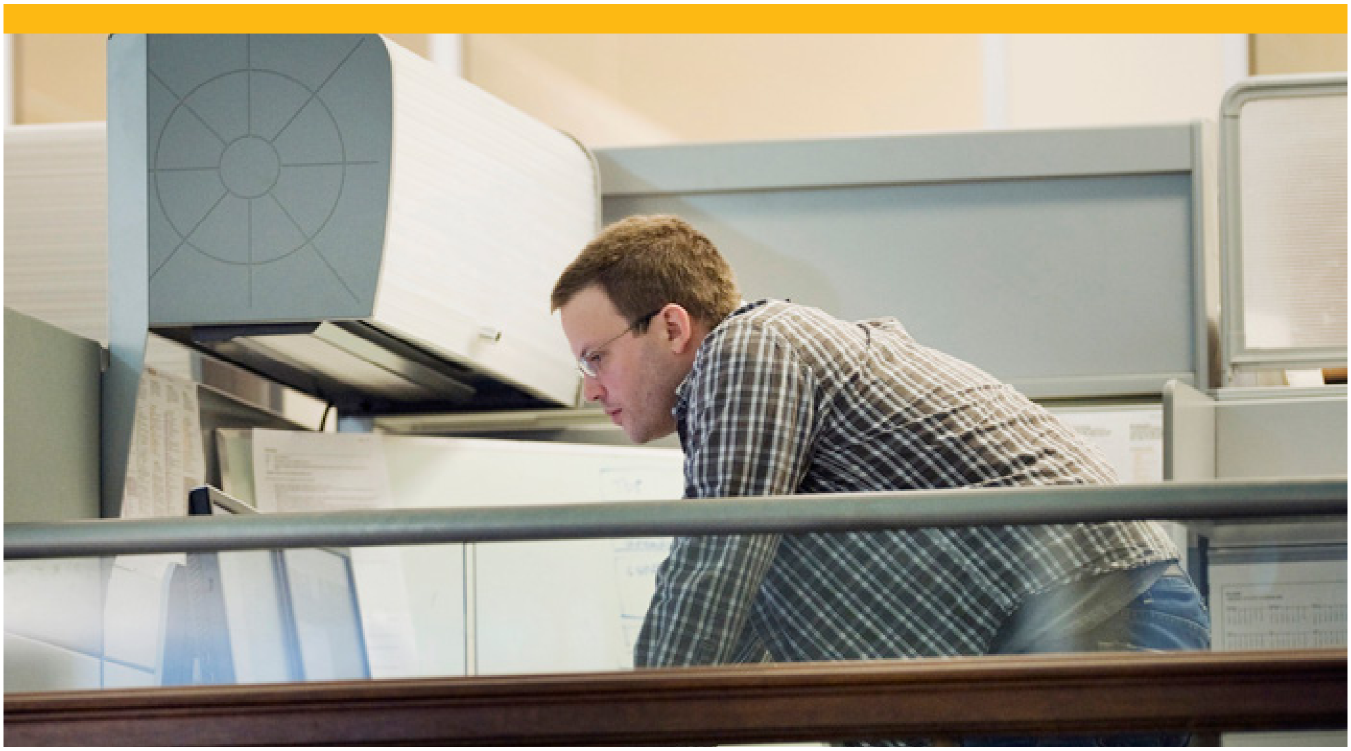


SAP® Sybase® IQ 16 Best Practices Guide

SAP World Wide Product Support

Version 1.5

Created October, 2013

Table of Contents

Executive Summary

Goals of this Document

This document presents the best practices for SAP Sybase IQ 16. It includes recommendations for using and configuring the most important features of IQ, and precautions for avoiding and troubleshooting some of its most common issues. It also highlights IQ 16's most significant new features and describes the latest enhancements to existing features.

This document is intended to serve as a starting point for optimizing your implementation of IQ 16. As a general purpose document, it is not intended to be a comprehensive guide for every environment. Rather, it is a set of guidelines, suggestions and observations on how to better use IQ 16 and its newest capabilities.

Tips for Using this Document

Please review the table of contents for an overview of the topics covered in this guide and to identify specific topics of interest.

Recommended Reading

Be sure to read the "New Features Summary Guide", "Installation and Configuration Guide", the product "Release Bulletin", and product "Cover Letter" (where applicable). Each of these documents contains information that is critical to understanding IQ's latest features and to achieving a successful installation of SAP Sybase IQ in your environment.

Introducing SAP Sybase IQ 16

SAP Sybase IQ 16 is the beginning of the IQ 16 family of releases. It was released in March 2013. The focus of IQ 16 has been to support a new generation of XLDB applications — applications characterized by sophisticated analytics on huge volumes of data. Data compression, load and query scalability, resiliency, and improved monitoring and administration have become even more critical with data volumes moving into the petabyte range, and users expecting faster answers to increasingly complex questions. IQ 16 delivers in all of these areas, as well as introducing new data security features to protect an enterprise's most valuable asset — its data. You will see a whole new level of capability in IQ 16 — more efficient use of storage, much faster loading, more aggressive distribution of queries, less I/O due to more intelligent caching of data, and a more resilient grid architecture. We hope you will be as excited as we are about IQ 16, and that it will delight you with its power and performance.

What's New in SAP Sybase IQ 16

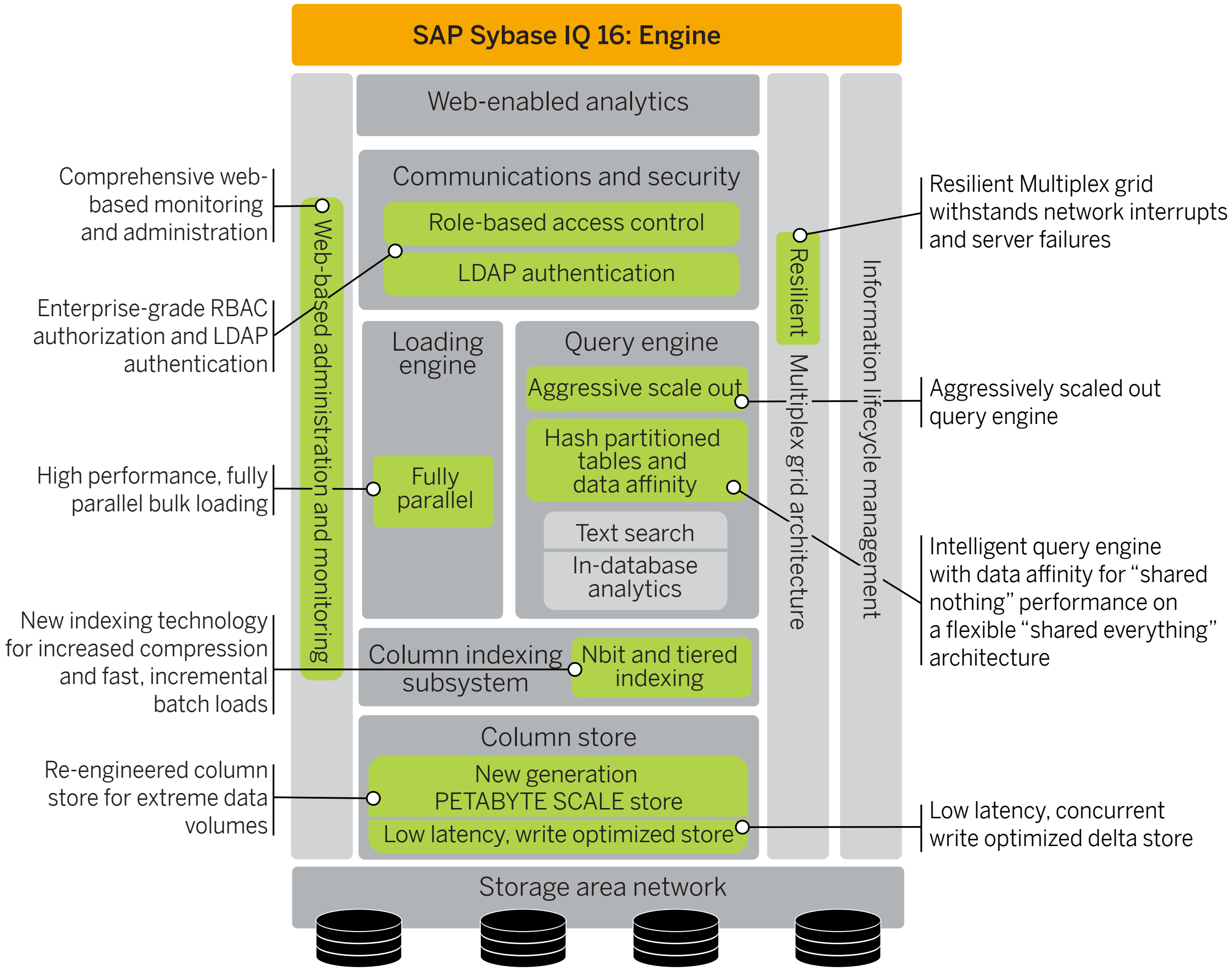
Product Goals

Like most of the SAP Sybase IQ releases, SAP Sybase IQ 16 (hereafter referred to as IQ 16) is packed with new innovations and features. IQ 16 has been one of the biggest releases in Sybase IQ's history. Prior to developing this release, conducted months of market research involving customer and field interviews, analyst and practitioner viewpoints, competitive studies, partner inputs, and internal ideas. The R&D and QA teams were increased to build the software, and formal beta programs were conducted in order to validate and hone the quality of the product. There has been a strong passion and commitment behind this release. Consequently, IQ 16 is the culmination of several years of releases designed to provide advanced analytics combined with scaled performance across multiple SMP processors and multiple host servers.

The following diagram illustrates the wide array of capabilities that IQ 16 has to offer. The areas in green represent the new features being introduced in IQ 16. You can read about IQ 16's new features in depth in the "SAP Sybase IQ 16 New Features Summary" in the product user documentation:

Hardware Sizing And Server Configuration For IQ 16

The increased parallelism and storage management capabilities in IQ 16 allow it to take advantage of more CPU and memory resources than in previous versions. For detailed information on the best practices for IQ hardware sizing and server configuration (such as main memory, cache size, storage size, page size, object placement and networking), please see the white paper: "A Practical Hardware Sizing guide for SAP Sybase IQ 16".


SAP Sybase IQ 16: Engine
Web-enabled analytics
Communications and security
Role-based access control
LDAP authentication
Loading engine
Fully parallel
Query engine
Aggressive scale out
Hash partitioned tables and data affinity
Text search
In-database analytics
Column indexing subsystem
Nbit and tiered indexing
Column store
New generation PETABYTE SCALE store
Low latency, write optimized store
Storage area network
Web-based administration and monitoring
Resilient
Multiplex grid architecture
Information lifecycle management
Comprehensive web-based monitoring and administration
Enterprise-grade RBAC authorization and LDAP authentication
High performance, fully parallel bulk loading
New indexing technology for increased compression and fast, incremental batch loads
Re-engineered column store for extreme data volumes
Resilient Multiplex grid withstands network interrupts and server failures
Aggressively scaled out query engine
Intelligent query engine with data affinity for "shared nothing" performance on a flexible "shared everything" architecture
Low latency, concurrent write optimized delta store

Also, see the later sections of this document for a few additional sizing tips such as "Sizing the Shared Temporary DBspace", and "Correctly Sizing the Coordinator".

Guidance On Server Level Features

Catalog Changes In IQ 16

SAP Sybase IQ uses SAP Sybase SQL Anywhere (SA) to manage the IQ catalog and to perform necessary server-level functions.

The changes in SA between IQ 12.7 and IQ 16 have been such that the IQ catalog needs to be rebuilt for IQ. If you are upgrading from IQ 12.7 to IQ 16, you must run a command line program called "iqunload" to migrate IQ's catalog database. This utility updates the IQ catalog only, and does not touch other IQ DBspaces. If you are upgrading from IQ 15 to IQ 16, you do not need to run iqunload. You only need to issue the SQL command "ALTER DATABASE UPGRADE".

For thorough information on how to migrate your IQ database, please see the section on "Migration Tips" within this document, and the chapters "Migration (Windows)", or "Migration (Linux and Unix)" in the IQ user documentation.

View Dependencies And Altering Schemas

- When you have a view referencing another table or view, the link between both objects is stored in the SYSDEPENDENCY table.
- The system procedure "sa_dependent_views ('view_name', 'owner')" lists the dependencies for all the dependent views on a given table or view.
- When you alter the schema of an underlying table, the server will attempt to recompile the dependent views.
- If this "auto recompilation" fails, then manual recompilation of these views may be necessary.
  - For example:

```
create table T1 (C1 int, C2 bit, C3 varchar(10));
create table T2 (C1 int, C2 bit, C3 varchar(10));
create view V1_T1 as
       select C1, C2, C3 from T1 union all select * from T2;
create view V1_V1 as select C1 from V1_T1 where C2 = 1;
create view V2_V1 as select C1 from V1_T1 where C3 = 'A';
```

- **Note:** In the example on the previous page, the 1st "level" view, "V1_T1 has" two dependent views. Therefore, the sa_dependent_views procedure will return 2 rows.
  - For example:

```
> select * from systab where table_id in (select dep_view_id
     from sa_dependent_views (,V1_T1','DBA'));
> 2 rows returned
```

The Request Log (ZRLOG File)

The request log contains a record of individual requests made between the IQ server and an application or user. It is an extremely useful troubleshooting aid since it shows the activity between the server and its clients. It is also a good starting point for performance and tuning, especially when it is not obvious where the performance bottlenecks are. For more details on the benefits of using the request logging feature, please see the document "Administration: Request logging". To make use of request logging:

1. Generate The Request Log:

- Use the IQ startup parameters "-zr all -zo zrlog.txt" to generate the request log file.

2. Load The Request Log:

- To load the request log file, use the procedure, "sa_get_request_times".
- For example: call sa_get_request_times (zrlog.txt);
- This function loads the zrlog's content into a temporary table called satmp_request_time.

3. Query The Request Log:

- Once the satmp_request_time table has been loaded, you can query this table to determine detailed run time information.
- For example:
  - To select all the operations performed by connection ID 129, enter:
    ```
    select * from satmp_request_time where conn_id = 129;
    ```
  - To select all from all connections executed during a period:
    ```
    select * from satmp_request_time
       where start_time between ‘2010-07-26 10:30:00‘ and ‘2010-07-28
       21:00:00‘;
    ```

  To select the statement text of the connection id 317:

  ```
  select (stmt) from satmp_request_time where conn_id = 137;
  ```

- **Notes:**
  - When you exit from dbisql, the data in the satmp_request_time temporary table is deleted.
  - By default, dbisql displays 5,000 rows and a max of 100,000 rows. Therefore, if you have a large file, you may need to redirect the result set using the “OUTPUT TO” command.

Tip for Reading Long Statement

- Since statement text is returned on a single line, it can be difficult to read long statements. To format statement text for easier reading, use the function **sa_statement_text**. This formats the text statements so that individual items appear on separate lines.
- For example: `call sa_statement_text (‘<statement>‘);`

IQ 16 Storage And Indexing — What’s New

NBit Dictionary Compression

Continuous NBit dictionary compression replaces 1, 2, and 3 byte FP dictionary compression. All data types except LOB (character and binary) and BIT datatypes may be NBit columns. NBit dictionary compression provides more efficient compression and greater performance than 1, 2, and 3 byte dictionary compression.

The IQ UNIQUE column constraint explicitly determines whether a column loads as Flat FP or NBit FP. An IQ UNIQUE n value set to 0 specifies a Flat FP. An n value greater than 0 but less than FP_NBIT_AUTOSIZE_LIMIT (database option that controls rollover from NBit to flat) creates an NBit column initially sized n. Use IQ UNIQUE in cases where you want to load the column as Flat FP or when you want to load a column as NBit when the number of distinct values exceeds the auto-size limit.

IQ UNIQUE no longer controls the bit encoding. It is only used to size the dictionary. Hence, the larger the IQ UNIQUE that is specified, the more memory is required to instantiate a dictionary. If you do not specify the IQ UNIQUE, IQ can then size the dictionary itself and roll the nBit to a Flat FP if there is not enough memory. If you do specify IQ UNIQUE, IQ will always try to get enough memory to instantiate the nBit to that size.

With nBit, each page of the FP index is encoded using as many bits as it needs to encode the values on that page. So rather than the whole FP being rolled over from 8 bit (1 byte) to 16 bit (2 byte) to 24 bit (3 byte) encoding, individual pages at converted to a specific bit encoding when needed. And also, rather than being able to only use 8,16, 24 bit encoding, nBit can encode using the entire range 1 thru 31 bit encoding.

Tiered HG Indexes

New tiered High Group (HG) index structure decouples load performance from HG index size.

In IQ 15, load throughput could degrade as the amount of data in an HG index increased. As the index grew, loading the same amount of data could take more time. The new tiered structure decouples load performance from the HG index size to increase throughput.

The **`CREATE_HG_WITH_EXACT_DISTINCTS`** option determines whether newly created HG indexes are tiered or non-tiered. If this option is ON, all new HG indexes are non-tiered. To take advantage of the new structure, set this option to OFF. Use **sp_iqrebuildindex** to convert non-tiered HG indexes to tiered HG and vice-versa.

New Row Level Versioned Store

New in IQ 16 is the RLV (Row Level Versioned) Store – This is an in-memory delta store that is optimized for high-performance row-level updates. It is currently supported only in an IQ simplex configuration (single server). The RLV store acts as a staging area for write events. If a table is enabled for the RLV store, then all **LOAD TABLE, INSERT, UPDATE,** and **DELETE** events write directly to the RLV store. In-memory data in the RLV store is automatically (and periodically) merged into the IQ main store. You can set parameters for automatic merges, and you can merge on-demand.

The IQ main store is optimized for high performance queries, rather than table writes. The RLV store adds high-performance write capability to your simplex database. You can use the RLV store in conjunction with your existing insert mechanisms such as bulk loading. For example, your applications can make real-time updates during business hours, while overnight a bulk load operation bypasses the RLV store and loads updated data directly to the IQ main store.

If a table is enabled for RLV storage, SAP Sybase IQ table DDL behavior is unchanged. DDL events on a table (adding a column, for example) require an exclusive lock on that table.

Query Engine — What's New

Hash Partitioned Tables

Table partitioning can improve query performance by dividing large tables into smaller, more manageable storage objects. IQ 16.0 now supports hash, range, and hash-range partitioning schemes. Hash partitioning is part of the base product. Range and hash-range partitioning is part of the VLDB option.

- Hash partitioning maps data to partitions based on partition-key values processed by an internal hashing function. Hash partitioning can enhance join performance on large tables and distributed queries (DQP). You may not assign hash partitions to individual DBspaces.
- Range partitioning divides large tables by a range of partition-key values established for each partition. Partitioning allows you to group together data that should be grouped together. That is, you can group data together that shares a common attribute. Partitioning also separates data that should be separated, such as historical data from current data. As part of an information life cycle management strategy, range partitioning can shorten backup and restore times; provide a finer level of granularity for data validation; and support tiered storage.
- Hash-range partitioning provides the benefits of hash partitioning and range partitioning. Hash partitioning provides the best distributed query performance; range sub-partitioning enhances administrative tasks as part of an information life cycle management strategy.

The range partitions of a range or hash-range partitioned table can be altered to:

- Add/drop partitions.
- Split a partition.
- Merge adjacent partitions.
- Un-partition the table.

You cannot add, drop, split or merge a hash partition.

Data Affinity

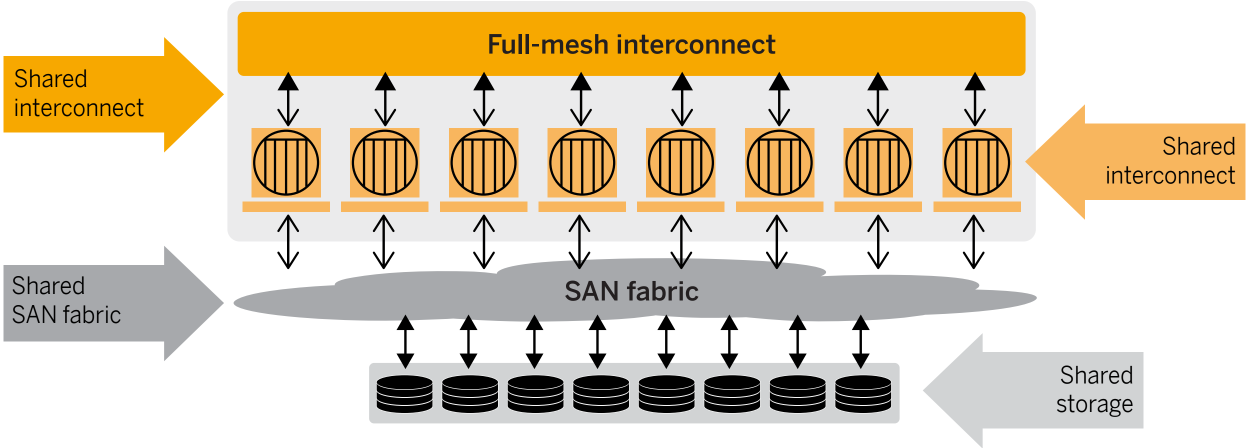
Hash partitioned tables support another IQ 16 feature: data affinity. In IQ 15, the query engine was able to break a query into independent fragments, and pass the fragments to separate servers for processing. However, the query engine did not have knowledge about which data resided in which server's cache so that it could assign work more intelligently. Now in IQ 16, the query engine tracks cache contents across the Multiplex, and sends work to the machine that already has the pertinent data in its cache. Caches stay hot, and I/O is reduced. Hash partitions support this data affinity feature by pre-partitioning data so that data sets are ready to be fetched into cache.

Data affinity is automatically managed by IQ. The user does not need to specify that certain data needs to be "pinned" to a particular machine. This gives the query engine more flexibility to leave data resident in cache, or to move it out to make room for a completely different kind of query.

For optimum scale-out performance, SAP recommends that you have 4-8 multiplex nodes with total memory across the cluster that can hold most of the working set data for the queries. Affinity is automatically enabled for Multiplex databases. No user intervention or maintenance is required.

Multiplex Best Practices

An SAP Sybase IQ Multiplex is a group of SAP Sybase IQ servers that connects to a central store for permanent shared data. Each server maintains its own local storage for catalog metadata, and transaction logs. An IQ Multiplex can have a mix of reader and writer nodes. This improves performance by allowing IQ to separate the update workloads across multiple nodes. For example, if you have a system with many tables undergoing frequent data refreshes, you can create multiple writer and reader nodes. This will allow IQ to provide separate nodes to manage all the data changes and separate nodes to handle all the database read operations. Here is a depiction of the architecture:



Compute nodes:
- Can be 86, BM Power, HP Itanium and Solaris Sparc servers
- Each node can have a different CPU core count, memory size
- One node is designated as the coordinator for meta data management
- Any node can be a write node
- Any node can be a read node

Shared storage:
- Can be SSD, FC, SATA or a combination of all
- Shared storage can be physical SAN with FC or iSCSI
- Shared storage can be virtual SAN over DAS
- Storage can be logically grouped with compute
- Storage can be tiered for data aging

The Coordinator Node

The coordinator node is the heart of the IQ multiplex configuration. It is the only node in the multiplex with the ability to coordinate all read/write operations, including schema changes.

