 'db_file'
    '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' ...

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

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.
- **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.

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.

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.

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.
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'`

Parameter	Description
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.
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 15. 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.

Index