The coordinator performs the following functions:

- Manages all global read-write transactions
- Maintains and manages the global catalog
- Maintains the table version log (TLV log).
- Maintains metadata updates for DML and any DDL changes on IQ or SA store objects and syncs them across all nodes.
- Stores information about DDL operations.
- Communicates information about new table versions to the secondary servers (readers or writers).
- Coordinates shared IQ table locking between nodes.
- Only the coordinator node has the ability to manipulate shared DBSpaces (like adding DBFiles to the IQ_SHARED_TEMP DBspace), and administer logical servers for distributed query processing and workload isolation.

Correctly Sizing The Coordinator Node

It is important to note that all DDL operations are run on the coordinator node regardless of which node the SQL was initiated on. Typically, this is not an issue. However, there is a potential issue when creating indexes on fully populated tables, since the actual work is performed by the coordinator rather than the writer node where the create index command was initially executed. Consequently, when sizing the host for the IQ coordinator, consider the following:

- Place the coordinator on an isolated host (where there will be no contention with other IQ nodes).
- Be sure this host has the capacity to support all DBA maintenance tasks and any real-time DDL operations such as *create index*.
- Configure the coordinator to also play the role of writer.

Simulating Local Store Functionality In IQ 16

Local Stores Were Deprecated In IQ 15

- If you are migrating from an IQ 12.7 configuration with local stores, you will need to transfer the data from your local stores into IQ 16's main shared store. The *iqlsunload utility* is provided to assist with this task.
- Also, the node level isolation accomplished in IQ 12.7 by using local stores is now provided by a combination of login policies and DBspace management in the shared store of IQ 16.

HowTo Stimulate A Local Store In IQ 16

- Here is a sample process for simulating "local store" type functionality in IQ 16:
  - Create a User DBspace.
  - GRANT RESOURCE to a specific group of users.
  - Create a login policy that will restrict these users from accessing any other node except the writer nodes where the storage isolation has been implemented.
  - Assign this login policy to these specified users:
    - Only this select group of users will be able to create tables and give select permissions.
    - Only these users will be able to login into the specified servers.
  - Load, inserts, deletes, updates and selects are now restricted to a specific server.

Inter-Node Communication (INC)

- Communication between the coordinator node and the secondary nodes (readers and writers) is handled by the Inter-Node Communication (INC) layer. (SQLRemote was deprecated in IQ 15.) The INC layer provides TCPIP based communication between nodes with the help of remote procedure calls. This communication link consists of a heartbeat process and pooled connections.
- INC communication occurs between the Coordinator and secondary nodes via a dedicated connection. This "dbo" owned connection is established when a secondary node starts, and it remains active for the entire time the node is active. The coordinator monitors this connection to check for the secondary nodes' "aliveness". This connection is only used for non-transactional communication. For transactional communication between the coordinator and the secondary nodes, INC uses a connection from the INC connection pool. This connection can be customized using the MPX_MAX_CONNECTION_POOL_SIZE and the MPX_MAX_UNUSED_POOL_SIZE options.
- Note the following:
  - INC reports messages in the .iqmsg file.
  - The INC Connection Pool is separate from the users' connection pool.
  - All secondary nodes use INC connections for DDL commands and read/write DML operations. Whenever a user performs a DDL or READ/WRITE operation on a secondary node, an INC connection is "associated" with the user connection. This connection persists until the command commits or is rolled back at which time, the connection is returned to the INC connection pool.
  - If IQ runs out of connections from the INC connection pool, you will see this error:

```
"The number of connections in the connection pool have exceeded the
 total number of connections allowed in the connection pool. No more
 connections can be allocated."
```

Tip: Set The Max Number of User Connections (-GM Option)

- For high-volume servers, the default setting for the maximum number of user connections may not be sufficient. That is, if there are many user connections performing concurrent DDL or READ/WRITE operations, or if user transactions are very large (many users performing concurrent data loads or updates without a commit), then the default setting for the maximum number of user connections may not be adequate. In this case, the setting for max user connections must be increased for the secondary server. To do this, use the "-gm" server option. This option specifies how many users can connect to the secondary node at the same time.
- It is also recommended that you set the MPX_MAX_CONNECTION_POOL_SIZE to a reasonable value. Making this number too large can result in too many INC connections to the coordinator.
- For high-volume servers, you may also want to consider reducing the DDL or READ/WRITE workload on the secondary node by performing some of these operations directly on the coordinator.
- In order to optimize INC connection performance, it is recommended that you carefully assess the expected concurrent read/write and DDL workload for your applications, and to monitor this workload on an ongoing basis.

Multiplex Inter-Node Communication (MIPC)

- IQ 16 can aggressively distribute a query across all the nodes in a Multiplex. The Multiplex Inter Node Communication (MIPC) mesh is a peer-to-peer inter-node communication infrastructure that supplements the Inter Node Communication (INC) protocol, and is used for distributed query processing. MIPC allows Multiplex nodes to communicate directly with each other, and supports the more robust communication requirements of DQP.
- There are both public and private configuration options for MIPC. The private option allows you to specify host-port pairs that Multiplex servers will use exclusively for DQP related communications. If no private interconnection configuration is provided, MIPC uses the host-port pairs specified for other types of communication such as external user connections and INC connections. Note: TCP/IP is the only supported protocol for host-port paring at this time.
- Internal testing has showed that a private MIPC network may have significant performance benefits over a shared MIPC network.

Global Transaction Resiliency

IQ 16 brought resiliency improvements to the Multiplex grid. With IQ 16, if a communication failure with the coordinator node occurs, the secondary writers will suspend their global read-write transactions on the user connections and resume these transactions when the INC communication is restored. The suspend and resume feature is enabled in the following situations:

- INC connection failure and re-establishment
- Coordinator shutdown or crash and restart
- Coordinator shutdown or crash and failover

With this feature in SAP Sybase IQ, DML read-write transactions on Multiplex Secondary (Writer) node can survive temporary communication failures between Coordinator and Secondary node with Writer role and temporary failure of the Coordinator due to Coordinator Server crash, shutdown or failover.

When a DML transaction is executed by a user on a Secondary (Writer) node involving RW (e.g. insert/load) operations on shared database objects, a global transaction is initiated by the Writer on the Coordinator node. The transaction starts on an INC connection from the Writer to the Coordinator Node.

If temporary INC connection failures such as temporary communication failures and Coordinator failures are resolved within the time-out defined specified by MPX_LIVENESS_TIMEOUT (default 3600 secs = 1 hour), the INC connection is reestablished and global transactions will continue uninterrupted. If a failure persists longer than the MPX_LIVENESS_TIMEOUT value, the global transaction cannot be resumed, and is rolled back.

Global transaction status (active/suspended) can be seen using the sp_iqconnection System Stored Procedure on a Writer node, or the sp_iqmpxsuspendedconninfo System Stored Procedure on the Coordinator Node. The Sp_iqmpxincstatistics procedure can be used to get a snapshot of the aggregate statistics of INC status since server startup.

- **Note**: This new feature does not affect transactions initiated on the Coordinator Node.

Dynamic Collision

Dynamic collisions only occur on secondary nodes. A collision occurs when a schema change is committed while a secondary node is querying the same object. A dynamic collision results in the termination of the connection to the secondary. If such a scenario occurs, then you will see message similar to the following in the .iqmsg file:

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

Tip: Minimize Query Disruptions

- In order to minimize disruptions to query operations, it is advised that you avoid performing schema changes when the multiplex is being heavily used.

Logical Servers

SAP Sybase IQ PlexQ introduced the concept of “logical servers” in IQ 15. A logical server is a grouping of physical nodes in the Multiplex. When a query is executing on a machine in a logical server, only the nodes within the particular logical server will participate in the query. This allows workloads to be isolated from each other for security or resource balancing purposes. Logical servers are elastic — physical machines may be added to or removed from a logical server dynamically as workload demand changes.

- Multiple logical servers can be created to serve different groups of applications or users. You can dynamically add and remove nodes from a logical server to accommodate the individual application resource needs. You also have flexible options for choosing which servers participate in the execution of a distributed query, or whether to turn it off entirely on a global basis or for particular queries.
- When distributed query processing is enabled, SAP Sybase IQ distributes the query execution (in parallel) to those nodes that are members of the logical server.

‘Server’ Logical Server

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

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

Logical Server Login Direction

SAP Sybase IQ 16 provides load balancing within a logical server. When a user tries to log into a node which is overloaded, the connection is redirected to a Node with less load in the same logical server. The DBA will need to enable logical server policy LOGIN_REDIRECTION to ‘ON’. Setting it to ‘ON’ will enable login redirection for logical servers specified by the user’s login policy. When LOGIN_REDIRECTION is set to ‘OFF’ (Default: OFF), logical redirection is disabled at the logical server level, allowing external connection management.

1. Create a logical server policy login_redirectiion to ‘ON’
   Create logical server policy LRD LOGIN_REDIRECTION=’ON’

2. Associate the new logical server policy with a logical server
   Alter logical server LS policy LRD

Logins are only redirected if LOGIN_REDIRECTION is enabled and:

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

Multiplex DDL Syntax Changes

Note DDL syntax changes in IQ 16 for the following SQL commands (see "User Documentation"):

- Create LS POLICY
- Create Logical Server
- Alter logical Server
- Alter LS Policy
- Alter login Policy
- Comment on LS Policy
- Drop LS Policy

Multiplex Do's

- All multiplex nodes must be at the same ESD level. IQ 16 does NOT support mixed-versions in a Multiplex. Also, all nodes should be running the same operating system.
- IQ Multiplex does not support DIRECT IO, only raw devices for shared IQ stores.
- For optimal performance, keep the IQ logs (such as .db, transaction.log, and iqmsg) on local disks.
- For performance reasons, secondary writers should ONLY run large loads, and not a high volume of sub-minute read/write transactions. These small transactions entail proportionately more communication overhead with the coordinator. It is fine for the coordinator to run large loads, as long as it is running on a machine with enough resources. The coordinator should be treated as a first class node that is capable of handling workloads equal to any other node in the multiplex. That is, run large loads on the coordinator as long as: 1) operations on other nodes do NOT slow down; or, 2) you do NOT exhaust resources on the coordinator and need more CPU/RAM to load.
- Keep the catalog as small as possible by not placing user tables in this space. Catalog size has a negative impact on synchronization.
- Only the Coordinator node can perform modifications to the IQ Main Store DBspace files.
- Database backups can only be executed on the coordinator of a multiplex configuration.
- Prior to adding or altering a DBfile in the IQ_SYSTEM_MAIN DBspace, you must shut down all secondary nodes. Once the DBfile modifications are completed, the secondary node must be synchronized.
- Prior to performing a failover procedure, it is absolutely critical that you shutdown the current coordinator server. If you perform a failover when the current coordinator is running, then you may corrupt the Multiplex.
- It is recommended that you ensure that there are no active connections to a logical server before dropping a member of a logical server or the logical server itself. Use sp_iqconnection to examine the logical server context of active connections. (See next section: "Distributed Query Processing").
- The Create/Alter/Drop Multiplex Server commands can be executed from any server in the multiplex as long as the coordinator is running. However, it is important to note that when executing these commands, IQ does not verify if the secondary server has access to the shared DBspaces. You must ensure that each secondary server has access to the shared DBspaces before adding it to the Multiplex.
- Since an object's transaction management data is stored on the IQ_SYSTEM_MAIN DBspace, it is very important that this DBspace be adequately sized.

Multiplex Don'ts

- Never start a secondary node in single-node mode. If this occurs when a good coordinator is present, then the Multiplex will think you are promoting the secondary node to the role of a coordinator. Having two coordinators in a Multiplex setup is not supported, and could lead to data corruption.
- Do not run excessive amounts of Global DDL commands on secondary nodes. Use local temporary tables wherever possible as DDL on local temporary tables does not have the INC overhead.
- A designated failover server cannot be excluded from the multiplex unless it is the last un-excluded server in the multiplex.

Distributed Query Processing (DQP) And Logical Servers

Distributed Query Processing (DQP)

- IQ 16 supports Distributed Query Processing (DQP), which improves performance and availability in multiplex configurations by dividing queries into multiple independent pieces that are distributed across multiple nodes in a group of servers called a "logical server".
- Distributed query processing comes as part of the Multiplex Grid option for SAP Sybase IQ, and is enabled by default. In general, any query that qualifies for parallel processing also qualifies for DQP.
- Refer to the "Using Sybase IQ Multiplex" user guide, and "Scale-Up and Scale-Out Performance Features"

Sizing The Temporary DBspace

- Distributed Query Processing (DQP) requires temporary storage that all multiplex nodes can share in order to assign work, and to store and transmit intermediate results.
- Sizing the shared temporary DBspace depends on the distributed query processing workload. In a worst case scenario, you will need an amount equal to the sum of all local temporary DBspaces.
- As a starting point, it is best to create a Temporary Storage DBspace that is equal to half the size of the peak local temporary storage usage for each node.
- To calculate the peak local temporary storage usage, run the sp_iqstatus system stored procedure during a typical query workload on a pre-DQP server.
- To track shared temporary store usage on an ongoing basis, you can execute the sp_iqspaceused system stored procedure on a regular basis.
- If you have a high speed network, you might want to set an option DQP_ENABLED_OVER_NETWORK, to allow results sharing across the network, instead of using the shared temporary dbspace.

Migration and Upgrade Tips For IQ 16

Whenever possible, SAP recommends that you migrate to the latest version of IQ. This will give you access to the newest features and fixes in the product. The following section contains recommendations for avoiding some of the more common migration and upgrade issues.

Recommended Reading

One of the most important (and sometimes overlooked) steps in preparing for a new SAP installation is the review of SAP's latest release information. Please be sure to read the New Features Summary, the Installation and Configuration Guide, the Release Bulletin, and Cover Letter (when available) for the version of SAP Sybase IQ you plan to install. Each of these documents contains information that is important to understanding IQ's latest features and critical to a successful installation of SAP software.

Backup Before An Upgrade Or Migration

- SAP STRONGLY recommends that you backup your entire database before performing a migration or upgrade. Without this, recovering from upgrade issues becomes very difficult.
- This backup should include the following files: Also, save the lengths of the following files:
  - The IQ_SYSTEM_TEMP DBspace file, typically named dbname.iqtmp.
  - All SYSTEM DBspace files (such as the IQ Catalog Store), that are typically named dbname.db.
  - All user DBspaces, typically named dbname.iq.
  - The transaction log file (required for system recovery), typically named dbname.log.
  - Files belonging to any additional DBspaces that have been added to the IQ main store.
  - Files belonging to any additional DBspaces in the catalog store. These are listed in SYSDBSPACES.
- Also, 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.
  - The files or raw devices used to hold temporary DBspaces.

- It is also a good idea to backup the following optional ASCII message files (which may be useful in troubleshooting) such as:
  - dbname.iqmsg
  - $IQDIR15/logfiles/*.srvlog
  - $IQDIR15/logfiles/*.stderr
  - If IQ message log wrapping is enabled, backup the complete set of .iqmsg log files. If problems occur during a restore, the .iqmsg file contains information that proves that the database was shut down before the backup started.
- For more details on backup files, see the section on "Back Up The Right Files", in the SAP Sybase IQ 16 User Documentation.

Backup After An Upgrade Or Migration

Although optional, SAP recommends backing up your entire database AFTER a successful migration.

Establish A Performance Baseline for Your New Install

We strongly recommend establishing a PRE-MIGRATION baseline of query performance and query results for all your applications. This baseline can be used to evaluate post-migration performance and query results. Without this baseline, troubleshooting post-migration performance and data related issues becomes very difficult.

Test Your New Install

Based on many years of technical support, we cannot stress the following enough: Be sure to THOROUGHLY TEST your applications before moving a new version of IQ into production.

Run sp_iqcheckoptions Before And After An Upgrade

- Run sp_iqcheckoptions before and after an upgrade in order to verify that your post-upgrade option values are correct.
- This is critical, since options are sometimes reset to their default values as a result of the upgrade process. Therefore, you will want to identify any such options and return them to their desired values.

Verify Database Consistency Before And After Migration

- SAP strongly recommends that you verify the consistency of your databases BEFORE and AFTER you migrate to a new version of IQ. You can do this by running sp_iqcheckdb. Without this check, it can be very difficult to diagnose and resolve any consistency issues that might occur.
- For example, to verify a database, do the following:
  - Issue a CHECKPOINT command.
  - Run sp_iqcheckdb in verify mode: sp_iqcheckdb ‚verify database'.
  - If you run the sp_iqcheckdb procedure from Interactive SQL, redirect output to a file by typing the following: **`sp_iqcheckdb ‚verify database' >& filename`**
  - Issue a COMMIT statement.
- For information on interpreting the sp_iqcheckdb results and corrective action, see "Administration: System Recovery and Database".

Migrating From IQ 15 To IQ 16 (Alter Database Upgrade)

- The changes in SA between IQ 15.x and IQ 16 do NOT require an unload/reload of the IQ Catalog.
- Migrating from IQ 15.x to IQ 16 only requires running the ALTER DATABASE UPGRADE command.
- Read the *Migration* chapter in the SAP Sybase IQ 16 user documentation for detailed instructions.

Migrating From IQ 12.7 To IQ 16 (Unload And Reload The Catalog)

You must upgrade to IQ 12.7 ESD #5 at a minimum before you migrate to IQ 16.

IQ Catalog Changes

- IQ uses a SAP Sybase SQL Anywhere (SA) to manage the IQ catalog, and to perform server-level functions such as database connections, security, SQL parsing and the handling of geospatial data types. Therefore, it is important to note that the IQ Catalog is a set of SA tables.

- Similar to SAP Sybase ASE's "master database", the IQ catalog contains metadata that describes DBspace usage, permissions, database options, and the layout of tables, columns and indexes. SA also performs functions such as syntax parsing, client connectivity, remote server connectivity (using the CIS layer), request logging, and auditing.
- As IQ versions have changed, so have the underlying versions of SA changed. In IQ 12, the SA engine was SA 9. In IQ 15.0-15.3, the SA engine was SA 11.0.x. In IQ 15.4, the engine was SA 12.0.1. In IQ 16, the engine is SA 16.
- Because the catalog is rebuilt, the sys.sysobject creation_time will be replaced by the migration time. If object creation dates are important to the customer's IQ installation, the DBA should save them manually in a user table before migrating.

Use The iqunload Utility to Migrate From IQ 12 To IQ 16

- Between IQ 12.7 and IQ 15/16, the changes in SA were such that upgrading a database from IQ 12.7 to IQ 16 is very different than upgrading from one version of IQ 12 to another version of IQ 12. Upgrading from IQ 12.7 to IQ 16 requires an unload/reload of the catalog.
- This is accomplished by using the "iqunload" utility. Note: User DBspaces are unaffected and do not need to be reloaded.
  – For more details on using the IQUNLOAD utility, please see the "SAP Sybase IQ 12.7 Database Migration of the SAP Sybase IQ 16.0 Migration Guide" for your operating system.

Tips For The iqunload Utility

This section presents several tips for using the IQUNLOAD utility to migrate an IQ 12.7 database to an IQ 16 database.

Use The Schema Unload Mode Before Performing The Load

- Before performing the actual unload, we recommend that you first run the iqunload utility in the "schema unload" mode. Do this by using the syntax: **`iqunload –n`**
- Running the iqunload utility in "schema unload" mode lets you connect to an existing IQ 12.7 server and generate a reload.sql script that contains the entire schema for your 12.7 database. You can also run this script against a new SAP Sybase IQ 16 database to recreate the schema of the legacy database. Running iqunload in the schema unload mode provides two major benefits:
  – Allows you to verify the correctness of the schema by applying it to a new SAP Sybase IQ 16 database.
  – Provides an estimate of the time the migration process will require.

Use The "-iqro" Switch When Performing An Unload/Reload Of The Catalog

- The "start_iq" server parameter "**-iqro 1**" opens the server's DBSpaces in read-only mode. It is critical that you use the "**-iqro**" switch to start IQ in READ-ONLY mode during several steps of the unload/reload process. Failure to do so can result in serious recovery issues, should a problem occur.

Known Issues For The iqunload Utility

There are several known issues that you may encounter when performing the iqunload. Although they are not common, you should be aware of the issues listed below.

Drop The Global Temporary Tables

- If you have created a GLOBAL TEMPORARY TABLE, drop it before running the iqunload (iqunload -au). Otherwise, the iqunload utility may fail with an error. You can then recreate this table after the migration.

Reset The DATE_FORMAT And DATE_ORDER

- If you have changed the default values for the DATE_FORMAT or DATE_ORDER database options, reset them before running the iqunload. You can do this using the syntax: "SET option DATE_FORMAT = ;" and "SET option DATE_ORDER =;". Setting these options back to their default values permits iqunload -au to succeed. Otherwise, you may encounter an error such as:

```
Reload of SYSIQINFO failed due to the following errors:
The reloaded table "SYS.SYSIQINFO" contains a row
with key column(s) "create_time" values: ( 2011-12-27 11:44:12.557 )
which has unexpected values  or column mismatches
with original table "SYS.SYSIQINFO" with key column(s)
"create_time" values( 2007-04-25 16:26:33.754 ):
Reloaded Table "commit_txn_id" column value: ( 0 ) ,
Original Table "commit_txn_id" column value: ( 2883768 ).
```

Drop Any Java Classes OrJava Procedures

- If you have any Java stored procedures or Java Classes, drop them before running the iqunload utility (iqunload –au). If you do not do so, then your iqunload may fail with an error similar to the following:

```
***** SQL error: External environment could not be started, ‘main thread‘ could not be
found. This file contains the statement that caused the schema reload failure.To complete
the schema reload, you need to modify your database to avoid generating the statement
below.

Once you have modified your database appropriately, re-run the schema reload process.NOTE:
You may want to generate the schema only for the database being reloaded,and load this
schema into an empty database to check for SQL errors.

INSTALL JAVA FROM @byte_code.
```

Failing To Start The IQ Server in Read-Only Mode May Result In Errors

- Failing to start your IQ 16 write server (with its migrated database) in READ-ONLY (-iqro) mode prior to performing the load verification step may result in a corrupted migration.

Resolving the “Parse Stack Overflow” Issue

The Parse Stack Overload Error

- During IQ 16 testing, you may receive an error message similar to the following:

```
Syntax error near ‘parse stack overflow‘ on line 25SQLCODE=-131, ODBC 3 State=‘42000‘
```

- Typically, this error indicates that IQ has encountered a SQL statement that is too long. This happens when a program generates a lengthy SQL command.
- This error message is generated at the SAP Sybase SQL Anywhere level, where the SA server manages connections and parses commands before the query is passed to IQ. In IQ 16, the structures generated in parsing commands consume a bit more memory than in IQ 12.7. Therefore, a long command that ran successfully in IQ 12.7 (but was close to overflow), now crosses the line into overflow in IQ 16.
- You have several options for avoiding this overflow error:

  1. Break the query into smaller pieces.

  2. Use a larger IQ catalog page size. In this case, go to the next larger page size. he default page size for IQ is 4096; and, the permissible page sizes are 4096, 8192, 16384, and 32768 bytes. Note: The IQ catalog page size is DISTINCT from the IQ database page size.

Fix The Stack Overflow Error By Using iqunload To Increase The Catalog Page Size

- If you choose to increase your catalog page size, and you are already in the process of running the iqunload utility as part of an IQ 16 migration, then we strongly recommend you use the iqunload utility to rebuild your server with a larger page size. For this option, consider the following procedures:
  - Use the “-ap” option to enable the catalog database to be created with a larger page size. For example:

    ```
    iqunload -au -ap 8192 -c [connection parameters]
    ```

  - Also, add the -gp parameter to your IQ 16 server configuration file (.cfg) using the same value as in the iqunload -ap command. For example: -gp 8192. This will keep your IQ configuration file and the startup messages written to your .iqmsg log in synch with your catalog database configuration.
  - Lastly, be sure to increase the size of your catalog cache. For example, if you increase your catalog page size from 4K to 8K, you must double the amount of catalog cache specified in your “-c” or “-cl”, and “-ch” startup configuration parameters. Otherwise, the catalog cache will only accommodate half the number of 8K pages as it did 4K pages. Per the “Maximum Catalog Page Size” section of the Administration Guide: “If you use larger page sizes, remember to increase your cache size. A cache of the same size will accommodate only a fraction of the number of the larger pages, leaving less flexibility in arranging the space.”

Other Solution Options

- If you cannot rebuild the IQ 16 server by rerunning iqunload, you will need to do the following:
  - Use the IQ data extraction facility to offload the database.
  - Recreate the database with the larger catalog page size.
  - Load the database from the extracts.
- Also, please note the following:
  - IQ Backup and Restore can NOT be used to change the catalog page size.
  - Using the "-gp" startup parameter without rebuilding the database may seem to offer an easier solution than rebuilding the database. But, by itself, it does not. When starting IQ, the "-gp" value sets the size of catalog pages in memory for the IQ server. For example, "-gp 8192" results in an 8KB catalog page size in memory. If the database was built with a 4096 catalog page size, then at startup, the 4KB pages are loaded into 8KB page frames and the other 4KB per memory page is wasted. In this situation, the following boot-time messages appear in the .srvlog file:

```
...
E. 07/29 09:13:13. Note: database page size is smaller than the cache page size, resulting
in inefficient use of memory.
...
I. 07/29 09:13:16. Using a maximum page size of 8192 bytes
...
I. 07/29 09:13:38. Performance warning: Database ”iqdemo“ has a page size of 4096 that
does Not match maximum of 8192 set for server, causing inefficient use of cache ….
```

  - **Note**: The "-gp" parameter has somewhat different usage in a standalone SAP Sybase SQL Anywhere database implementation, but in SAP Sybase IQ, its effect is as described above.

Use The "-iqro" Switch When Performing The Alter Database Upgrade Command

The "-iqro" Switch

It is critical that you use the "-iqro" switch for starting your server in READ-ONLY mode before running the "ALTER DATABASE UPGRADE" command. Doing so will avoid serious recovery issues if you encounter problems with the database migration.

The Max Connections Error

- In certain cases, when using "–gm 1" (max connections = 1) with the ALTER DATABASE UPGRADE statement, the upgrade command will fail with this error: **"Not allowed while 'DBA' is using the database"**.
- This error is the result of setting the –gm flag to 1 when there are one or more automated processes that immediately connect to the IQ server whenever it boots. This can happen when your server is already registered to the SCC for administration. In this case, a connection is started before you can log in and run the "alter database upgrade" command.
- You can verify this by running the "sp_iqconnection" procedure to check for multiple connections. If there are, this command will return multiple rows, such as:

```
1  SQL_DBC_114d49330  DBA 2011-06-27 15:40:37.645 OPEN IQUTILITYOPENCURSOR 40:37.0 0 NONE
0

64610206 2011-06-27 15:36:10.0   0    0    13   9593    12  local

2  SCC_IQ-110627-15:36:12 311 DBA 2011-06-27 15:40:31.558 EXEC_ANY_IMM 15:40:31.0  0 NONE
0

   0  2011-06-27 15:36:31.0   0     0     21  492   2   TCPIP

(2 rows)
```

Resolving The Max Connections Error

There are several ways to resolve this max connection issue:

Method 1:

Drop the "extraneous" connection for the SCC server:

- Run the sp_iqconnection procedure to view the connections.
- Drop the extra connection. For example: drop connection 2
  Note: This drop remains in effect until you reboot the server.
- Rerun the "alter database upgrade" command. The upgrade should complete successfully.
- Shut down and restart the coordinator normally (without the -gm 1 and -iqro 1 startup flags).

Method 2:

Start IQ with startup flags for two or more connections and read-only mode.

- For example: **`-gm 2 -iqro 1`**

Method 3:

Temporarily start IQ on a different port using the "-x" switch:

- The –x switch controls the port that IQ uses when you start the server. Starting the server on a port that is not "known" to any other process will "hide" the server from these other automated processes.
- On a simplex server, use the –x switch to specify a new port number. For example:

```
start_iq @yourserver.cfg yourdb.db -x 'tcpip{port=<new_port>}'
```

- On a multiplex server use both the –x and -iqmpx override switch to specify a new port number. For example:

```
start_iq @yourserver.cfg yourdb.db -iqmpx_ov 1 –x 'tcpip{port=<new_port>}'
```

- **Note:** This procedure can be used any time you want to "hide" your IQ server from other users or processes in order to perform maintenance operations.

Rebuilding FP Indexes for Enhanced Compression

- In IQ 16, there have been a number of enhancements to compression techniques. The new compression techniques are automatically applied to all newly added values.
- But, new compression techniques are NOT automatically applied to existing (pre-IQ 16) data.
- Therefore, in order to compress pre-16 data using IQ 16's enhanced compression algorithm, you must do one the following:

  1. Rebuild the FP indexes using the sp_iqrebuildindex stored procedure. **Note:** Currently, this cannot be used with varchar and varbinary columns wider than 255 bytes.

  2. Move the data to another DBspace using the ALTER TABLE … MOVE statement.

Licensing Changes for IQ 16

There are no new licensing options for IQ 16. The options are the same as for IQ 15:

- Unstructured data analytics (UDA)
- Advanced security
- Multiplex grid
- Very large database (VLDB)
- In-database analytics

The InfoPrimer option has been retired. Customers are encouraged to migrate to SAP Data Services for ETL capability into IQ.

All the new features in IQ 16 are part of the base product, except for LDAP authentication, which is part of the Advanced Security Option.

The Evaluation Version vs. The Evaluation License

The Evaluation Version for IQ

- An Evaluation version of IQ (Demo Version) is an installation without a license. It is NOT the same as an "Evaluation License". With the evaluation version:
  - This license option can be selected while installing IQ. Once selected, the installer will NOT ask you to choose the PE (Product Edition) or LT (License Type). The subsequent iq.default.lmp file created will contain empty PE and LT values. For example:
    - PE =
    - LT =
  - In an evaluation version, all standard and optional features of IQ are available under 'installation grace'. You will always see the "Checked out graced license…" message for each individual license. There is no limit on the quantity of license available under grace. The only restriction is that these licenses expire after the grace period, which is 30 days from the creation date of the database. In the case where a server has started multiple databases, then it is 30 days from the creation of the oldest database. At expiry, the IQ server shuts down gracefully.

The Evaluation License For IQ

- An Evaluation License is a valid IQ license:
  - This license is checked out like a regular license (not in grace). Also, a separate license for each optional feature is needed and must be downloaded and installed. For example, the IQ_CORE evaluation license only enables IQ's core functionality. If you want to evaluate the security feature, you will need to download and install the IQ_SECURITY license separately.
  - An evaluation license is only available by request when a potential customer needs to evaluate the product features beyond the usual 30 day grace period provided by an Evaluation Edition.
  - If you create a new database, you can install it with the evaluation license to obtain a 30 day grace on this new server. You cannot use the same license to start the old server.

SySAM2

SAP uses a utility called SySAM2 (Sybase Software Assets Manager – version 2) to install, maintain and verify SAP product licenses. Below, is a discussion of SySAM2's features. For more details, please review the "Licensing Your Software" chapter in the "Installation and Configuration Guide" for the platform of your choice.

SySAM2 License Files

- SySAM2 uses the information stored in two special license files:
  - The license manager file: Typically referred to as the ".lmp" file.
  - The license file: Commonly referred to as the ".lic" file.
- The ".lmp" file: This is a nw license manager property file that is created when you install a new IQ 16 server.
  - The full name of this file is: **`"<your_iq_dbname>.lmp"`**.
  - This is IQ's human-readable license file and there is one these ".lmp" files for each IQ database you create.
  - IQ will use the default license (iq.default.lmp) as a template when creating an ".lmp" file for a new server.
  - In IQ 16, the iq.default.lmp file is located in the $SYBASE/IQ-16/SySAM directory.
- The .lic file:
  - This is the license file you acquire when you purchase IQ 16.
  - You must create and download this license file whenever you download the IQ installation software or install IQ on a new machine.
  - The .lic file should be placed in the $SYBASE/SYSAM-2_0 directory.
  - Its full name commonly includes a date and the host machine's name. For example: **`BigUNIXBox_201106080415.lic`**

Overview Of SySAM Licensing Checks

- SySAM performs licensing checks at various times during the server's runtime cycle. Some licenses are checked at startup, some upon access of a feature, and some are checked at both times. Further, most license are checked periodically during the server's runtime according to a "heartbeat" cycle.
- For those licenses that are checked at startup, some are only checked after a particular feature has been added to the database. For example, SySAM only performs a VLDBMGMT licensing check after a 1 Tb threshold for IQ store size has been exceeded, or after a Multiplex node has been created.
- The following describes when some of these checks occur:
  - Always Checked on Startup:
    - IQ_CORE
  - Checked on startup if the feature has been enabled:
    - IQ_VLDBMGMT -> See note below, on graced licenses.
    - IQ_MPXNODE
  - Checked upon feature access:
    - IQ_VLDBMGMT
    - IQ_SECURITY
    - IQ_UDA
    - IQ_IDA
- Please refer to the most current "Installation and Configuration Guide" for more information on license feature sets and changes in the licensing behavior.

Scan The Log For SySAM Errors (And Take The Appropriate Action)

- Because SySAM checks for license issues on a regular basis during the server's run cycle, SySAM errors can be detected and reported at any time. When a message is issued, it is important that you take the appropriate action in a timely manner. Failure to correct these warnings or errors may result in the server shutting down or failing to boot. Therefore, it is CRITICAL that you regularly scan your IQ log for SySAM messages.
- For example, creating a new DBSpace will automatically generate a graced license for the VLDBMGT option. This can occur when creating a new DBSpace that crosses the 1 Tb threshold. In this case, SySAM will "check-out" a new VLDBMGT license for your site. Depending on your existing license allotments, this new license may be checked out as a 30-day graced license. When this occurs, a "graced license" warning will be written to the IQ log. You MUST resolve this issue within the 30-day grace period, otherwise the server will shutdown.
- In the case of a graced license, you have two choices:

  1. Remove the graced license; or,
  2. Purchase a new license before the grace period expires.

Tip: Scan the Log for SySAM Errors

- It is strongly recommended that you create a process to programmatically scan the IQ log for SySAM messages.

Troubleshooting "Product Edition" And "License Type" Errors

The information in your license manager property file (".lmp") file must match the license information in your license file (".lic" file). If not, you may encounter SySAM licensing errors. Therefore, note the following:

- IQ uses the license types specified in the configuration parameters of your ".lmp" file. These values are stored in two lines that start with "PE=" (Product Edition) and "LT=" (License Type).
- If SySAM detects a mismatch between your license manager file and your license file, it will write an error message in the IQ log. If you encounter this error; AND, you HAVE a valid license, consider the following procedure:
  - Edit the .lmp file so that the Product Edition (PE) and License Type (LT) values match those in the VENDOR_STRING of your .lic file. That is, the values for PE and LT in the .lmp file should match those in the .lic file.
  - For example:
    - If the vendor_string in your .lic file indicates your product edition is an "Enterprise Edition", then PE should be equal to "EE". That is PE=EE.

    - If the vendor_string in your .lic file indicates your license type is “CPU”, then LT should be equal to “CP”. That is LT=CP.
- – Note:
  - The .lic file is located in the $SYBASE/SYSAM-2_0.
  - In 15.4, the .lmp file is in your $SYBASE /IQ-15_4/SySAM directory. You can also use the output from the lmutil (described below) to identify the correct values for PT and LT.
  - If a server is started with a “graced” license, then the Product Edition and License Type values will be set to blanks (i.e., “PE=” and “LT=”).

For more SySAM troubleshooting tips, see the “Troubeshooting SySAM” section of the installation guide, and the “SySAM 2 User’s Guide”.

SySAM Debugging Options

- You can use the license manager utility (lmutil) to generate an output file that will assist in diagnosing license problems. Note the following:
  - The lmutil is located in the $SYBASE/SYSAM-2_0/bin directory.
  - Use the ‘lmdiag’ option to generate an output file that will assist in diagnosing license checkout problems.
  - Usage: $SYBASE/SYSAM-2_0/bin/lmutil lmdiag –c <license_file_name>
- To determine the status of a specific license server, use the following syntax: **sysam status –a**
- To determine if the license server is supporting a given license feature, use the following syntax: **sysam status –f <feature_name>**
  - For example:
  - This command: **sysam status –f IQ_CORE**
  - Returns:

```
lmutil - Copyright (c) 1989-2005 Macrovision Europe Ltd. and/or Macrovision Corporation.
All Rights Reserved.
Flexible License Manager status on Thu 5/5/2011 09:51
Users of IQ_CORE:(Total of 10 licenses issued; Total of 0 licenses in use
```

Operational Management

IQ Administration Tools

IQ can be monitored and administered in several ways:

1. By using SAP Control Center (SCC), SAP’s web-based GUI tool
    - **Note:** Sybase Central has been retired with IQ 16
2. By using selected IQ stored procedures.
3. By using a variety of OS-level commands and utilities.

In high volume environments, all three of these methods may be necessary to effectively monitor and administer SAP Sybase IQ.

SAP Control Center (SCC)

SCC is SAP’s web-based GUI for administration and monitoring within a single tool. It can be used to monitor and administer SAP Sybase ASE and RepServer as well as IQ. SCC consists of three main components:

1. The SCC Host Server: A separately installed server running on your network.
2. The Local SCC Agent: A software component that is installed as part of the IQ server installation. It communicates with the SCC server’s SCC agent.
3. An Internet Browser: You can use any network browser that supports flash player. This includes browsers such as Firefox, MS Internet Explorer or Google Chrome.

Stored Procedures For IQ Management

Stored procedures like *sp_iqstatus, sa_procedure_profile, sp_iqsysmon, sp_iqworkmon*, can be executed individually or in a script that is run on a periodic basis or triggered by events. You can also use IQ Utilities to monitor buffer cache and other statistics.

OS-Level Tools for IQ Managment

OS-Level tools such as *vmstat*, also provide useful information on memory and CPU use and disk activity. Some of the OS tools that might assist in debugging issues can be found in the "Diagnostics Checklist" section of this guide.

SCC Administration Features for IQ 16

The SAP Sybase IQ installer includes IQMAP 3.2.7 plugin for SCC 3.2.8 for these platforms:

- LinuxAMD64
- Solaris x64
- Solaris Sparc
- Win64

On Win32 and LinuxAMD32, SCC 3.2.8 and IQMAP 3.2.7 must be installed via the SCC 3.2.8 installer.

SCC for IQ 16 is not supported on:

- AIX64
- HP64
- IBM LinuxPPC64

SCC New/Enhanced Features:

In IQ 16, SAP Control Center replaces Sybase Central. The IQ plug-in for SCC 3.2.7 provides complete IQ Server administration and management functionality:

- **External Environment Administration:** Prior to SCC 3.2.7 only the Java external environment was supported. But this release of SCC supports administration of external environments: ESQL, Java, .Net, ODBC, Perl and PHP.
- **Table Administration:** SCC 3.2.7 allows DBAs to perform table management tasks:
  – Creating/deleting a base table
  – Creating/deleting a global temporary table
  – Creating/deleting a proxy table
  – Modifying table properties
  – Moving a table to another DBspace
  – Managing table columns
  – Managing table constraints
  – Managing table indexes
  – Managing table partitions
  – Generating table DDL commands
  – Viewing table data in the Execute SQL window
- **Domain Administration:** DBAs can create, view and update domains. The Administration Console tree lists domains under schema objects
- **View Administration:** DBA can perform view management tasks such as:
  – Creating view
  – Modifying view
  – Disabling view
  – Recompiling and enabling a view
  – Deleting a view
  – Generating view DDL commands
- **Web Service Administration:** Following web service management tasks can be performed:
  – Creating a Web service

- Deleting a Web service
  - Disabling a Web service
  - Enabling a Web service
- Modifying a Web service
- Generating Web service DDL commands
- Enhanced Logical Server Support:
  - When multiplex server instance is authenticated, if logical server resource for logical server doesn't exist, then SCC automatically generates one logical server resource per logical server. SCC sets the resource name to the multiplex name followed by a colon and the logical server name. For example, IQMPX1::IQLS1
  - Logical server policies can be created, edited, and deleted to control logical server access and context.
  - Monitor logical servers from the Perspective Resource View, Heat Chart, or Administration Console.
  - SAP Sybase IQ can now do load balancing by redirecting new multiplex logins to node with a lighter load in the same logical server.
  - Two SCC generated administration scripts include new connection parameters in the connection string:
    - stop_server.sh (.bat on windows) includes LS=SERVER
    - sync_server.sh (.bat on windows) includes LS=COORDINATOR
- **Login Mappings:** Map windows user profile or a Kerberos principal to an existing user in the database to maintain a single user ID for database connections, operating system, and network logins.
- **Transport layer security:** Configure transport layer security with RSA encryption in a multiplex environment.
- **Row-Level Versioning (RLV) administration:** New RLV data store can be used in a simplex database to perform row-level updates, inserts, deletes, in real-time. When a table is enabled for storage in the RLV data store, multiple writers can write to different rows of the table concurrently.
- **Global Transaction Resiliency:** Monitor the suspended, resumed, and rolled back status of INC (Internode Communication) connections for global transactions.
- **Remote Access:** By creating Remote Server definitions, proxy tables, Remote procedures and External logins, remote data can be accessed from other data sources
- **LDAP Server Administration:** LDAP servers can be created, modified and deleted.
- **Role-Based Security Administration:** User and role management can be performed for role-based users, user-extended roles, and standalone roles. Tasks include create, modify and delete, as well as grant and revoke system privileges and object permissions to users and roles.
- **SAP Sybase IQ Server and SCC agent logs:** View and filter logs, copy and paste from log snapshots. The log displays as static snapshot that does not refresh automatically.

Tip: Backup the SCC Repository

- The SCC repository is located within the following directory: $SYBASE /SCC-3_2/services/Repository.
- It is highly recommended that you make full and incremental OS-level backups of this directory on a regular basis; especially when managing multiple servers.

Troubleshooting AIX Multi-Processor Issues

- If IQ is installed on a multi-processor AIX box, you might see the following discrepancy in reporting of processors:

```
From iqdemo.001.srvlog:
I. 05/01 10:31:51.
I. 05/01 10:31:51. 14 physical processor(s) detected.
I. 05/01 10:31:51. Running AIX 5 3 on PPC
I. 05/01 10:31:51. Server built for PPC processor architecture

From iqdemo.iqmsg:
I.      05/01     10:31:53.    0000000000     Using licenses from: /testhost/iq15/
SYSAM2_0/licenses:/testhost/iq15/IQ-15_0/demo
I. 05/01 10:31:55. 0000000000 Checked out license for 7 IQ_CORE
   (2010.11150/15-n ov-2010/13E7 0944 F517 63F0) will expire Tue Nov 16 00:00:00 2010.
I. 05/01 10:31:55. 0000000000 WARNING: Sybase IQ functionality that requires the
   IQ_CORE license will be disabled on Tue Nov 16 00:00:00 2010, unless a suitable
   IQ_CORE license is obtained before that date.
```

Note: In the output on the previous page, the difference between the number of CPU's (14) and the number of cores (7) occurs because we report the physical CPU's instead of the logical CPU's on AIX. This is a reporting issue only. There is no problem with SySAM's detecting and checking-out the correct number of licenses.

DBISQL And Open Client Utilities

Open Client Utilities

- In IQ 16, these Open Client utilities are installed:
  - ISQL
  - DSEDIT
  - DSCP (UNIX only)
  - OCSFG (Windows only)

ISQL

- ISQL is a command-line Interactive SQL utility that uses the Adaptive Server Enterprise Open Client API.
- For syntax and parameters, see the "SAP Sybase ASE 15.7 Utility Guide".
- ISQL doesn't permit you to create user-defined database options. If you need to add your own database options, use the dbisql Interactive SQL utility instead.

Interactive SQL (DBISQL)

DBISQL

- This is IQ's interactive SQL Utility. It enables you to execute SQL statements, build scripts, and display database data.
- DBISQL has two different interface modes:

  1. A graphical interface that is sometimes referred to as "Interactive SQL Java".

  2. A command-line interface that is sometimes referred to as dbisql.

DBISQL-Java vs. DBISQL-C

- Within IQ, there are two editions of the dbisql utilities, each is based on a different programming platform:
  1. **DBISQL-C:** This is the first edition of DBISQL and it was built on C. It is now considered the "legacy" version of dbisql. This version is sometimes referred to as dbisql-C, dbisql-Classic or Interactive Dbisql-Classic.

  2. **DBISQL-Java:** This is the most recent version of DBISQL and it was built on Java. This version was designed to replace the C-based version of dbisql.

DBISQL-C, DBISQL-Classic, And Interactive ISQL-Classic Have Been Deprecated

- The C-based edition of dbisql (including the DBISQL-C, DBISQL-Classic, and Interactive SQL Classic versions of dbisql) has been deprecated in IQ 16.
- However, there are no plans to remove this C-based edition. Interactive SQL Classic is provided for backwards compatibility for running SQL scripts and as a lightweight tool for deployment.
- It is important to note: This C-based edition does NOT provide the same functionality as that Java version of DBISQL provides, and it does NOT support all the features in the current version of SAP Sybase IQ.

DBISQL-Java Is The New Standard

- Given that the C-based version of DBISQL has been deprecated and does not support the latest features of IQ, SAP strongly recommends that you use the Java edition of the dbisql utility, rather than DBISQL-C, or ISQL, unless instructed otherwise; or if it is required for a specific task.
- Support for the C-based version is only provided for backwards compatibility with existing applications and scripts.

DBISQL-Java Features

Below, is a brief description of the two versions of DBISQL-Java.

Interactive SQL (Java-Based)

- This is the graphical version of IQ’s dbisql utility.
- To start the Interactive SQL version of dbisql, choose the “Interactive SQL Java” utility from the SAP Sybase IQ 15 program group on Windows.
- The Windows client version of this utility can be installed in several ways: It is installed as part of the Windows IQ server install, as part of the IQ Client install, or it can be installed using the Sybase Open Client SDK.
- You can also run Interactive SQL Java using an xterm window on UNIX and Linux systems.
  To do this, you would type the following within your SAP Sybase IQ environment on UNIX:
  `xterm -sb`

Command Line-Mode (DBISQL-Java)

- On UNIX, the dbisql utility can also be run as a command line interface.
- To start this version of dbisql, type “dbisql -nogui” at the UNIX command prompt within your IQ environment on UNIX. You also need to specify the IQ server name, database name, server port, server account and password. For example, to connect to the IQ utility server, you might enter the following at the UNIX command prompt:

```
dbisql -c „uid=DBA;pwd=sql; eng=my_server; dbn=my_db; links=tcpip{port=2638}“ –nogui
```

Returning Multiple Result Sets In DBISQL

By default, Interactive SQL shows the first result set of the most recently executed statement. To return multiple result sets in DBISQL, use the following procedures.

Windows GUI

- If you are using the graphical version on Windows platforms, do the following:
  - Click Tools > Options in the Interactive SQL window.
  - The Interactive SQL Options window appears.
    - On the Sybase IQ page, select the Results tab:
      - Choose Show all Result Sets.
      - Click OK.

UNIX Command Line

- If you are running DBISQL as a command line program (-nogui mode) on UNIX, do the following:
  - Navigate to your $HOME directory and locate the file .isqlPreferences12.
  - Change this line: **`<entry key=“SybaseIQ.showMultipleResultSets“>0</entry>`**
  - To this line: **`<entry key=“SybaseIQ.showMultipleResultSets“>1</entry>`**
  - Save your changes to **.isqlPreferences11**

UNIX GUI

- If you are running dbisql in GUI mode on UNIX
  - Click Tools > Options in the Interactive SQL window.
  - The Interactive SQL Options window appears.
  - On the Sybase IQ page, select the Results tab.
  - Choose Show All Result Sets.
  - Click OK.

DBISQL-Java Changes

Tip: Do Not Mix SQL Dialects

- The language parser in dbisql has changed considerably since it was first released.
- Do NOT mix Watcom-SQL and Transact-SQL dialects when executing batch scripts in IQ 16, otherwise you could get errors like:

```
Syntax error near ‚PROCEDURE‘ on line 6

SQLCODE=-131, ODBC 3 State=“42000“
```

Interactive SQL Restrictions

- Do not use the Import option in the dbisql Data menu (or Command | Options | Input Format in: dbisqlc). This option is not supported for use with IQ databases. Use the LOAD TABLE statement or the INSERT statement to load data into IQ tables.
- If you set the terminal type on UNIX and Linux systems to "dumb" or "unknown" and then start dbisqlc, SAP Sybase IQ returns an error. For example:
  - Executing this:

```
% setenv TERM dumb
% dbisqlc
```

  - Will return this error:

```
error at line 1
Unable to initialize screen routines
```

- To avoid problems, run dbisql (Interactive SQL Java) instead, or use an xterm window to run dbisqlc on UNIX and Linux systems. For example, you can start an xterm window with a scroll bar as follows: `% xterm –sb`

Network Connectivity

Client Access API'S

- In the past, client access to SAP Sybase IQ was limited to JConnect (SAP's implementation of JDBC), ODBC, and Open Client.
- Starting from IQ 15, there have been a number of new data accesses APIs. These include the following:
  - ADO.NET
  - Perl (provided by the SAP Sybase SQL Anywhere Perl DBD::SQLAnywhere DBI module)
  - Python (provided by the sqlanydb interface)
  - PHP (provided by the SAP Sybase SQL Anywhere PHP module)
  - OLE DB
  - Native Ruby Driver
  - Ruby/DBI Driver
  - Linux32(x86) platforms
- These new APIs make it easier to build and deploy database applications in multiple programming environments.

Connectivity Tips

- Source code, sample projects, and OS-specific binaries for Perl, Python, and PHP are installed in the **"%IQDIR16%\SDK"** directory on Windows and the **"$IQDIR16/sdk"** directory on UNIX.
- ADO.NET and OLEDB code samples are in the appropriate folder within **"%ALLUSERSPROFILE%\SybaseIQ\samples\SQLAnywhere".**
- Most Open Client applications expect an SAP Sybase ASE server and T-SQL behavior, as opposed to ANSI-SQL. Therefore, when writing stored procedures, make sure you consider the type of client connections will execute your code. Some language options use different default values depending on whether you are using an Open Client connection versus an ODBC connection.
- These options include:
  - ALLOW_NULLS_BY_DEFAULT
  - QUOTED_IDENTIFIER
  - STRING_RTUNCATION
  - ANSI_BLANKS
  - ANSINULL
  - CHAINED
  - FLOAT_AS_DOUBLE
- When using an ODBC driver, it is a good idea to set AutoPreCommit to "YES". With this setting, user queries will obtain the latest version of a database object. You can set this at the connection level or as a server option.

- An ODBC connection will set the following temporary options upon connection:

  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';
- In order to override these settings for your ODBC connection, you must set the desired values in the "InitString" of the ODBC DSN.
- For more details, see "Utility Guide" (dbdsn).

ODBC Driver Manager On UNIX

- unixODBC driver manager needs to be version 2.2.14 or higher.
- For more details see: "Programming: The unixODBC driver manager".
- SAP Sybase IQ now provides SA's libdbodxx shared object. In IQ 16, it is called libdbodbc16. This can be used on all supported UNIX platforms as an ODBC driver manager.
- For more details see:
  - For ODBC samples: "Programming: ODBC Samples".
    - For creating an ODBC.INI file: "Administration: ODBC Data Sources on UNIX".

Network Connectivity

- Here are some suggestions for increasing network throughput when accessing an IQ database:
  - Network: Data retrieval depends on the speed of the network. The faster the network cards and LAN, the better concurrency you can achieve.
  - Packet Sizes: Use the –p server option to set the appropriate packet size. A larger packet size allows a larger data set to be transmitted in each packet.
  - Default packet size without –p will be 7300 bytes

Data Modeling Recommendations

This section provides guidance on tuning your database schema for SAP Sybase IQ.

Proper Data-Type Sizing

- Use the smallest data types possible for data.
- If time information is not necessary, then use DATE instead of DATETIME.
- If data can fit in TINYINT (1-byte) or SMALLINT(2-byte), then use it instead of INTEGER(4-byte) or BIGINT(8-byte).
- Don't over allocate storage when defining NUMERIC() or DECIMAL().
- Don't specify CHAR() or VARCHAR() larger than expected maximum length of data.

IQ Unique

- IQ 16 FP_NBIT_IQ15_COMPATIBILITY='Off' is the default. MINIMIZE_STORAGE will be ignored.
- Using IQ UNIQUE(0) will disable all FP or n-bit optimizations and force a flat FP to be used. It is not recommended to use this setting.
- Use the IQ UNIQUE option to force a specific cardinality on a column.
- The following apply only when FP_NBIT_IQ15_COMPATIBILITY='ON' (you want to maintain FP(1), FP(2), and FP(3) indexing instead of adoping new NBit dictionary compression):
  - It is good idea to set the Minimize_Storage option to ON before table creation. This allows IQ to optimize the FP indexes for each column.
  - Use the IQ UNIQUE option to force a specific cardinality on a column.
  - Using the Minimize_Storage option will place an IQ UNIQUE(255) on every column for every table created. and will remove the need to use SYBASE IQ UNIQUE.
  - Although, Minimize_Storage slightly slows down data loads, it improves query speeds drastically.

NULL Values

- It is good idea to specify NULL or NOT NULL for a column, as it allows the optimizer to better guess the join criteria.
- Unlike ASE, NULL data does not save space on the database page.

- Always use NULL or NOT NULL for columns in stored procedures including temporary tables if tds and cmdseq are used to create those stored procedures.

Unsigned Data Types

- Use unsigned data types where ever possible.
- Comparisons of unsigned data are faster than signed data.

LONG VARCHAR and LONG BINARY

- WD, TEXT and CMP indexes are the only indexes that are allowed on VARCHAR() data wider than 255 bytes. The TEXT index is the only supported index for LONG BINARY columns.
- Storage for these data types are allocated in 256 byte chunks.
- Note that you must have the Unstructured Data Analytics Option license to use data types of LONG VARCHAR or LONG BINARY.
- Maximum width of Varchar is 32K -1 (64K -1 ASCII hex for Varbinary).
- The maximum width of Long Varchar is 2GB.
- The word index is the only index that allows varchar() data wider than 255bytes.
- There are some Sybase IQ functions that return LONG VARCHAR types (REVERSE, SUSER_NAME, UCASE and others). If you use these functions with a SELECT INTO statement, and you don't have the Unstructured Data Analytics Option license, you should use the CAST statement to convert the return value of the function to the correct data type and size.

Large Object Storage

As a separately licensed option, SAP Sybase IQ supports character large object (CLOB) data with a length ranging from zero (0) to 512TB (terabytes) for an SAP Sybase IQ page size of 128KB or 2PB (petabytes) for an SAP Sybase IQ page size of 512KB. The maximum length is equal to 4GB multiplied by the database page size.

- Large objects can be stored in binary or text based objects.
- This option extends long binary data type from a maximum size of 2GB to an unlimited size.
- Indexes you can use on LOB are FP, WD and TEXT only.
- Some functions of value are byte_length64 (returns size of an object) or byte_substr64(returns portions of the object).

VARCHAR vs. CHAR

- One major feature of IQ 16 is variable storage for VARCHAR and VARBINARY. Unlike previous versions, it is recommended that VARCHAR types be used when possible
- VARCHAR types add a slight storage overhead. For example, a VARCHAR(100) with just 25 bytes in use will require 26 bytes of storage: 25 bytes for the variable data, and 1 byte for the size of data.
- CHAR data is blank padded, VARCHAR is not.

When to Use Indexes

- If you have join columns, then you should have HG indexes.
- All searchable columns should either have HG or LF indexes.
- Aggregation columns should have HNG indexes. Note that SUM(A*B) will use an FP instead of an HNG.
- DATE, TIME and DATETIME columns should have DATE, TIME or DTTM indexes.
- If uncertain, place an LF or HG index on the column, depending on cardinality.
- Use Primary Key, Unique Constraint or UNIQUE HG index where appropriate.
- Do NOT have an HNG index on date/time/datetime columns. Replace it with DATE, TIME or DTTM index.
- If a column is used for word searching, then place a WD index on it.
- If a column is used for "full text" searching, then place a TEXT index on it.
- Group By's can take advantage of multi-column indexes as long as the index completely matches the column and order.
- HG inserts are the most expensive in IQ with respect to the other indexes.
- Integer comparisons are quicker than character comparisons; hence it is preferable to have join columns on integer datatypes (unsigned if possible).
- Keep data types as narrow as possible to improve performance by reducing disk I/O and memory requirements.
- Multi-column primary keys should have an additional LF or HG index placed on each individual column.

Temporary Tables

- When you use ‘On Commit Preserve Rows’ on temporary tables, then rows remain in the table after the transaction has been committed.
- Temporary tables are available at the current level (parent) and all of its children.
- A parent cannot see a child’s temporary table.
- Global temporary tables are static across connections and reboots.

Cursors

- Using cursors generally means row based processing and this is not optimal in IQ.
- IQ is designed for set based processing.
- If cursors are used, then make sure to use NO SCROLL cursors.
- ‘Open with Hold’ means the cursor will remain open across transactions. If they are not used, then the cursor is closed when a commit is issued.

New Security Features

RBAC Authorization

SAP Sybase IQ 16 provides RBAC Authorization as part of the Base Product Security features. It gives full control over which system privileges and roles can be granted to each entity. It enforces the separation of duties principle by making it possible to assign disjointed sets of privileged tasks to separate autonomous entities.

Role-based security is based on the concept of system and object-level privileges and roles. A privilege controls the ability to perform a specific task and can be granted directly to a user or to a role. Roles can then be granted to users.

Each system privilege or role has designated administrators whose responsibility it is to control which users are granted the system privilege or role, and which users can act as administrators.

LDAP Authentication

LDAP authentication is part of the Advanced Security Option, and can now be integrated with SAP Sybase IQ 16 to use any existing enterprise-wide directory access framework. It can be used for user authentication using searched distinguished names.

To protect against a server failure, you can create a primary and a secondary LDAP server configuration object which can be used for failover and failback purposes as needed.

LDAP authentication provides an efficient and secure design for frequent, short-lived connection that can scale to multiple domains and LDAP servers. Some LDAP features include:

- Authentication using searched distinguished name (DN)
- Failover to a secondary LDAP server for high availability
- Automatic failback to previously failed servers
- Integration with OpenLDAP third-party libraries
- Secure communication with LDAP servers
- Efficient design for frequent, short-lived connections
- Extensibility to multiple domains and multiple LDAP servers

Kerberos Authentification

Like IQ 15, Kerberos authentication is still available on IQ 16. It is a single sign-on system for distributed environments that allows you to maintain a single user ID and password for database connections, operating system, and network logins. After initial authentication, it eliminates the need to transmit passwords over the network.

The main component of Kerberos is the Key Distribution Center which holds all the keys, authenticates users, and generates tickets for authorized users to use services in the environment.

Database Maintenance Recommendations

Backup and Recovery

This section includes some recommendations for backup and recovery:

- It is a good idea to have your database schema in a file format in case you need to rebuild the server should a database restore not be possible. Any major changes in Schema should be followed by proper up-keeping of DDL soft copy.
- Backup the database when there are major changes in the database as well as at regular intervals. Backups are extremely important for continued operations in case of unforeseen events.
  - Keep copies of the valid database in safe location(s). IQ (latest versions) allows you to verify the database dumps. Verification will inform you of any issues, so you can take corrective action in a timely manner.
  - Always keep a good copy of the catalog .db file. In a Multiplex environment, make sure you have a good copy of the coordinator's catalog .db file.
  - In order to reduce the time for a backup, it is recommended that you investigate the virtual backup option provided with IQ. This allows you to do more frequent full backups, since they don't take as long as a regular backup.
- Keep monitoring the .log file size. If it grows too big then take appropriate actions to truncate the .log file. An extremely large .log file can have an adverse impact on database performance.
- IQ (latest versions) has configurable options (-iqmsgsz, -iqmsgnum switches) for keeping the message file size constrained, and making copies of previous versions of the file.
- It is ideal to have a test environment as similar as possible to the production environment. Any modifications should be tested in the test environment for correctness and desired performance before implementing those changes in the production environment.
- For any IQ related errors, contact Primary Support as soon as possible, before taking any drastic action. The support staff can guide you properly to avoid major problems.
- Create events in the IQ server that monitor OS space availability. Use the sa_disk_free_space system stored procedure to get this information.
- In order to maintain IQ availability, it is important that you monitor the available disk space for IQ's critical operational files. This includes the IQ .log and .db files, along with the .iqmsg file and several other files. There must be sufficient space to accommodate these files in order to avoid interruptions in IQ service.
- On UNIX platforms, the /tmp file system is often used for temporary work. This directory also needs sufficient space. Before starting an IQ server IQTMP environment variable can be set to point to different temporary location.

Monitoring The Main DBspace And Temp DBspace Usage

This section describes several stored procedures and events that will enable the monitoring of main and temporary DBspace use, and to drop connections that consume too many resources.

Stored Procedures For Monitoring DBspace

The sp_dropConnOnMainUsed and sp_dropConnOnTempUsed stored procedures:

1. Once the free space within the main or temp DBspace falls below a specified threshold value, the stored procedure "sp_dropConnOnMainUsed" identifies the connection consuming the most space, and the stored procedure "sp_dropConnOnTempUsed" drops this connection.
2. Note: sp_dropConnOnMainUsed will not drop DBA connections for Main Store. However, sp_dropConnOnTempUsed will drop any connection (including DBA connections) that is using too much space in the Temp store.
3. Whenever a connection is dropped, these stored procedures will write to their respective log files: 'dropConnOnMainUsed.log' and 'dropConnOnTempUsed.log'.
4. Unless explicit path names are passed to these stored procedures, the default location for their logs is the same directory where the database (*.db) file is located.

How To Use The “dropConn” Stored Procedures For Monitoring DBspace

To implement this monitoring capability, do the following:

1. Create the stored procedures in your database from a DBA connection using the procedure code in the following sections:

   a. sp_dropConnOnMainUsed

   b. sp_dropConnOnTempUsed

2. Create events in the database using a DBA connection:

```
CREATE EVENT „dropConnOnMainUsed“
SCHEDULE „dropConnOnMainUsed“ START TIME ‚04:00‘ EVERY 300 SECONDS
HANDLER
   BEGIN
             call sp_dropConnOnMainUsed(‚dropConnOnMainUsed.log‘, 90)
   END;

CREATE EVENT „dropConnOnTempUsed“
SCHEDULE „dropConnOnTempUsed“ START TIME ‚04:00‘ EVERY 300 SECONDS
HANDLER
   BEGIN
       call sp_dropConnOnTempUsed(‚dropConnOnTempUsed.log‘, 90)
   END;
```

3. For Multiplex environments, you will want to ensure that the events are only executed from a writer node. Add the following lines before calling the stored procedures to generalize the events for a Multiplex:

```
declare srvType char(1);
select ServerType into srvType from sp_iqmpxversioninfo();
//Return immediately if Reader Node.
if srvType=‘Q‘
then
return ;
end if ;
```

Note: The parameters to be configured are the following :

- **Start time** --> Set your preferred start time. In this example, it was 04:00.
- **Frequency of calling the stored procedure** --> Set your preferred frequency.
  In this example, it was
  300 seconds (5 min) in the above example.
- **Threshold value** --> This is the Free Space percent (Not Used Space). In the above example, it was 90%.

Starting from IQ version 15.3, IQ is capable of distributed query processing. For this purpose IQ needs a “Shared” temporary store (dbspace). When you use IQ version 15.3 or higher an empty dbspace will be created for you called IQ_SHARED_TEMP. Add disk space (raw device) to this dbspace using command Alter Database Add file or use SCC to add the disk space.

Stored procedure sp_iqmpxcheckdqpconfig **can be used to diagnose DQP configuration issues**.

Many times messages about one or more particular dbspaces out of space get buried in a large .iqmsg file, it might be prudent to write a script to scan .iqmsg file for such messages. SAP primary support has a template for such a script which can be obtained from them for UNIX platforms. It needs to be modified per your environment and use it at your own risk.

Message Log Management Using –iqmsgsz and –iqmsgnum

- There are two new message log management parameters called -iqmsgsz and –iqmsgnum, starting form IQ 15.x version.
- The new -iqmsgsz and –iqmsgnum parameters allows you to control the size of the message log file, and the number of archives to be saved.

Tip: Use The Startup Switches To Limit The Size Of The Message Log

- Use the -iqmsgsz and -iqmsgnum switches to prevent the message log file from growing indefinitely.

Monitoring The Catalog's Disk Space

- When you create new objects in IQ, their definitions are stored in the IQ catalog. Over time, the catalog file (".DB") and the catalog log (".LOG") files will grow in size.

Tip: Monitor The Size Of ".DB" and ".LOG" Files

- Always monitor the OS-level disk space where your ".DB" and ".LOG" files reside. You must ensure there is sufficient space to accommodate these files in order to avoid interruptions in IQ's availability.

Monitoring Free Space Using The New "sa_disk_free_space" Stored Procedure

- IQ 15.x introduced a new system procedure for monitoring the free space for system DBspaces called "sa_disk_free_space". This procedure returns the system DBspace name, the free disk size and the total disk size.
- The system DBspace names that can be used with the sa_disk_free_space syntax are:
  - SYSTEM
  - TRANSLOG
  - TRANSLOGMIRROR
  - TEMPORARY or TEMP

Tip: Monitor Your Free Space

- Be sure to monitor your IQ DBspaces frequently by running the "sa_disk_free_space" procedure at regular intervals throughout the day.
- Run it either manually, by scheduling its execution in a CREATE EVENT statement, or in an UNIX script to monitor free space. For example:

```
 call sa_disk_free_space (system);      --> returns disk info for the system
                                             DBspace (.DB)
call sa_disk_free_space (translog);    --> returns disk info for the transaction
                                             log  (.LOG).
call sa_disk_free_space (temp);        -->returns disk info for the disk where
                                            temporary files reside.
```

Auditing

- With the auditing feature turned on, the following additional data is saved in the transaction log:
  - All login attempts (successful and failed), including the terminal ID.
  - Accurate timestamps of all events (to a resolution of milliseconds).
  - All permissions checks (successful and failed), including the object on which the permission was checked (if applicable).
  - All actions that require DBA authority.

Tip: Enabling Auditing

- To enable auditing for all connections and all types of operations, log in as DBA and run:

```
      SET OPTION auditing = ‚On‘;
  CALL sa_enable_auditing_type( ‚all‘ );
```

- Combine this with truncating the transaction log (e.g. each day at 12.00). This will enable you to collect the audit data on a per day basis.
- For details on all the auditing types, see: "sa_enable_auditing_type system procedure".

Temporary Files

- Starting from IQ 15.x, there are now more options to control and monitor temporary files.

Tip: Manage Your Temp Files

- On UNIX/Linux platforms, the locations of the temporary files created by IQ can be controlled by pointing the IQTMP16 variable to the desired directory (with the desired permissions).
- You can also use the "-dt" server startup switch to specify the location of a temporary file. If this option is not specified, then the database server will check IQ's environment variables, in the following order:
    1. IQTMP16 (or the SATMP environnement variable for SQL Anywhere's temporary files)
    2. TMP
    3. TMPDIR
    4. TEMP

Database Connection Enhancements

- Temporary connections are now named starting from IQ 15.4. These connections are used internally to perform operations such as running backups, initializing databases etc. Users can get information on these connections by executing the "sa_conn_info" and the "sa_conn_list" system stored procedures.
- Support for the "escape" connection has been added to the ODBC data sources starting from IQ 15.4.
- Starting from IQ 15.4, you can now assign a customized name to the ODBC driver. This facilitates installation and registration of multiple independent copies of the SQL Anywhere ODBC driver on a client system.

Scan The Log For SySAM Messages

- SAP strongly recommends that you create a process to programmatically scan the IQ log at regular intervals for SySAM messages and take the appropriate action. These messages can occur at any time and are the result of a wide range of issues ranging from software installation to increasing the size of DBspaces. Failure to correct SySAM warnings or errors in a timely manner may result in the server shutting down or failing to boot.

IQ Database Options That Help Optimize Performance

The following lists several options that can improve performance.

The "QUERY PLAN" Option

- In general, turn on the QUERY_PLAN option only when needed to collect query and load plans for analysis purposes. Keeping this option ON will grow the .iqmsg file size at a rapid rate.

The "FORCE_NO_SCROLL_CURSORS" Option

- Make sure the "FORCE_NO_SCROLL_CURSORS" option is turned ON. In rare situations where you need backwards scrolling cursors, you can turn it off.

The "default_disk_striping" Option

- For best performance, make sure "default_disk_striping" is set to ON in order to allow IQ to write to all available disk stripes during a write operation.

The "xp_cmdshell" Option

- If possible, avoid the use of "xp_cmdshell" calls within IQ because these calls can be expensive.

The "PACKETSIZE" Option and the "INSERT...LOCATION" Operation

- For better performance when executing an INSERT...LOCATION operation, increase the communication packet size to 4K if the data volume is large (the default is 512 bytes on most platforms). For example: INSERT LOCATION PACKETSIZE 4096. Before doing this however, be sure to verify that the remote ASE (sp_configure default network packet size/ max network packet size) can handle this packet size.

Backup and Restore Tips

Backup Methods

- The IQ Server provides several types of backup:
  - **Full Backup:** This is the default backup type. It performs a full backup of catalog stores and all used blocks.
  - **Incremental Backup:** Performs a full backup of the catalog store and all blocks that have changed since the last backup of any kind are backed up.
  - **Incremental-Since-Full-Backup:** Performs a full backup of the catalog store and all blocks that have changed since the last full backup. Note: The incremental clause is not supported with READ-ONLY DBspaces.
  - **Virtual Backup:** Performs a full backup of the catalog store and selected IQ metadata: The information specific to the freelist, the backup and checkpoint is backed up.
  - **Virtual Decoupled:** Performs a virtual backup, where all DBspaces are copied after the decoupled backup finishes, followed by a non-virtual incremental backup.
  - **Virtual Encapsulated:** A virtual encapsulated 'shell command' allows the shell command to be executed as part of the backup. The shell command performs system-level backup of the IQ store. A non-zero status returned from the shell command indicates a backup failure and the virtual backup returns an error.

Backup and Restore Tips

- Use Virtual Backup: The whole database can be backed up and restored in seconds/minutes, minimizing downtime. SAN technologies provide the capability of creating multiple mirrored copies of IQ DBfile devices. These copies make it possible to offload maintenance tasks, such as consistency checks, on the mirrored copies. The system stored procedure sp_iqfile can be used to create a backup list of files that comprise the database.
- Backup and restore commands may only be executed on the coordinator in a Multiplex environment.
- In an IQ backup, the IQ temporary store and params.cfg are not backed up, but information needed to recreate the IQ temporary store and metadata is backed up. The DBA should make a copy of the params.cfg file and save the contents of the system views: "SYSDBFILE" and "SYSDBSPACE".
- A DBspace is composed of a set of DBFiles. When creating DBFiles, use logical links to make it easier to move the underlying physical devices as needed.
- DBspaces and DBFiles can be Read-Only (RO), Read-Write (RW), and online or offline. You may restrict full, incremental-since-full or incremental backup to just the set of read-write files in the IQ main store. All read-write files will be backed up. An IQ backup may backup a subset of read-only DBspaces and/or read-only DBFiles in the IQ main store.
- In IQ 16, 20% of IQ_SYSTEM_MAIN is reserved for the freelist. The freelist is used to track the blocks used by DBspaces. It is considered part of the catalog information. Used blocks for the freelist are backed up, but reserved blocks are not since they can be easily reconstructed without a backup of the information.
- During a restore, if the directory structure and DBfile names are the same in both the DR environment and the production environment, there is no need to drop the secondary servers from the Multiplex.
- The "restore" command provides a "verify" option to validate the database backup archives. This option only reads the blocks and does not perform a restore.
- In a restore operation, the IQ store files (.iq), catalog database (.db file) and transaction log (.log file) must not exist in the location to which the database is being restored. If any of these files exist, they must be moved to a different location prior to starting the full restore.

Backup Verification And Diagnostics

IQ 16 includes several utilities that provide information about the backup and restore activities performed against an IQ database:

- **db_backupheader:** Reads the backup archive and displays the DBspaces and DBFiles that existed when the backup was done.
- **sp_iqbackupsummary:** Shows all the DBFiles included in a particular backup.
- **sp_iqbackupdetails:** Summarizes the backup operations performed by a particular backup.
- **sp_iqrestoreaction:** Lists the restore actions needed to bring a database to a consistent state for a given date.

Diagnostics Checklist For Troubleshooting Problems In IQ

This section describes information you should collect before sending message to SAP Primary Support.

Diagnostics for All Problems

Collect the following diagnostic information for all IQ problems:

- Provide a clear problem description and/or exact error message:
  - Is this Simplex (Single node) or Multiplex (Multiple Node)?
  - If Multiplex, which node: coordinator, reader(s) or writer(s)?
  - The time of the problem
  - Can the problem be reproduced?
  - Is it a non-fatal stack trace, a hung server, or a server crash situation?
- Has anything changed recently on IQ or the OS?
- Is this problem occurring on a Production Server?
- Provide the IQ log files - Please collect log files from the last clean boot sequence:
  - .iqmsg (This file usually is in the same directory where .db file resides)
  - .srvlog (This file is in IQ installs, logfiles directory)
  - .stderr (This file is in IQ installs, logfiles directory)
  - –zo output file after setting "–zr all" or "-zr sql"
  - Provide IQ configuration file, .cfg
- Provide the Operating System (OS) information: version, patch level, total memory, and number of CPUs/cores.
- Provide the OS error logs. For example /var/adm/messages or "errpt" output based on the operating system where IQ is installed.
- Provide the complete IQ version output. Execute: **"start_iq –v2"**
- Provide the version of any other tool involved in the problem, e.g. dbisql, OCS, JDBC, SCC, ODBC, jConnect, etc., and any logs pertaining to these tools.

Diagnostics For Slow Server Performance

In addition to the diagnostics that should be collected for any IQ problem, collect the following additional diagnostics for troubleshooting slow server performance:

1. IQ Server-Level Diagnostics For Slow Server Performance

Collect the following server level information for slow server performance:

- The .cfg file used to start the IQ server.
- Provide the IQ log files — Please collect log files from the last clean boot sequence:
  - .iqmsg
  - .srvlog
  - .stderr
  - –zo output file after setting "–zr all" or "-zr sql"
- The exact command used to start the IQ server.
- Using dbisql or isql, provide the output from the following:
  - sp_iqstatus
  - select * from sysfile
  - select * from sysiqdbfile
  - select * from sysoptions order by 2,1
  - Monitoring output (Script for using isql):

```
create table iqmontable (c1 int)
go
iq utilities main into iqmontable start monitor '-interval 20'
go
// run the above for a minute
iq utilities main into iqmontable stop monitor
go
```

```
iq utilities private into iqmontable start monitor '-debug –interval 20'
go
// run the above for a minute
iq utilities private into iqmontable stop monitor
go
drop table iqmontable
go
```

- **Note**: IQ will create two output files with "iqmon" in the filename. They will be located in the same directory where the .db file resides.

2. Operating System Diagnostics For Slow Server Performance

Collect the following additional output from the following UNIX commands for slow server performance:

- netstat –s:
- netstat –m
- pstack <process_id>
  - **Note:** For simplex a server or coordinator and/or secondary node(s) showing problems, gather at least 3-4 "pstack" outputs that are at least 3-5 minutes apart. The pstack utility may differ based on the operating system that IQ is running on. For example, on AIX the tool is 'procstack'; on Solaris/Linux/HP_UX it is 'pstack'.
- Memory map:
  - On most UNIX systems, the command is: pmap <PID>
  - On AIX the command is: 'svmon –rP'
- iostat –cdDex 10
- vmstat –S 5
- mpstat 180 10
- Provide tracing output:

Note: Use these tools with caution and ONLY as a last resort as they can severely impact performance:

- truss (AIX, Solaris etc.) = dtrace (Solaris 10+, preferred tracing option) = strace (Linux) = tusc (HP-UX)
- truss –afv all –Dd –o <output_file_name> -p <PID>
- **Note**: Execute the truss command for at least 40 seconds. Use this ONLY as a last resort as it might further slowdown the server and in some situations might not provide good results.
- strace –f –tt –o <output_file_name> -p <PID>
- tusc -o <output_file_name> -c -f -l –p <PID>
- tusc -o <output_file_name -ccc -f -l -n -p -v -T „%H:%M:%S" –p <PID>
- In some cases, 'sar' output might be required.

Diagnostics for Slow Query Performance

In addition to the diagnostics collected for any IQ problem, collect the following information for troubleshooting slow query performance:

- Which client tool is being used to execute the query? For example dbisql, isql, etc.?
- Which operating system is the client tool running on?
- What protocol is being used?
- Was the query scripted (i.e. stored procedure and/or written and tested earlier), generated or an ad-hoc query?
- What is the SQL for the query?
- Collect output from sp_iqindexmetadata
- What is the DDL definition for all the objects involved in the query, such as:
  - Tables
  - Views
  - Indexes
  - User defined data types

- Generate HTML Query and Load plans for good and bad performance periods by setting the following options just prior to executing the query:

```
set temporary option Query_Plan = ‘ON’;
set temporary option Query_Plan_After_Run = ‘ON’;
set temporary option Query_Detail = ‘ON’;
set temporary option Query_Timing = ‘ON’;
set temporary option Query_Plan_As_HTML = ‘ON’;
set temporary option Index_Advisor = ‘ON’;
set temporary option Query_Name = ‘<Query_name>’;
set temporary option DML_Options10 = ‘ON’;

set temporary option Query_Plan_As_HTML_Directory = ‘<html_plans_directory>’;
set    temporary option index_advisor=‘on‘ ;
```

Note: If you do not set Query_Plan_As_HTML_Directory, then by default the html plans will get generated in the same location where the .iqmsg file is created.

Diagnostics for Non-Fatal Stacktraces

In addition to the diagnostics collected for any IQ problem, collect the following information for troubleshooting a non-fatal error with a stack trace:

- The SQL or program that caused the stack trace.
- Did the SQL or program work fine before?
- Can the problem be reproduced on other servers?
- Can the issue be reproduced consistently?
- Is there anything unique occurring when the issue happens?
- Is there any particular sequence that needs to be followed to reproduce the problem?
- Is there any dependency on the data used during the stacktrace?
- If reproducible problem, can SQL, table DDL and possibly sample data can be sent to SAP for further diagnosis?

Diagnostic or a Crash or Fatal Stacktrace

In addition to the diagnostics that should be collected for any IQ problem, collect the following diagnostics for troubleshooting a crash or fatal stack trace:

- Can the issue be reproduced in a simpler repro?
- Can the server be started normally after the crash?
- Provide the stack trace file: stktrc*.iq
- Does this issue result in data corruption?
- If there is corruption, execute sp_iqcheckdb on the objects involved. For example:

  ```
  sp_iqcheckdb ‘verify database/table/index’
  ```

For a crash we also need in some situation the COMPLETE core file. By default IQ does not write entire core file, it truncates the core file to 32k. To get complete core file you need to set environment variable BEFORE starting IQ,

export IQ_UNLIMIT_CORE=”YES”

Depending upon the Operating system, you might need to set some operating system parameters to write complete core at desired directory.

Diagnostics For A Hung Server

In addition to the diagnostics collected for any IQ problem, collect the following information for troubleshooting a server hang:

- Check to see if any of the DBspaces ran out of space.
- Collect at least 3-4 pstack output(s) against the IQ server at least 3-5 minutes apart:
  - On Solaris, Linux and HP_UX the utility is: ‘pstack’
  - On AIX the utility is: ‘procstack’

Diagnostics For A Connection Issue

In addition to the diagnostics collected for any IQ problem, collect the following additional information for troubleshooting a connection issue:

- Can you ping the server using dbping or tcpip's ping utility?
- Are the connectivity problems specific to a particular client tool?
- Collect the client application logs:
  - ODBC client – turn on odbc trace.
  - Open Client – try pinging via dsedit. Setup RIBO and trace.
- Ensure the server name cache info has the correct server information. Locate sasrv.ini file. This file contains server information, including server name, protocol, and address. The default location of sasrv.ini is "**%ALLUSERSPROFILE%\Application Data\SQLAnywhere 11**" on Windows and "**~/.sqlanywhere11**" on UNIX.
- Try running a network trace such as: "tcpdump".

Diagnostics For A Backup And Resore Issue

In addition to the diagnostics collected for any IQ problem, collect the following additional information for troubleshooting backup and restore problems:

- Execute the restore utility using the "verify" option to verify that the backup is good.
- Check if any of the dump file is missing or the restore sequence is different than dump sequence.

DiagnosticsFor SySAM Issues

In addition to the diagnostics collected for any IQ problem, collect the following additional information for troubleshooting SySAM problems:

- Collect the output of sp_iqlmconfig.
- Collect the ".lic" file if this is an "unserved" license.
- Verify that directory for this ".lic" file is correctly identified in the search path written to the .iqmsg log. For example:

```
I. 04/12 12:30:18. 0000000000 Using licenses from:

C:\IQ\SYSAM-2_0\licenses\SybaseIQ.lic;C:\Documents and Settings\All Users\SybaseIQ\demo\*.
lic
```

- Provide the SySAM server log if it is served license. Check SySAM documentation on troubleshooting tips.
- On Linux machines get complete data as, "more /proc/cpuinfo > cpuinfo_output". On Solaris run /usr/sbin/prtdiag or /usr/sbin/psrinfo . On AIX run, lscfg or lsconf. On HP/UX use machinfo command
- dbname>.lmp file (this is located in the same directory where the .db files resides).

User Defined Functions

User-defined functions are user-written procedures that return a value to the calling environment for use in a SQL query or statement. They provide a mechanism for extending the functionality of a database server by allowing users to create their own custom functions that can be called from within a SQL statement.

UDFs can run either within the IQ server or external to the IQ server. IQ provides three types of UDFs:

1. External C/C++ UDFs that run within the IQ server and provide the highest level of performance – require the IQ_UDF license (for partner certified libraries), or the IQ_IDA license (for user-written libraries)
2. External Language UDFs that run outside of IQ and are written in languages such as Java, PERL, or client-side embedded C/C++ - do not require a special license
3. Interactive SQL UDFs that run within IQ (limited to the functionality provided by the SQL languages supported by IQ, and may lack the performance of external C/C++ UDFs) – do not require a special license

UDF Programming Practices

UDF Do's For External C/C++ UDFS

- If a UDF needs to be modified at runtime, SAP recommends following a set of procedures for safely unloading the old UDF library and reloading the new library. See "User-Defined Functions Guide" for more information.
- The C/C++ code for UDFs is compiled into one or more external libraries that are subsequently loaded into the IQ server's process space when needed. For simplicity of managing the UDF installation, Sybase recommends that UDF developers package multiple UDF functions within a single library.
- Although you can still use the v3 UDF API (requires either the IQ_UDF or IQ_IDA license) in IQ 16, it is recommended that you upgrade to the v4 API. This version allows you to distribute queries incorporating UDFs across a SAP Sybase IQ multiplex. Also, TPFs (table parameterized functions) in v4 allow you to define a partitioning schema that breaks the data set into row set partitions that execute in parallel across the multiplex.
- In 16, the execution of UDFs on a simplex server or on a multiplex writer node remains restricted to SAP Certified Partners and requires the purchase of another license — the IQ_UDF license.
- UDFs which run under the IQ_IDA option are restricted to reader nodes only. When running IQ_IDA UDFs, you must ensure that your failover logic NEVER results in a reader node being promoted to a writer, or a coordinator node. This promotion will cause all subsequent UDFs to fail.
- UDF developers must be careful with memory management and the use of the pointer. Be sure to manage memory and temporary results as defined by the API.
- In addition, be sure to implement robust error handling. Write UDFs so that they track whether a query has been canceled by the user, and clean up and exit properly. Check the return status of all callback functions, and return the correct status of success or failure from the UDF.
- Use log and console messages sparingly while debugging the C/C++ or external language code. All debugging output should be removed or disabled when compiling the final production UDF library, jar, etc., leaving only log or console messages which report error conditions.
- Write all UDFs in a thread-safe manner that allows them to be called simultaneously by different users. If a UDF accesses a global or shared data structure, implement the appropriate locking around its accesses to that data.
- Read the "User-Defined Functions Guide" for practices to avoid, such as performing complex, or memory-intensive operations that are repeated every invocation.

UDF Don'ts For External L C/C++ UDFs

- Do not hard-code library paths in SQL registration scripts. This practice makes it difficult to provide flexibility to the user to install the UDFs into the same directory as Sybase IQ.
- Do not write ambiguous code, or constructs that can unexpectedly loop forever, without providing a mechanism for the user to cancel the UDF invocation. See the function 'get_is_cancelled()'.
- Do not perform complex, or memory-intensive operations that are repeated every invocation. When a UDF call is made against a table that contains many thousands of rows, efficient execution becomes paramount. SAP recommends that you allocate blocks of memory for a thousand to several thousand rows at a time, rather than on a row-by-row basis.
- Do not open a database connection, or perform database operations from within a UDF. All parameters and data required for UDF execution must be passed as parameters to the UDF.
- Do not use reserved words when naming UDFs.
- Do not log excessive, detailed information to the iqmsg file (C/C++ UDFs) or console (external language UDFs), especially within loops or other segments of code which execute frequently. This will dramatically affect the performance of the UDF.
- Certified SAP partners who deploy their UDFs on a writer node need to do so with great caution. Use the secured feature (-sf) startup option with each server to enable or disable the execution of UDFs. However, in general, SAP strongly recommends that UDFs are installed and executed on read-only multiplex nodes only.

UDF Performance and Tuning Tips

Here are a few performance tips for in-process table UDFs:

- Avoid data copies by using fetch_block instead of fetch_into.
- The size of a row block structure can have an impact on performance. The number of rows in a row block is configurable using the
- TABLEUDF_ROW_BLOCK_CHUNK_SIZE_KB option. This option fixes the memory usage for the server constructed row block. If a row block is too small, the UDF will need to be invoked many times to return all the data. The number of rows in the row block will be chunk_size / row_width. (The row_width is the size in bytes for each column in a row). The default value is 128K bytes.

- For Java table UDFs, we recommend the following:
  - Set the temporary option "default_table_udf_row_count" to the number of rows you expect your UDF to produce. This affects how much space IQ thinks it needs to process the query, and will affect the type of join preference that IQ chooses. The default value for default_table_udf_row_count is 200,000.
  - Also, for Java UDFs in general, disallow "ALLOW SERVER SIDE REQUESTS". If they are allowed, then Java UDFs will execute via "functional compensation", which will negate potential performance gains that have been achieved between IQ 15.3 and IQ 15.4. The default setting is to disallow server side requests.
  - Turning off any console log messages (commenting out Java "sys.println(…)" statements) used while debugging the Java code.

UDF Debugging

- Use the "external_UDF_execution_mode" database option to control the amount of error checking and call tracing that is performed automatically when statements are evaluated.
- SAP Sybase IQ also captures errors from the operating system to the .stdout (Windows and UNIX) and .stderr (UNIX only) files located in the same directory as the database file. Note that logging messages to the iqmsg or console negatively affects performance, so all debug messages should be removed prior to final compilation and deployment to a production environment.
- Enable memory tracking to help you locate memory leaks in your UDFs. SAP Sybase IQ will help you detect memory leaks by tracking memory allocations, and will log a message to the user message log file that a particular UDF is leaking memory. Note that tracing and memory tracking impose some performance penalties, so you will want to turn these off in production environments.
- The UDF APIs also provide callback functions enabling the UDF to write to the .iqmsg message file, so you can log your own information as appropriate.
- Microsoft Visual Studio 2008 developers can use the Microsoft Visual Studio Debugger to step through the user-defined function code.

Database Options

Purpose Of Database Options

- Database options provide a method for controlling many aspects of IQ's behavior.
- Unlike SAP Sybase ASE, IQ provides hundreds of database options that enable you to customize IQ's behavior to meet your application needs.
- Most database options are available to all users and are dynamic. However, some options set permanent server-wide behavior and are static. For these options, DBA authority is usually required, and a reboot of the server is needed for these changes to go into effect.
- Since IQ contains only one database, changing a single database option can affect the entire server. Depending on how you qualify an option, the scope of these changes ranges from temporarily affecting an individual user, to permanently affecting all users on the server.
- Therefore, database options are an important and powerful feature of IQ. They must be fully understood before use and should be implemented with caution. Further, some database options should only be used under explicit instructions from SAP Technical Support.

Range of Behavior

Database options control the following types of behavior:

Compatibility

- Controls the extent to which the Sybase IQ behaves similar to SAP Sybase Adaptive Server Enterprise and whether errors are generated when SQL does not conform to SQL92.

Error Handling

- Controls what happens when a specified error occurs, such as dividing by zero.

Concurrency Transactions

- Controls the degree of concurrency and the details of COMMIT behavior.

Performance and Optimizer Behavior

- Controls performance features such as index usage and optimizer tips.

Query Behavior

- Enables or disables specific features and the "forcing" of specific query behavior.
- Resource Usage
- Controls resource usage such as cache, and execution time.

Diagnostics

- Controls the creation and configuration of troubleshooting and performance diagnostics.

How To Set A Database Option

To SET an option, use the SET OPTION statement:

Syntax

```
SET [ EXISTING ] [ TEMPORARY ] OPTION
... [ userid. | PUBLIC. ]option-name = [ option-value ]
```

Examples

- SET temporary option Query_Plan_As_HTML = 'ON';
- SET option DBA.Query_Plan = 'ON' ;
- SET option PUBLIC. Query_Plan = 'ON' ;

How To Reset Or Delete A Database Option

To RESET or DELETE a database option, set the option WITHOUT a value. This will return the option to its original or default value, depending how the options was originally set:

Examples

- SET temporary option Query_Plan_As_HTML = ;
- SET option DBA.Query_Plan = ;
- SET option PUBLIC. Query_Plan = ;

Rules For Resetting Database Option Values

- When an option is set WITHOUT a value, the option setting is changed as follows:
- If the option value was a personal option setting, the value reverts back to the PUBLIC setting.
- If the option value was a TEMPORARY option, the setting reverts back to a PERMANENT setting.

How To Control Database Options

Given the range and degree of control that database options exercise, the use of these options must be precisely controlled. To support this precision, the SET OPTION statement includes syntax that limits the scope, duration and precedence of its effects.

Option Scope

- There are three levels of scope: PUBLIC, USER AND TEMPORARY:
  - **SET PUBLIC:** This is a server-wide setting. It is for DBA use only, and it affects all users.
  - **SET User-id:** This only affects the user-id named in the option. NOTE: If a user-id or PUBLIC keyword is NOT specified, then by default, the option applies only to the user issuing the command.
  - **SET TEMPORARY:** This option remains in effect for the duration of the user connection that set the option.
  - **SET TEMPORARY PUBLIC:** This option remains in effect until IQ is restarted (requires the DBA authority).

Option Duration

- There are several duration levels of duration: PUBLIC, TEMPORARY, TEMPORARY PUBLIC (TEMPORARY-UNTIL-THE-NEXT-RESTART) and PERMANENT:
  - **PUBLIC:** Is a permanent server-wide option.
  - **TEMPORARY:** Is effective immediately and persists until changed or a user's session ends.
  - **TEMPORARY PUBLIC:** Is for DBA use only. It is effective immediately and persists until the IQ server is restarted. Upon restart, the option reverts to its previous permanent setting.
  - **PERMANENT:** This option Is NOT set explicitly. It is set indirectly by specifying an option without the TEMPORARY qualifier. It applies to the user or group that sets it. Or, if set by the DBA, it applies to ALL users.

Precedence

- TEMPORARY takes precedence over USER and PUBLIC settings.
- USER takes precedence over Public.

Dynamic vs. Static Options

- Options that affect the entire server may require the server to be restarted before taking affect.
- Options that affect only the current connection generally take place immediately. In many cases, you can change these option settings in the middle of a transaction.

Tip: Do Not Change Options When A Cursor is Open

- Do not change options when a cursor is open since this can lead to unreliable results.

Tips On Setting DB Options

Set User

- To set an option for a particular user or group, specify the user name or group name. **For example:** SET option USER1. Query_Plan = 'ON' ;
- To set an option for current user, specify the option without specifying a user name or group name. **For example:** SET option Query_Plan = 'ON' ;

Set Public

- To set an option for every user, use the PUBLIC keyword. For example: SET option PUBLIC.mpx_autoexclude_timeout = '0' ;
- **Note:** The use of the PUBLIC option requires DBA privileges.

Set Temporary (Without The Public Option)

- To set an option for the duration of the current user's connection only, set the option using the TEMPORARY keyword, without specifying PUBLIC or group. **For example:** SET temporary option Query_Plan = 'ON' ;

Set Public Temporary

- If an option is set to TEMPORARY for the PUBLIC group, it remains in effect for as long as the database is running and reverts back to the permanent value when the server is restarted. **For example:** SET PUBLIC. temporary option Query_Plan = 'ON' ;

Permanent (An Implied Setting)

- The permanent setting is implied. It is NOT explicitly set. That is, there is no "SET PERMANENT" command for database options. To make an option setting PERMANENT, set the option
  WITHOUT using the TEMPORARY key word. This sets the option value permanently for the user or group issuing the statement. **For example:** SET option Query_Plan = 'ON' ;

Using sp_iqcheckoptions To Display Your Database Options

The sp_iqcheckoptions procedure lists the current and default values for database options that have been CHANGED from the default:

- When sp_iqcheckoptions is run as DBA, it lists all options set on a permanent basis for all groups.
- When sp_iqcheckoptions is run as a user, it lists temporary options set for DBA and those temporary options set by the current user.
- All users see non-default server start-up options.
- Sample Output for sp_iqcheckoptions:

```
User_name    Option_name             Current_value  Default_value  Option_type
----------   -----------             ------------   -------------  -----------
‘DBA’        ‘Query_Plan_As_HTML’      ‘On’             ‘Off’    ‘Permanent’
‘DBA’        ‘Query_Plan_Text_Access’ ‘On’              ‘Off’    ‘Temporary’
‘DBA’        ‘Query_Plan_Text_Access’ ‘On’              ‘Off’    ‘Permanent’
```

Tip: Run spiqcheckoptions Before And After An Upgrade To Verify Option Settings

- Running sp_iqcheckoptions will enable you to verify that your post-migration option values are correct.
- This is critical, since in some cases, options may be reset to their default values as a result of the upgrade process. Therefore, you will need to identify these options and return them their desired values as needed.

Database Options That Affect Performance

For better performance, DBAs may want to change the following db options from their default settings:

Scrollable Cursor

- SET OPTION public.Force_No_Scroll_Cursors=‘on’;
  - By default, all query results are buffered in Temp Cache) to permit scrolling back and forth through all result rows. If the result set has millions of rows, this may seriously degrade performance. Therefore, it is recommended to disable this option (by setting it **“ON”**).
  - This option may be set OFF for those users that require this functionality.

Query Temp Space

- SET OPTION public.Query_temp_Space_Limit=‘0’;

If set to a non-zero value it specifies the number of Megabytes that each query can use. A query will fail to execute with the error *‘Query rejected because it exceeds total space resource limit’* if the temp space estimate exceeds this value.

Compatibility Settings

- When writing stored procedures or embedded application code, be sure to EXPLICITLY specify settings for compatibility as these options will get set to different values for Open Client vs. ODBC connections.

Table Load and Unload Settings

- **`DISABLE_RI_CHECK = “On”;`**
  - When loading data from ASE or other sources, you may want to disable referential integrity checking by turning this option **“ON”**.

ROW_COUNT =

- Unlike SAP Sybase ASE, this option **ONLY** controls the number of rows returned from a SELECT statement. It has NO effect on (does not limit) the size of the UPDATE or DELETE commands.

Database Options For Tuning Parallelism

- To help manage the additional memory requirements of parallelism in IQ 16, IQ provides several database options that allow you to either allocate more memory or to reduce parallelism to match the resources of your particular machine. Therefore, in addition to the existing startup settings that specify the size of main cache and temporary cache (-iqmc and –iqtc), IQ 16 has several options for tuning parallelism. These include the following:

max_query_parallelism:

– This option sets an upper bound which limits the level of parallelization the optimizer will permit query operators such as joins, GROUP BY, and ORDER BY to be. This value should be greater than or equal to the number of cores.

max_iq_threads_per_team:

– This option controls the number of threads allocated to perform a single operation. It should be at least twice the number of cores.

max_iq_threads_per_connection:

– This option controls the number of threads for each connection. It should be at least twice the value for max_iq_threads_per_team.

Database Options For Collecting Query Performance Data

Tip: Set the Following Database Options to Collect Query Plan Information:

```
• SET temporary option DDL_Information = ‘ON’ ;
• SET temporary option DML_Options10 = ‘ON’;
• SET temporary option Index_Advisor = ‘ON’;
• SET temporary option Query_Plan_After_Run = ‘ON’;
• SET temporary option Query_Detail = ‘ON’;
• SET temporary option Query_Name = ‘query name here’;
• SET temporary option Query_Plan = ‘ON’ ;
• SET temporary option Query_Plan_As_HTML = ‘ON’;
• SET temporary option Query_Plan_As_Html_Directory = ‘directory/path‘;
• SET temporary option QUERY_PLAN_TEXT_ACCESS = ‘ON’ ;
• SET temporary option Query_Timing = ‘ON’ ;
• SET temporary option QUERY_PLAN_MIN_TIME Option = <num milliseconds>
```

Note:

- If Query_Plan=”ON” is set globally, the IQ MSG file will grow quickly.
- To reduce output size, you can set Query_Plan_As_HTML without setting Query_Plan.

New and Changed Database Options for IQ 16

Below, is a listing of the database options that were either added or whose defaults were changed in IQ 16. Also, see the following appendix for a listing of database options that were deprecated in IQ 16.

For more details on these options, see the following documents:

1. For IQ-Level Options: “Reference: Alphabetical list of Options”.

2. For SA-Level Options: “SAP Sybase SQL Anywhere 12.0.1 Database Administration: Alphabetical list of Options”.

Listing Of Deprecated Options For IQ 16

Below are the options that were removed in 16:

LARGE_DOUBLES_ACCUMULATOR – replaced by FLOATING_POINT_ACCUMULATOR

MAX_WARNINGS – Since SAP Sybase IQ 16 no longer supports JOIN INDEXES, the MAX_WARNINGS option is no longer required

The following options have been depreciated in IQ 16:

APPEND_LOAD – still exists, but will be ignored

DQP_ENABLED login policy option – replaced by DQP_ENABLED logical server policy option

Listing Of New Options For IQ 16

Below, is a listing of the database options that were added in IQ 16.

AFFINITY_AUTOEXCLUDE_TIMEOUT and CACHE_AFFINITY_PERCENT

Description: For DQP eligible queries, SAP Sybase IQ identifies preferred nodes for scans of specific row ranges and partition IDs. Once a query establishes an affinity for a piece of data, subsequent queries that need the same data use the same node for their work. Use these options to control allocation.

AFFINITY_AUTOEXCLUDE_TIMEOUT Option

The amount of time in minutes before SAP Sybase IQ removes a shutdown node from the affinity map and reassigns its partitions to other nodes.

Allowable Values: 0 to 10080 minutes (1 week)

Default: 10

CACHE_AFFINITY_PERCENT

This option defines the percentage of the buffer cache used for affinitized data buffers. SAP Sybase IQ buffer caches are organized as a long MRU/LRU chain. Non-affinitized data buffers are put into the chain after affinitized buffers when this percentage is non-zero, so that affinitized data stay in the cache longer than non-affinitized data. If there are insufficient affinitized data buffers to fill this entire percentage, non-affinitized data may consume the remainder.

Note: Before changing this option, check the value of the WASH_AREA_BUFFERS_PERCENT option. WASH_AREA_BUFFERS_PERCENT affects the LRU side of the buffer cache and CACHE_AFFINITY_PERCENT affects the MRU side. The total of these two values cannot exceed 100 percent.

Allowable Values: 0 to 100 (percent)

Default: 70

ALLOW_SNAPSHOT_VERSIONING

Description: Applies to all base tables in the database (as opposed to RLV-enabled tables only). Restricts table versioning for all base tables to either table-level or row-level snapshot versioning. This option does not apply to the IQ catalog store.

Default: 'YES'

BASE_TABLES_IN_RLV

Description: This option turns row-level versioning on and off. When set to ON, tables are registered in the RLV store. RLV-enabled tables are optimized for real-time updates.

The **{ ENABLE | DISABLE } RLV STORE** clause of the **CREATE_TABLE** statement always overrides the **BASE_TABLES_IN_RLV_STORE** option.

Allowed Values: 'OFF'|'ON'

BLOCKING and BLOCKING_TIMEOUT

BLOCKING

Description: When BLOCKING is off, a transaction receives an error when it attempts a write operation and is blocked by the read lock of another transaction.

When BLOCKING is on, any transaction attempting to obtain a lock that conflicts with an existing lock held by another transaction waits until every conflicting lock is released or until the blocking_timeout is reached. If the lock is not released within blocking_timeout milliseconds, then an error is returned for the waiting transaction.

Allowed Values: 'ON'|'OFF'

Default: 'OFF'

BLOCKING_TIMEOUT

Description: When the blocking option is on, any transaction attempting to obtain a lock, which conflicts with an existing lock, waits for the indicated number of milliseconds for the conflicting lock to be released. If the lock is not released within BLOCKING_TIMEOUT milliseconds, an error is returned for the waiting transaction.

Set the option to 0 to force all transactions attempting to obtain a lock to wait until all conflicting transactions release their locks.

Allowed Values: 0 to 4294967295 (milliseconds)

Default: 0

CREATE_HG_WITH_EXACT_DISTINCTS

Description: determines whether the database engine creates HG indexes as single HG or tiered HG:

- If CREATE_HG_WITH_EXACT_DISTINCTS='ON', all subsequent HG indexes explicitly created with a **CREATE INDEX** command or implicitly creating or altering a table with a PRIMARY KEY or a FOREIGN KEY declaration, will be non-tiered HG indexes.
- If CREATE_HG_WITH_EXACT_DISTINCTS='OFF', all subsequent HG indexes explicitly created with a **CREATE INDEX** command or implicitly creating or altering a table with a PRIMARY KEY or a FOREIGN KEY declaration, will be tiered HG.

This option is ON by default in all newly created 16.0 databases, and all 16.0 database upgraded from IQ 15.x. To take advantage of the new tiered structure, set this option to OFF. Use **sp_iqrebuildindex** to convert non-tiered HG indexes to tiered HG and vice-versa.

DQP_ENABLED

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

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

Invalid setting for option 'DQP_ENABLED'

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

Allowed Values: 'OFF'|'ON'

Default: 'ON'

DQP_ENABLED_OVER_NETWORK

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

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

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

ENABLE_ASYNC_IO

Description: Allows a DBA to enable or disable the asynchronous IO used by the RLV persistence log for row-level versioning.

Allowed Values: 'TRUE'|'FALSE'

Default: 'TRUE'

FLOATING_POINT_ACCUMULATOR

Description: Controls which accumulator to use for **SUM** or **AVG** of floating-point numbers.

Setting 1 (fast accumulator) is faster and uses less space for floats and doubles than setting 2. This setting uses a single double precision variable to add double and float numbers, and is subject to the known accuracy limitations of such an approach.

Setting 2 (default) (medium accumulator) uses multiple double precision variables to accumulate floats and doubles. It is very accurate for addends in the range of magnitudes 1e-20 to 1e20. While it loses some accuracy outside of this range, it is still accurate enough for most applications. Setting 2 allows the optimizer to choose hash for faster performance more easily than setting 3.

Setting 3 (large accumulator) is highly accurate for all floats and doubles, but its size often precludes the use of hash optimization, which will be a performance limitation for most applications.

Allowed Values: 1 to 3

Default: 2

FP_NBIT_IQ15_ COMPATIBILITY

Description: The FP_NBIT_IQ15_COMPATIBILITY option provides tokenized FP support similar to that available in SAP Sybase IQ 15. All newly created and modified tokenized FP indexes in 16.0 will be NBit. The only 15 style FP(1),FP(2), and FP(3) byte FP indexes available in 16.0 are those from an upgraded database that have had only Read-Only activity.

The FP_NBIT_IQ15_COMPATIBILITY ON/OFF setting only pertains to tokenized FP creation and cut-off behavior:

If FP_NBIT_IQ15_COMPATIBILITY='ON', the database engine:

- Enables the MINIMIZE_STORAGE, FP_LOOKUP_SIZE, FP_LOOKUP_SIZE_PPM options
- Creates DATE datatypes as NBit FP, even if IQ UNIQUE(0) is specified
- Rolls over to Flat FP at the 3 byte FP cutoff (16,777,216 distinct)
- Can tokenize data widths <= 255

If FP_NBIT_IQ15_COMPATIBILITY='OFF':

- MINIMIZE_STORAGE, FP_LOOKUP_SIZE, FP_LOOKUP_SIZE_PPM options are ignored.
- DATE datatypes are not automatically NBit
- Data widths <=32767 may be tokenized
- NBit FP (tokenized) upper bound limit is NBit 31 (2,147,483,648 distinct values)
- NBit sizing options determine rollover behavior:
  - IQ UNIQUE(0) loads a column as Flat FP
  - Columns without IQ UNIQUE load as NBit up to the auto-size limits
  - Columns where IQ UNIQUE(n) is less than the auto-size limit load as NBit

Important: The FP_NBIT_IQ15_COMPATIBILITY option provides tokenized FP support similar to that available in SAP Sybase IQ 15, not complete database compatibility. All SAP Sybase IQ 15 runtime behavior is available using the 16.0 interface. Avoid running a database created with16.0 database withFP_NBIT_IQ15_COMPATIBILITY='ON'.

FP_LOOKUP_SIZE

Description: If FP_NBIT_IQ15_COMPATIBILITY='ON', FP_LOOKUP_SIZE controls the maximum number of lookup pages.

FP_LOOKUP_SIZE must be set public, so the allowed syntax is:

SET OPTION public.FP_LOOKUP_SIZE = 1

If FP_NBIT_IQ15_COMPATIBILITY='OFF' the database engine ignores this option.

Allowed Values: 1 to 4096 (MB)

Default: 16 (MB)

FP_LOOKUP_SIZE_PPM

Description: If FP_NBIT_IQ15_COMPATIBILITY='ON', FP_LOOKUP_SIZE_PPM controls the maximum number of lookup pages and restricts this number to a parts-per-million value of main memory, that is, the value of FP_LOOKUP_SIZE_PPM * size of main memory/ 1,000,000, where the size of main memory is specified by the **-iqmc** server startup parameter.

Allowed Values: 1 to 1000000

Default: 2500

The following FP_NBIT_* options apply only to databases where FP_NBIT_IQ15_COMPATIBILITY='OFF'. If FP_NBIT_IQ15_COMPATIBILITY='ON', the database engine ignores these options.

FP_NBIT_AUTOSIZE_LIMIT

Description: FP_NBIT_AUTOSIZE_LIMIT limits the number of distinct values in all newly created columns without an explicit IQ UNIQUE setting. Columns constrained by the FP_NBIT_AUTOSIZE_LIMIT option load with a Flat FP or NBit FP index:

- If FP_NBIT_AUTOSIZE_LIMIT is greater than 0 and less than 2,147,483,647, columns load with an NBit FP index
- If FP_NBIT_AUTOSIZE_LIMIT equals 0, columns load with a Flat FP index

FP_NBIT_AUTOSIZE_LIMIT and FP_NBIT_LOOKUP_MB establish a ceiling for sizing NBit columns during data loads. As long as the number of distinct values is less than FP_NBIT_AUTOSIZE_LIMIT and the total dictionary size (values and counts) is less the FP_NBIT_LOOKUP_MB, the column loads as an NBit. If the load exceeds the FP_NBIT_AUTOSIZE_LIMIT but is less than FP_NBIT_ROLLOVER_MAX_MB, the column rolls over to Flat FP.

DML operations that exceed the FP_NBIT_ROLLOVER_MAX_MB and FP_ENFORCE_LIMITS='ON' rollback and report an error. If the FP_NBIT_ENFORCE_LIMITS='OFF', the column transitions to the next NBit level. **sp_iqindexmetadata** returns details about Flat FP or NBit FP columns. **sp_iqrebuildindex** can change explicit or implicit NBit FP column limits, or reformat the default column index as Flat FP or NBit FP.

FP_NBIT_ENFORCE_LIMITS

Description: DML operations check the FP_NBIT_ENFORCE_LIMITS option when the number of distinct values in a column exceeds the explicit limit set in an IQ UNIQUE constraint, which is above the FP_NBIT_AUTOSIZE value, or when the dictionary size for an implicit **NBit** rollover exceeds the FP_NBIT_ROLLOVER_MAX_MB limit.

- If FP_NBIT_ENFORCE_LIMITS='ON', the DML operation throws an error and rolls back
- If FP_NBIT_ENFORCE_LIMITS='OFF', the DML operation continues and the NBit dictionary limits are ignored

sp_iqindexmetadata returns details about Flat FP or NBit FP columns. **sp_iqrebuildindex** can change explicit or implicit NBit FP column limits, or reformat the default column index as Flat FP or NBit FP. Using **sp_iqrebuildindex** to increase the number of distinct values beyond current limits for a Flat FP column when FP_NBIT_ENFORCE_LIMITS='ON', returns an error. If FP_NBIT_ENFORCE_LIMITS='OFF', **sp_iqrebuildindex** rebuilds the index to the maximum token, which is the largest distinct value.

Allowed Values: 'OFF'|'ON'

Default: 'OFF'

FP_NBIT_LOOUP_MB

Description: FP_NBIT_AUTOSIZE_LIMIT and FP_NBIT_LOOKUP_MB establish a ceiling for sizing implicit NBit columns. As long as the number of distinct values is less than FP_NBIT_AUTOSIZE_LIMIT and the total dictionary size (values and counts) is less the FP_NBIT_LOOKUP_MB, the column loads with an NBit FP index. Limits are enforced by the FP_NBIT_ENFORCE_LIMITS option.

DML operations that exceed the FP_NBIT_LOOKUP_MB limit rollover to a Flat FP index.

If an operation exceeds FP_NBIT_LOOKUP_MB and FP_NBIT_ROLLOVER_MAX_MB limits and FP_NBIT_ENFORCE_LIMITS=‘OFF‘, the NBit FP transitions to the next NBit level.

sp_iqindexmetadata returns details about Flat FP or NBit FP columns. **sp_iqrebuildindex** can change explicit or implicit NBit FP column limits, or reformat the default column index as Flat FP or NBit FP.

Allowed Values: 1 to 4,294,967,295 (MB)

Default: 64 (MB)

FP_NBIT_ROLLOVER_MAX_MB

Description: FP_NBIT_AUTOSIZE_LIMIT and FP_NBIT_LOOKUP_MB establish a ceiling for sizing implicit NBit FP columns. DML operations that exceed these values check the FP_NBIT_ROLLOVER_MAX_MB limit, which sets the dictionary size (values and counts) for implicit NBit rollovers:

- If the total dictionary size does not exceed the FP_NBIT_ROLLOVER_MAX_MB, the NBit column rolls over to a Flat FP.
- If the dictionary size exceeds the FP_NBIT_ROLLOVER_MAX_MB limit and FP_NBIT_ENFORCE_LIMITS=‘ON‘, DML operations throw an error and roll back.
- If the dictionary size exceeds the FP_NBIT_ROLLOVER_MAX_MB limit and FP_NBIT_ENFORCE_LIMITS=‘OFF‘ (default), DML operations throw an error, and the NBit dictionary continues to grow.
- If FP_NBIT_ROLLOVER_MAX_MB=‘0‘, the NBit column rolls over to Flat FP.

sp_iqindexmetadata returns details about Flat FP or NBit FP columns. **sp_iqrebuildindex** can change explicit or implicit NBit FP column limits, or reformat the default column index as Flat FP or NBit FP.

Allowed Values: 1 to 4294967295

Default: 16384

JOIN_SIMPLIFICATION_THRESHOLD

Description: The query optimizer simplifies its optimization of join order by separate handling of both lookup tables (that is, nonselective dimension tables) and tables that are effective Cartesian products. After simplification, it optimizes the remaining tables for join order, up to the limit set by **MAX_JOIN_ENUMERATION**.

Setting this option to a value greater than the current value for **MAX_JOIN_ENUMERATION** has no effect.

Setting this value below the value for **MAX_JOIN_ENUMERATION** might improve the time required to optimize queries containing many joins, but may also prevent the optimizer from finding the best possible join plan.

If you change this value, set the **JOIN_SIMPLIFICATION_THRESHOLD** as a temporary or user option, and to a value of at least 9.

Allowed Values: 1 to 64

Default: 16

LOG_DEADLOCKS

Description: When this option is set to On, the database server logs information about deadlocks in an internal buffer. The size of the buffer is fixed at 10000 bytes. You can view the deadlock information using the sa_report_deadlocks stored procedure. The contents of the buffer are retained when this option is set to Off.

When deadlock occurs, information is reported for only those connections involved in the deadlock. The order in which connections are reported is based on which connection is waiting for which row. For thread deadlocks, information is reported about all connections.

When you have deadlock reporting turned on, you can also use the Deadlock system event to take action when a deadlock occurs.

Allowed Values: 'OFF'|'ON'

Default: 'OFF'

MIN_ROLE_ADMINS

Description: This option sets the minimum number of required administrators for all roles. This value applies to the minimum number of role administrators for each role, not the minimum number or role administrators for the total number of roles. When dropping roles or users, this value ensures that you never create a scenario where there are no users and roles left with sufficient system privilege to manage the remaining users and roles.

Allowed Values: 1 to 10

Default: 1

QUERY_PLAN_MIN_TIME

Description: In previous releases, the QUERY_PLAN option was ON by default, meaning that every query generated a query plan regardless of the size of the query. In SAP Sybase IQ 16.0, use the new QUERY_PLAN_MIN_TIME option when QUERY_PLAN is ON to prevent the server from generating query plans for micro queries, resulting in a database server performance improvement.

Allowed Values: 0 to 4294967295 (milliseconds)

Default: 0 (off).

REVERT_TO_V15_OPTIMIZER

Default: SAP Sybase IQ 16.0 supports several new join and grouping algorithms that leverage Hash and Hash-Range partitioned tables, as well as a few other new algorithms. By default, all of these new algorithms are considered by the optimizer and will be selected where valid and appropriate. Setting REVERT_TO_V15_OPTIMIZER='ON' disables all 16.0 changes to the optimizer cost models. It also disables all of these new join and grouping algorithms, unless they are valid and are specifically requested via a positive value for either the AGGREGATION_PREFERENCE option, theJOIN_PREFERENCE option, or a join condition hint string.

The REVERT_TO_V15_OPTIMIZER option is normally used for internal testing and manually tuning queries. Only experienced DBAs should use it.

Note: An error will result if your query references an RLV-enabled table and REVERT_TO_V15_OPTIMIZER='ON'.

Allowed Values: 'OFF'|'ON'

Default: 'OFF'

ROUND_TO_EVEN

Default: When ROUND_TO_EVEN option is set to ON, the **ROUND** function rounds half to the nearest even number. When the option is set to OFF, the **ROUND** function rounds SAP Sybase IQ half away from zero.

SAP Control Center and Interactive SQL automatically turn the ROUND_TO_EVEN option OFF.

Example 1:

The ROUND function returns a different value based on whether ROUND_TO_EVEN is ON or OFF.

When ROUND_TO_EVEN is set ON, SELECT ROUND (convert(double, 0.25), 1) from iq_dummy returns 0.2.

When ROUND_TO_EVEN is set OFF, SELECT ROUND (convert(double, 0.25), 1) from iq_dummy returns 0.3.

RV_AUTO_MERGE_EVAL_INTERVAL

Description: This option is be used to configure the period of wait time, in minutes, between activations of the merge evaluator. The merge evaluator examines the merge parameters of each row-level versioning (RLV) enabled table against configured threshold values to determine whether a non-blocking (background) merge of the RLV table to IQ main stores should occur.

If the interval ends while the evaluator is active, or when a merge is already in progress, the interval re-sets.

Any new value for the interval is used when the merge evaluator is next activated.

Allowed Values: 1 to 4294967295 (Minutes)

Default: 15 (minutes)

RV_MERGE_NODE_MEMSIZE

Default: This option sets the percentage of total RLV memory size as a merge threshold for the node. If the total RLV memory size surpasses the threshold, the merge condition evaluator will determine which table(s) to merge. If multiple tables must be merged to satisfy the node threshold, parallel merges will be triggered for each table to be merged.

Allowed Values: 0 to 100 (Percent)

Default: 75 (Percent)

RV_MERGE_TABLE_MEMPERCENT

Description: This option sets the percentage of memory used as a merge threshold for an RLV-enabled table. If the memory used surpasses the threshold, a merge will occur.

The system evaluates whether to merge the RLV and IQ main stores on a per-table basis. It enumerates through all loaded RLV tables, and for each one decides whether a merge is warranted. A merge for a single table is deemed warranted if:

1. The table violates either the memory threshold (**RV_MERGE_TABLE_MEMPERCENT**) or the row threshold (**RV_MERGE_TABLE_NUMROWS**), and

2. The system does not determine that a large percentage of the RLV rows are uncom itted, and are therefore unable to be merged.

Note: When **RV_MERGE_TABLE_MEMPERCENT** = 0, then the system uses a (per-table) threshold of 100% / N, where N is the number of RLV-enabled tables that have been loaded.

RV_MERGE_TABLE_NUMROWS

Description: This option sets the number of rows used as a merge threshold for an RLV-enabled table. If the number of rows used surpasses the threshold, a merge will occur.

The system evaluates whether to merge the RLV and IQ main stores on a per-table basis. It enumerates through all loaded RLV tables, and for each one decides whether a merge is warranted. A merge for a single table is deemed warranted if:

1. The table violates either the memory threshold (RV_MERGE_TABLE_MEMPERCENT) or the row threshold (RV_MERGE_TABLE_NUMROWS), and

2. The system does not determine that a large percentage of the RLV rows are uncommitted, and are therefore unable to be merged.

Allowed Values: 1000 to 100000000

Default: 10000000

RV_RESERVED_DBSPACE_MB

Description: This option allows you to control the amount of space set aside in the RLV store for small but critical data structures used during release savepoint, commit, and rollback operations.

Allowed Values: 50 – 4294967295 (MB)

Default: The minimum of 50 Mb or half the size of the RLV dbspace

SNAPHOT_VERSIONING

Description: Applies to RLV-enabled tables only (as opposed to all base tables in the database). Controls whether RLV-enabled tables are accessed using single-writer table-level versioning, or multiple writer row-level versioning. This option does not apply to the IQ catalog store.

Allowed Values: 'row-level'|'table-level'

- **row-level:** Enables concurrent writer access and row-level versioning for RLV-enabled tables.

  The first transaction to modify a table row establishes a row write lock that persists until the end of the transaction.

  Subsequent transactions attempting to modify a locked row either fail with a lock/future version error, or block until the lock is released based on the value of the BLOCKING option.

- **table-level:** Enables single-writer access and table-level versioning.

  The first transaction to access the table establishes a table write lock which persists until the end of the transaction.

  Subsequent transactions attempting to write to a locked table either fail with a lock/future version error, or block until the lock is released based on the value of the BLOCKING option.

Default: 'table-level'

TRUSTED_CERTIFICATES_FILE

Description: This option identifies the path to the location of the list of trusted certificate authorities. The list must be stored in a TXT file. The file may be shared in a location in a Windows environment on the local drive to be used by all SAP Sybase applications on that machine.

Allowed Values: A valid network path to the location of a TXT file containing the list of trusted certificate authorities that sign server certificates.

Default: NULL, meaning that no outbound TLS connection can be started because there are no trusted certificate authorities.

Listing Of Enhanced Options For IQ 16

Below is a listing of existing database options whose functionality was enhanced in IQ 16.

AGGREGATION_PREFERENCE

Description: This option sets what aggregation behavior should be used for distinct/grouping.

Enhancement: The allowed value range is now -6 to 6.

These values are new in SAP Sybase IQ 16.0:

4 – prefer aggregation with a distinct/grouping sort.

5 – prefer aggregation with a sort if grouping columns include all the partitioning keys of a hash partitioned table.

6 – prefer aggregation with a hash if grouping columns include all the partitioning keys of a hash partitioned table.

-4 – avoid aggregation with a distinct/grouping sort.

-5 – avoid aggregation with a sort if grouping columns include all the partitioning keys of a hash partitioned table.

-6 – avoid aggregation with a hash if grouping columns include all the partitioning keys of a hash partitioned table.

Default: Not set

Allowed Values: = 6 to -6

JOIN_PREFERENCES

Description: Controls the choice of algorithms when processing joins.

Enhancement: In 15.4, this option was enhanced to apply to users with DBA authority. Unlike other users, DBA accounts automatically unlock 15 minutes after the last failed login attempt.

The allowed value range is now -12 to 12.

These values changed in SAP Sybase IQ 16.0:

- 6 – previously the action was prefer prejoin. The new action is prefers asymmetric sort merge join.
- -6 – previously the action was avoid prejoin. The new action is avoiding asymmetric sort merge join.

These values are new in SAP Sybase IQ 16.0:

- 8 – prefer asymmetric sort merge push down join.
- 9 – prefer partitioned hash join if the join keys include all the partition keys of a hash partitioned table.
- 10 – prefer partitioned hash-push down join if the join keys include all the partition keys of a hash partitioned table.
- 11 – prefer partitioned sort-merge join if the join keys include all the partition keys of a hash partitioned table.
- 12 – prefer partitioned sort-merge push-down join if the join keys include all the partition keys of a hash partitioned table.
- -8 – avoid asymmetric sort merge push down join.
- -9 – avoid partitioned hash join if the join keys include all the partition keys of a hash partitioned table.
- -10 – avoid partitioned hash-push down join if the join keys include all the partition keys of a hash partitioned table.
- -11 – avoid partitioned sort-merge join if the join keys include all the partition keys of a hash partitioned table.
- -12 – avoid partitioned sort-merge push-down join if the join keys include all the partition keys of a hash partitioned table.

Listing Of Options With New Behavior In IQ 16

MINIMIZE_STORAGE

Description: If **FP_NBIT_IQ15_COMPATIBILITY**=‘ON‘, **MINIMIZE_STORAGE** optimizes storage for new columns by using as little as one byte of disk space per row wherever appropriate. By default, this option is ‚OFF‘ for the PUBLIC role, and the specialized storage optimization does not occur for all newly created columns; when **MINIMIZE_STORAGE**=‘OFF‘ for the PUBLIC role but ‚ON‘ as a temporary user option, one-byte storage is used for new columns created by that user ID.

In SAP Sybase IQ 15.x databases, setting **MINIMIZE_STORAGE**=ON is equivalent to placing an **IQ UNIQUE 255** clause on every new column, with the exception of certain data types that are by nature too wide for one-byte storage. When **MINIMIZE_STORAGE**=‘ON‘, there is no need to specify **IQ UNIQUE**, except for columns with more than 65536 unique values.

When the ratio of main memory to the number of columns is large, turning **MINIMIZE_STORAGE**=‘ON‘ is beneficial. Otherwise, storage of new columns generally benefits by turning this option OFF.

Appendices

Appendix 1: sp_dropConnOnMainUsed Procedure

```
create procedure
//
// This procedure is to monitor the IQ server by checking DBspace usage and to prevent
// the IQ server from running out of main space.
//
// Information written by this procedure into the log is as follows:
//
// 1. Main Store
// 2. Temp Store
// 3. Versioning Space
//
// If usage of Main store reaches the specified threshold,
// IQ server disconnects the connection which holds the most Main space.
// Exceptions: DBA connections never dropped
dba.sp_dropConnOnMainUsed(LogFile varchar(50),FreeMainSpace integer)
begin
  declare maintotal unsigned bigint;
  declare mainused unsigned bigint;
  declare temptotal unsigned bigint;
  declare tempused unsigned bigint;
  declare databasename varchar(30);
  declare versionsize varchar(255);
  declare servername varchar(30);
  declare connname varchar(30);
  declare TempKB unsigned bigint;
  declare connuserid varchar(30);
  declare CurrTime varchar(30);
  declare MsgText varchar(255);
  declare connid integer;
  declare blocksizeX2 unsigned bigint;
  declare local temporary table m_iq_txn_table(
    TxnID unsigned bigint null,
    CmtID unsigned bigint null,
    VersionID unsigned bigint null,
    State char(12) null,
    TxnCreateTime char(26) null,
    ConnHandle unsigned bigint null,
    IQConnID unsigned bigint null,
    Dbremote bit not null,
    CursorCount unsigned bigint null,
    SpCount unsigned bigint null,
    SpNumber unsigned bigint null,
    MainTableKBCreated unsigned bigint null,
    MainTableKBDropped unsigned bigint null,
    TempTableKBCreated unsigned bigint null,
    TempTableKBDropped unsigned bigint null,
    MainWorkSpaceKB unsigned bigint null,
    )
    in SYSTEM on commit preserve rows;
  declare local temporary table iq_status_main(
    Name varchar(40) null,
    Value varchar(128) null,
    )
    in SYSTEM on commit preserve rows;
  set CurrTime=“left“(convert(varchar(30),getdate(*),115),16);
  execute immediate ‚iq utilities main into iq_status_main status‘;
  select substring(Value,cast(locate(Value,‘=‘) as tinyint)+1,length(Value)) into
versionsize
    from iq_status_main where name like ‚%Other%‘ order by Name asc;
  select Value into versionsize from iq_status_main where name = ‚Other Versions:‘;
  call sp_iqspaceused(maintotal,mainused,temptotal,tempused);
  set databasename=db_name(*);
  set servername=@@servername;
  // IQ main store free space > FreeMainSpace then return. If it is < FreeMainSpace
  // then drop the connection which is taking maximum main space.
  if cast(100-(mainused*100/maintotal) as integer) > FreeMainSpace then
    drop table m_iq_txn_table;
    drop table iq_status_main;
    return
  end if;
  select first block_size/512 into blocksizeX2 from SYSIQINFO;
  execute immediate ‚iq utilities main into m_iq_txn_table command statistics 10000‘;
  //Drop connection only when they could release some amount of space.
  select top 1 ConnHandle,
    connection_property(‚Name‘,connHandle) as Name,
    connection_property(‚Userid‘,connHandle) as Userid,
    max(cast(MainTableKBCreated*blocksizeX2/2 as unsigned bigint)+cast(MainTableKBDropped*
blocksizeX2/2 as unsigned bigint)) as MainWorkSpaceKB into connid,
    connname,connuserid,
    TempKB from m_iq_txn_table where
    MainTableKBCreated > 0 and
    Userid <> ‚DBA‘ and
    MainTableKBCreated > MainTableKBDropped
    group by ConnHandle order by
    MainWorkSpaceKB desc;
  if connid is not null then
    execute immediate ‚drop connection ‚ || connid;
    set MsgText=‘echo ‚ || CurrTime || ‚ IQ Main free space of: ‚ ||
      cast(cast(100-(mainused*100/maintotal) as numeric(5,2)) as varchar(10)) || ‚% ‚ ||
‚>> ‚ || LogFile;
    call xp_cmdshell(MsgText);
    set MsgText=‘echo ‚ || CurrTime || ‚ IQ Temp free space of: ‚ ||
      cast(cast(100-(tempused*100/temptotal) as numeric(5,2)) as varchar(10)) || ‚% ‚ ||
‚>> ‚ || LogFile;
    call xp_cmdshell(MsgText);
    set MsgText=‘echo ‚ || CurrTime || ‚ IQ Versioning Size : ‚ || cast(versionsize as
varchar(30)) || ‚>> ‚ || LogFile;
    call xp_cmdshell(MsgText);
    set MsgText=‘echo , || CurrTime || , DBA dropped Connection Handle : , || cast(connid
as char(10)) || ,, UserID : , ||
      cast(connuserid as char(10)) || ,. Used MainWorkSpaceKB : , || cast(TempKB/1024 as
numeric(10,2)) || , MB. >> , || LogFile;
    call xp_cmdshell(MsgText)
  end if;
  drop table m_iq_txn_table;
  drop table iq_status_main
end  ;
```

Appendix 2: sp_dropConnOnTempUsed Procedure

```
create procedure dba.sp_dropConnOnTempUsed(LogFile varchar(50),FreeTempSpace integer)
begin
  declare maintotal unsigned bigint;
  declare mainused unsigned bigint;
  declare temptotal unsigned bigint;
  declare tempused unsigned bigint;
  declare databasename varchar(30);
  declare versionsize varchar(255);
  declare servername varchar(30);
  declare connname varchar(30);
  declare TempKB unsigned bigint;
  declare connuserid varchar(30);
  declare CurrTime varchar(30);
  declare MsgText varchar(255);
  declare connid integer;
  declare blocksizeX2 unsigned bigint;
  declare local temporary table t_iq_txn_table(
    TxnID unsigned bigint null,
    CmtID unsigned bigint null,
    VersionID unsigned bigint null,
    State char(12) null,
    TxnCreateTime char(26) null,
    ConnHandle unsigned bigint null,
    IQConnID unsigned bigint null,
    Dbremote bit not null,
    CursorCount unsigned bigint null,
    SpCount unsigned bigint null,
    SpNumber unsigned bigint null,
    MainTableKBCreated unsigned bigint null,
    MainTableKBDropped unsigned bigint null,
    TempTableKBCreated unsigned bigint null,
    TempTableKBDropped unsigned bigint null,
    TempWorkSpaceKB unsigned bigint null,
    )
    in SYSTEM on commit preserve rows;
  declare local temporary table iq_status_temp(
    Name varchar(40) null,
    Value varchar(128) null,
    )
    in SYSTEM on commit preserve rows;
  set CurrTime="left"(convert(varchar(30),getdate(*),115),16);
  execute immediate ,iq utilities main into iq_status_temp status‘;
  select substring(Value,cast(locate(Value,‘=‘) as tinyint)+1,length(Value)) into
versionsize
    from iq_status_temp where name like ,%Other%‘ order by Name asc;
  select Value into versionsize from iq_status_temp where name = ,Other Versions:‘;
  call sp_iqspaceused(maintotal,mainused,temptotal,tempused);
  set databasename=db_name(*);
  set servername=@@servername;
  // IQ temp store free space > FreeTempSpace then return. If it is < FreeTempSpace
  // then drop the connection which is taking maximum temp space.
  if cast(100-(tempused*100/temptotal) as integer) > FreeTempSpace then
    return
  end if;
  set MsgText=‘echo , || CurrTime || , IQ Main free space of: , ||
    cast(cast(100-(mainused*100/maintotal) as numeric(5,2)) as varchar(10)) || ,% , || ,>>
, || LogFile;
  call xp_cmdshell(MsgText);
  set MsgText=‘echo , || CurrTime || , IQ Temp free space of: , ||
    cast(cast(100-(tempused*100/temptotal) as numeric(5,2)) as varchar(10)) || ,% , || ,>>
, || LogFile;
  call xp_cmdshell(MsgText);
  set MsgText=‘echo , || CurrTime || , IQ Versioning Size : , || cast(versionsize as
varchar(30)) || ,>> , || LogFile;
  call xp_cmdshell(MsgText);
  select first block_size/512 into blocksizeX2 from SYSIQINFO;
  execute immediate ,iq utilities main into t_iq_txn_table command statistics 10000‘;
  select top 1 ConnHandle,
    connection_property(,Name‘,connHandle) as Name,
    connection_property(,Userid‘,connHandle) as Userid,
    max(cast(TempWorkSpaceKB*blocksizeX2/2 as unsigned bigint)+cast(TempTableKBCreated*blo
cksizeX2/2 as unsigned bigint)+cast(TempTableKBDropped*blocksizeX2/2 as unsigned bigint))
as TempWorkSpaceKB into connid,
    connname,connuserid,
    TempKB from t_iq_txn_table
    group by ConnHandle order by
    TempWorkSpaceKB desc;
  execute immediate ,drop connection , || connid;
  set MsgText=‘echo , || CurrTime || , DBA dropped Connection Handle : , || cast(connid as
char(10)) || ,, UserID : , ||
    cast(connuserid as char(10)) || ,. Used TempWorkSpaceKB : , || cast(TempKB/1024 as
numeric(10,2)) || , MB. >> , || LogFile;
  call xp_cmdshell(MsgText);
  drop table t_iq_txn_table;
  drop table iq_status_temp

end ;
```

Appendix 3: SAP Sybase IQ Error Messages and Reporting Alerts

Error and status messages indicating conditions of potential concern to SAP Sybase IQ database operations can appear in several different IQ log files:

- the IQ message file (“the .iqsmg log”)
- the server log (“the .srvlog”)
- the standard error log (“the .stderr log”)
- the Request Log (“the zrzo log” or “SQL log”)

All four message sources should be monitored in order to catch all conditions of potential interest to a database administrator (DBA). There is no single means that a system generating DBA alerts can use to detect all error messages. This article examines the various types of SAP Sybase IQ messages and how they can be detected.

`.iqmsg log messages`

Standardized messages in the .iqmsg log

Prior to IQ 12.6, IQ error messages appearing in the IQ message file were identified mainly by source code module and line number within the module, along with message text. As a result of Change Requests implemented in IQ 12.6, SQLCode and SQLState were added, and a new Sybase IQ Error Messages Guide was published. This guide indexed error messages by SQLCode and SQLState so users as well as Sybase Technical Support could locate more information about errors found in the .iqmsg log. The “internal message number”— such as [20539] in the example that follows — was also prefixed to the message text for code module exception messages. Previously this number was only shown for non-exception reporting.

For example, what had been reported as:

```
Exception Thrown from opt_GroupNode.cxx:549, Err# 0, tid 69 origtid 69
  O/S Err#: 0, ErrID: 9216 (df_Exception)
  Feature, ROLLUP or CUBE in a subquery, is not supported.
-- (opt_GroupNode.cxx 549)
```

was reported instead as:

```
Exception Thrown from opt_GroupNode.cxx:549, Err# 0, tid 69 origtid 69
   O/S Err#: 0, ErrID: 9216 (df_Exception); SQLCode: -1001030, SQLState: ‘QFA2A‘,
Severity: 14
[20539]: Feature, ROLLUP or CUBE in a subquery, is not supported.
-- (opt_GroupNode.cxx 549)
```

All of this helped standardize IQ error message reports and make them more useful. Prior to these changes, SQLCode and SQLState were reported to the client only, along with the error message text. These changes in IQ 12.6 made it possible to correlate what the client received with the server-oriented error information logged in the IQ message file.

The ErrID field — above “9216” — is NOT a unique identifier for an error condition. It is an internal IQ identifier for the component reporting the error. This is why different messages can appear with the same “ErrID”: one component may detect multiple errors. These internal codes are not returned to the client. The other internal information including the O/S error code as displayed in the example above is likewise not returned to the client.

Some time ago it was requested that IQ error messages be reorganized to be like ASE messages. IQ Engineering replied that however good an idea that might be, it really wasn‘t possible for several reasons. One is that with IQ we are getting messages sometimes from the SQL Anywhere engine and sometimes from the IQ engine — different parts of the product. Also, IQ is written largely in C++ while ASE is written mostly in C, so the very way IQ produces or throws errors is different from ASE.

Error messages in the .iqmsg log are generally represented as shown above — an exception message generated by a code module including SQLCode, SQLState, Severity, internal message number and message text. The SQLCode is typically used as the primary identifier for referencing error message documentation or searching a knowledge base.

The IQ Error Messages guide includes several versions of reference tables for errors messages, sorted by SQLCode, SQLState, or "Sybase error code". Returning to the example we used above, we can look up this message:

```
Exception Thrown from opt_GroupNode.cxx:549, Err# 0, tid 69 origtid 69
   O/S Err#: 0, ErrID: 9216 (df_Exception); SQLCode: -1001030, SQLState: 'QFA2A',
Severity: 14
[20539]: Feature, ROLLUP or CUBE in a subquery, is not supported.
-- (opt_GroupNode.cxx 549)
```

by SQLCode -1001030, SQLState QFA2A, or "Sybase error code" 20539. The latter is what we previously called "internal error code".

SQLCodes can be positive as well as negative. Negative values are errors, and positive values are warnings.

SQLState is a system of codes defined by the ANSI SQL/92 standard. Each value is a 5-character string consisting of a 2-character class and a 3-character subclass. For more details including a table of class code definitions, see the SAP SYBASE IQ Error Messages guide topic "SQL State error codes".

Sybase error codes, the values in square brackets, may not always be unique. In such cases, the SQLCode value provides differentiation.

None of these by itself may be convenient for generating DBA alerts, since values are individualized. However, the Severity code could provide a starting point for detecting issues of concern for a DBA since it applies to ranges of SQLCodes. Individual SQLCodes or Sybase error codes could be used additionally to flag issues of specific concern. Operations DBAs looking for server-wide issues will likely be interested in different conditions than applications DBAs monitoring errors in business processing functions.

The IQ 15 Error Messages guide includes a table of Severity codes. It can be reached by clicking on the "Severity Code" value in any error message write-up, though it is not listed in the manual's Table of Contents. The IQ 16 "SAP Sybase IQ Error Messages" guide does list it

Severity Codes

Severity codes indicate the degree and source of the error.

Number	Description	Number	Description
0	Comment.	17	Insufficient resources.
1	Information.	18	Non-fatal internal error detected.
10	Status information; warning.	19	Fatal error in resource.
11	Specified database object not found.	20	Fatal error in current process.
12	Wrong data type encountered.	21	Fatal error in database processes.
13	User transaction syntax error.	22	Fatal error: table integrity suspect.
14	Insufficient permission to execute command.	23	Hardware error or system table corruption.
15	Syntax error in SQL statement.	24	Hardware error or system table corruption.
16	Miscellaneous user error (non-fatal).		

The Severity code can generally indicate how critical a problem may be so that it can be further examined. However, additional qualification may be required in order to make decisions on actions to be taken. Here are two examples of different kinds of issues both classified as Severity 14.

This message represents a serious server-level situation which may require quick DBA intervention:

```
Exception Thrown from hos_mem.cxx:370, Err# 1, tid 1503 origtid 1503
O/S Err#: 0, ErrID: 517 (hos_memexception); SQLCode: -1006042, SQLState: 'QBA42',
Severity: 14
[20280]: All available virtual memory has been used; allocation cancelled: [Extra info:
8392703]
-- (hos_mem.cxx 370)
```

This message, on the other hand, is a completely different type of issue that may or may not require any intervention:

```
Exception Thrown from db_iqutility.cxx:1929, Err# 105, tid 1194 origtid 1194
   O/S Err#: 0, ErrID: 4098 (db_sqlexception); SQLCode: -1009141, SQLState: ‘QCA51‘,
Severity: 14
[21061]: Table ‘ReCreator‘ cannot be found in the IQ store.
-- (db_iqutility.cxx 1929)
```

Differentiation of conditions can be a fine line. Consider these two very similar error messages for “table not found”:

```
Exception Thrown from db_iqutility.cxx:1929, Err# 105, tid 1194 origtid 1194
 O/S Err#: 0, ErrID: 4098 (db_sqlexception); SQLCode: -1009141, SQLState: ‘QCA51‘,
Severity: 14
 [21061]: Table ‘ReCreator‘ cannot be found in the IQ store.
-- (db_iqutility.cxx 1929)


Exception Thrown from db_iqutility.cxx:7212, Err# 7, tid 1188 origtid 1188
 O/S Err#: 0, ErrID: 1034 (db_dbccexception); SQLCode: -1000266, SQLState: ‘QDB09‘,
Severity: 23
 [21079]: Table ‘dbo.ix_staff_query_index‘ not found.
-- (db_iqutility.cxx 7212)
```

SQLCode -1009141 is explained in the documentation as:

“The stored procedure sp_iqtablesize cannot find the table. This stored procedure is for IQ tables only. It cannot be run against system tables or ASA tables.”

while SQLCode -1000266 is explained as:

“The specified table could not be opened. Either the table does not exists or the name format is incorrect The table name must follow the format [OWNER.]TABLE”

Note that the Severity code for the first one is 14 while for the second one it is 23.

Mapping alerts probably requires a combination of measures, including perhaps Severity and a list of specific error message code values known to represent issues of concern.

It is possible to encounter .iqmsg error messages with unspecified codes. Here is an example from IQ 12.7, showing the entire connection session:

```
Collation ISO_BINENG, Case Respect, Blank Padding On, Comparisons are Binary
 Txn 354435661 0 354435661
 Connect:  SA connHandle: 14856  SA connID: 31  IQ connID: 0000015137  User: ctxna_write
 Insert Started:
 CtxnOrder
 Exception Thrown from db_sqlins.cxx:15769, Err# 31, tid 1058 origtid 1058
    O/S Err#: 0, ErrID: 4098 (db_sqlexception); SQLCode: 0, SQLState: ‘?????‘, Severity: 0
 [-157]:
 Exception Thrown from db_sqlins.cxx:15769, Err# 31, tid 1058 origtid 1058
    O/S Err#: 0, ErrID: 4098 (db_sqlexception); SQLCode: 0, SQLState: ‘?????‘, Severity: 0
 [-157]:
 Rbck
 PostRbck
 Disconnect:  SA connHandle: 14856  SA connID: 31  IQ connID: 0000015137  User: ctxna_
write
```

This appears to be the result of a coding error, later fixed in IQ 15. Such messages should be reported to SAP Sybase IQ Technical Support for further analysis and correction.

Other types of messages in the .iqmsg log

In addition to standardized error messages classified by SQLCode, other types of messages can appear in the .iqmsg log:

- Abort (crash)
- Non-fatal error with stack trace
- Read (bufman) error
- Out of space error (IQ 12.7)
- Interrupts
- SySAM licensing

Abort

Here is an example of Abort messages appearing in the .iqmsg log:

```
 **************************************************
 *** Sybase IQ Abort:
 *** From: s_buf.cxx:968
 *** PID: 28656
 *** Exception Location: 0x0x6b6e01b0
 *** Error ID: 533, Exception Type: 3770376787
 *** Thread: 1802377536 (TID: 642, Original TID: 642)
 *** Error #: 2, O/S Error #: 0
 **************************************************

 ** Error from IQ connection: SA connHandle: 2619441 SA connID: 29 IQ connID: 0002535935
User: stdrptfil
 ** Time of error: 2011-07-25 18:30:49
 ** IQ Version: Sybase IQ/12.7.0/080707/P/ESD 5
 ** OS info: IQ built on: Enterprise Linux64 - amd64,Opteron64,X86_64/2.4.21-27.0.1.ELsmp,
Executed on: Linux/lx-chcssybp16/2.6.18-164.2.1.el5/#1 SMP Mon Sep 21 04:37:42 EDT 2009/
x86_64
 ** Command status when error occured: CURSOR ACTIVE
 ** Command text:
IQDEV8.GSRpt.spGSRpt_Standard_Order_2_LIST
...
 Dump all thread stacks at hos_throw.cxx:311 for PID: 28656

***************** This is the STACKTRACE ***************
…
```

Abort messages typically mean an IQ server has crashed. The system has to be restarted and the incident reported to SAP Sybase Technical Support. The key word to cue a DBA alert is “Abort”; beyond that any codes or text in the messages will be meaningful only for Technical Support analysis. The content of the stack trace is usually the most important part for identifying the nature of the crash.

On rare occasions an Abort can be interrupted by a second Abort resulting in the IQ server continuing to execute, though in an uncertain state. This is another reason it is important for monitoring to detect the Abort. If the IQ server has halted it needs to be restarted, but if an Abort has occurred and it is still running, SAP Sybase Technical Support should be contacted to discuss whether the server should be stopped and restarted.

Non-fatal error with a stack trace

Certain errors trigger additional diagnostics beyond a SQLCode type of message. These are logged in the .iqmsg log with a stack trace, and labeled as “Non-fatal” to distinguish them from Aborts. Here is an example:

```
Large Delete Failed for Index mincost.Transactions_2011Q4.ASIQ_IDX_T1984_I14_HG: 1381
5403 Rows Deleted. Expected 13815419.

 ********************************************************
 **
 ** Non-Fatal IQ Internal Error Detected
 ** at s_ohcIter.cxx:809 on thread 2028837184 (TID 1479)
 **
 ** Please report this to Sybase IQ support
 ** with the following diagnostic information,
 ** and, if possible, with the user‘s command.
 **
 ** Error from IQ connection: SA connHandle: 207245 SA connID: 55 IQ connID: 0000205176
User: MCD_DBA
 ** Time of error: 2011-11-15 15:47:53
 ** IQ Version: Sybase IQ/12.7.0/090824/P/ESD 7
 ** OS info: IQ built on: Enterprise Linux64 - amd64,Opteron64,X86_64/2.4.21-27.0.1.ELsmp,
Executed on: Linux/lx-chsybp208/2.6.18-238.19.1.el5/#1 SMP Sun Jul 10 08:43:41 EDT 2011/
x86_64
 ** Command status when error occured: CURSOR ACTIVE
 ** Command text:
delete mincost.Transactions_2011Q4 where HoldDate=‘20111108‘



***************** This is the STACKTRACE ***************

...

 ******************* End of STACKTRACE ******************
 **
 ** End of stack trace from Non-Fatal IQ Internal Error s_ohcIter.cxx:809
 **
 ********************************************************

 Exception Thrown from s_ohcIter.cxx:809, Err# 15, tid 1479 origtid 1479
 O/S Err#: 0, ErrID: 2068 (hs_idxexception); SQLCode: -1006003, SQLState: ‘QBA03‘,
Severity: 23
 [20241]: IQ Internal error. Please report this to Sybase IQ support. mincost.
Transactions_2011Q4.ASIQ_IDX_T1984_I14_HG 9 ???
-- (s_ohcIter.cxx 809)
...
 Exception Thrown from s_ohcIter.cxx:809, Err# 15, tid 1479 origtid 1479
 O/S Err#: 0, ErrID: 2068 (hs_idxexception); SQLCode: -1006003, SQLState: ‘QBA03‘,
Severity: 23
 [20241]: IQ Internal error. Please report this to Sybase IQ support. mincost.
Transactions_2011Q4.ASIQ_IDX_T1984_I14_HG 9 ???
-- (s_ohcIter.cxx 809
```

An alert for this type of error could trigger on the text “Non-Fatal IQ Internal Error Detected”, or maybe easier, on Severity code value 23 and SQLCode value -1006003 in the trailing messages.

Read error (bufman)

With this error, IQ’s buffer manager had a problem reading a page into memory. The contents of the page are dumped into the log. It is lengthy and that part has been omitted here.

```
 ********************************************************
 **
 ** Non-Fatal IQ Internal Error Detected
 ** at s_buf.cxx:1121 on thread 2130811200 (TID 1413)
 **
 ** Please report this to Sybase IQ support
 ** with the following diagnostic information,
 ** and, if possible, with the user‘s command.
 **
 ** Error from IQ connection:
 ** Time of error: 2011-06-20 03:52:09
 ** IQ Version: Sybase IQ/12.7.0/080707/P/ESD 5
 ** OS info: IQ built on: Enterprise Linux64 - amd64,Opteron64,X86_64/2.4.21-27.0.1.ELsmp,
Executed on: Linux/lx-chsybp161/2.6.18-164.2.1.el5/#1 SMP Mon Sep 21 04:37:42 EDT 2009/
x86_64
 ** Command status when error occured: NO COMMAND OR CURSOR ACTIVE

***************** This is the STACKTRACE ***************

...

===== Thread Number 2130811200 =====

 pc: 0x0 ()

 pc: 0x2aaadb6ecf1b pcstkwalk(stk_trace*, int, db_log*, hos_fd*)+0x10b()

 pc: 0x2aaadb6eba96 ucstkgentrace(int, int)+0xb6()

 pc: 0x2aaadb6ed36d StackTraceForThisThread(char const*, int, char const*, char const*,
char const*)+0x22d()

 pc: 0x2aaadb81e8be s_bufman_exception::s_bufman_exception[in-charge](char const*, int,
int, s_bufman const*, s_buf const*, short, unsigned long long, unsigned, s_diskblockheader
const*, int)+0x5ce()

 pc: 0x2aaadb812767 s_buf::ReadBufAndVerify(s_diskblockheader*, unsigned long long,
unsigned, unsigned, int, short, unsigned long long, unsigned, int)+0x213()

 pc: 0x2aaadb812918 s_buf::Read(unsigned long long, unsigned, int, short, unsigned long
long, unsigned)+0x140()

...

 ******************* End of STACKTRACE ******************
 **
 ** End of stack trace from Non-Fatal IQ Internal Error s_buf.cxx:1121
 **
 ********************************************************

 Exception Thrown from s_buf.cxx:1121, Err# 12, tid 1413 origtid 1413
 O/S Err#: 0, ErrID: 2055 (s_bufman_exception); SQLCode: -1009039, SQLState: ‘QSA39‘,
Severity: 14
```

```
 [20060]: temp Bufman: Incorrect page header read; buffer={btype=10,nlb=16,blk=1578,pbn=33
478464,npb=5,uid=1156293683,txn=0,sp=0,flgs=0xc10} page={btype=10,nlb=16,npb=4,ctr=136,cta
=134,pbn=33478464,txn=0,sp=0,ver=1,pb=97,pus=48654}.
-- (s_buf.cxx 1121)
 Page dump for sbuf @ 0x2ab2ad3fe3a0
 _dskBlk @ 0x2ab4e9382000
 Physical block #: 33478464 (0x1fed740)
 Blocks read: 5
 Expected Pad1: 51 (0x33), Pad2: 1578 (0x 062a)
 File #9 (catid 16419): /sybiqdb/IQPBO1R/./Temp/TEMP9R
 Starting Block: 33470464 (0x1feb800)
 File Physical Block Number: 8001 (0x1f41)
 File size: 2097148 blocks, 34359672832 bytes.

...
 &s_dskbkhdr| Block Type | Transaction ID | SavePoint | Ver | Physical Block | Data
|NumPBlk|NumLBlk| Req/Actual Comp Types
 -----------+---------------------+---------------------+------------+-----+-----------
----------+------------+-------+-------+---------------------
 0x2ab4e9382000 | sort | 0 | 0 | 1 | 33478464 | 0x2ab4e9382020 | 16 | 16 | NONE/NONE
...
 End of page dump.
 There was an s_buf::Read() error during s_bufman::FindForPrefetch().
 Exception Thrown from s_buf.cxx:1121, Err# 12, tid 1413 origtid 1413
 O/S Err#: 0, ErrID: 2055 (s_bufman_exception); SQLCode: -1009039, SQLState: ‘QSA39‘,
Severity: 14
 [20060]: temp Bufman: Incorrect page header read; buffer={btype=10,nlb=16,blk=1578,pbn=33
478464,npb=5,uid=1156293683,txn=0,sp=0,flgs=0xc10} page={btype=10,nlb=16,npb=4,ctr=136,cta
=134,pbn=33478464,txn=0,sp=0,ver=1,pb=97,pus=48654}.
-- (s_buf.cxx 1121)
```

The portion of the stack trace shown is typical. An alert could trigger on “Bufman: Incorrect page header read” (could be “temp” as here or “main”) or the SQLCode value. Notice that the Severity code is only 14. There are other SQLCode values for similar errors — for example SQL Code -1009046 — so given those two circumstances, it might be useful to specify a range of SQLCode values to trigger an alert.

Out of space error

In IQ 12.7 and prior, running out of dbspace caused the IQ server to wait for dbspace to be added, and the message indicating this was not in the exception format so it was difficult to detect. Here is an example of that message:

```
2003-07-22 09:08:09 0000000000 [20152]: You have run out of IQ STORE dbspace in
Database /sybiqfs/ASIQ-12_5/statsdb/statsdb.db. In another session, please issue a CREATE
DBSPACE ... IQ STORE command and add a dbspace of at least 8 MB.
```

IQ 15 no longer handles it this way. As documented: “If there is not enough temporary or main dbspace available for a buffer or dbspace allocation request, then the statement making the request rolls back.” Messages in the .iqmsg log look like this for IQ 15:

```
Exception Thrown from s_blockmap.cxx:3699, Err# 0, tid 318 origtid 318
O/S Err#: 0, ErrID: 2096 (s_nodbspaceexception); SQLCode: -1009170, SQLState: ‘QSB66‘,
Severity: 14
[20223]: You have run out of space in IQ_SYSTEM_MAIN DBSpace.
-- (s_blockmap.cxx 3699)
Rbck
PostRbck
```

An alert monitoring for this can look for the SQLCode value or the text “You have run out of space”.

Interrupt messages

In the .iqmsg log, interrupt messages can look like this:

```
Exception Thrown from s_run.cxx:752, Err# 0, tid 1859 origtid 1859
 O/S Err#: 0, ErrID: 519 (hos_attnexception); SQLCode: -299, SQLState: ‘57014‘, Severity: 16
 [2188]: Statement interrupted by user
```

or this variation (different ErrID):

```
Exception Thrown from st_command.cxx:618, Err# 0, tid 388 origtid 388
 O/S Err#: 0, ErrID: 5125 (st_commandException); SQLCode: -299, SQLState: ‘57014‘, Severity: 16
 [2188]: Statement interrupted by user
```

This related message may or may not be nearby in the .iqmsg log:

```
Cancellation request received: SA connHandle: 563454 SA connID: 24 IQ connID: 0000860607
User: tu_rdr
```

Interrupt messages occur when a user at a keyboard enters Control-C or a program issues a cancel on a connection. The “Cancellation request received” message appears in the .iqmsg log as a result. If a code module further on in the sequence of execution has interrupt handling installed the interrupt will be caught and an exception message logged as shown above.

Usually alerts monitoring will not want to pick up interrupt messages, but this shows what they look like in the .iqmsg log.

SySAM licensing

SySAM messages may appear in the .iqmsg log when the database is closed. The database is closed and reopened at boot time so one set of SySAM messages should appear near the start of the .iqmsg log.

Messages indicating a problem can look like the following two examples. Notice that the second one has a SQLCode assigned.

```
Failed to obtain license(s) for IQ_CORE feature from license file(s) or server(s).
Insufficient IQ_CORE CPU licenses. IQ requires 4 IQ_CORE (”DT“) CPU licenses for use on
this machine but only 2 could be obtained. Check whether additional licenses are available
at the Sybase Product Download Center.
License feature name: IQ_CORE

Failed to obtain license(s) for IQ_CORE feature from license file(s) or server(s).
 Licenses exist for IQ, but a license containing the configured (PE=EE;LT=EV;AP=IQ)
attributes could not be obtained. Verify that IQ is configured to use the correct type
of license, and either reconfigure; or generate and deploy the desired license from the
Sybase Product Download Center.
 License feature name: IQ_CORE
 License search path: C:\Sybase\SYSAM-2_0\licenses\IQDEV10_J9841-2-1.lic;D:\IQDEV\Scripts\
IQDEV10_J9841-2-1.lic;
 FLEXnet Licensing error:-73,125
 For further information, refer to the Sybase Software Asset Management website at http://
www.sybase.com/sysam
 License Checkout Failed for IQ_CORE
 Exception Thrown from hos_sysam.cxx:881, Err# 0, tid 29 origtid 29
 O/S Err#: 0, ErrID: 549 (hos_sysamException); SQLCode: -1006269, SQLState: ‘QBC69‘,
Severity: 14
 [21161]: No SYSAM License Available
```

.srvlog messages

Startup messages tracking boot of the IQ server process are written to the .srvlog. After that, this log contains mainly informational messages about inserts and deletes and for connections terminating abnormally, such as when a user program terminates a connection without proper exit steps.

Here is an example of an error occurring at startup:

```
I. 01/11 08:14:50. Finished checkpoint of ”pkns“ (pkns.db) at Fri Jan 11 2013 08:14
E. 01/11 08:14:52. Error: No SYSAM License Available
E. 01/11 08:14:52. No SYSAM License Available
I. 01/11 08:14:52. Database server shutdown due to startup error
```

Here are examples of messages for two serious conditions that can be shown in this log for a running server.

Fatal error: memory exhausted

Certain queries can consume SQL Anywhere "catalog cache" memory, and it can increase if the IQ configuration allows it and there is available RAM on the server host machine. Messages like this will be logged in the .srvlog file as SA catalog cache memory is increased to meet demands:

```
I. 02/18 23:57:46. Cache size adjusted to 261536K
I. 02/18 23:57:46. Cache size adjusted to 261796K
```

If these continue to the point where all available RAM is consumed, the database server terminates with the following error in IQ 15:

```
I. 02/18 23:57:46. Cache size adjusted to 262060K
E. 02/18 23:57:46. Fatal error: no free pages available in cache
```

(In IQ 12.7 this is displayed as "Fatal error: memory exhausted".)

There is no error number for this condition. Any message text in the .srvlog including "Fatal error" should be picked up for an alert.

Assertion failure

On rare occasions a non-viable code path can be followed and an "assertion error" logged in the .srvlog. Here are two examples:

```
E. 02/18 08:45:01. *** ERROR *** Assertion failed: 102801 (12.7.0.2074)
E. 02/18 08:45:01. Unable to find table definition when creating/altering a column

E. 07/06 12:26:41. *** ERROR *** Assertion failed: 102300 (12.7.0.2056)
E. 07/06 12:26:41. File associated with given page id is invalid or not open
```

This usually terminates the server; often when this occurs there are no other error messages in any other log. The SQL Anywhere engine suddenly shuts down, halting IQ.

.stderr messages

The .stderr log shows useful information from boot time about resources configured for the server. Anything from the server process directed to stdout can also appear here. In the event of an IQ Abort it will show a minimal set of messages indicating that an Abort occurred, part of the complete set of Abort diagnostic messages in the .iqmsg log.

Detailed SySAM messages will be written to this log file as well as the .iqmsg log. Here is an example showing a problem:

```
Using licenses from: /opt/sybase/SYSAM/licenses/license.dat:/opt/sybase/SYSAM-2_0/
licenses/SYBASE.lic:/opt/sybase/SYSAM-
2_0/licenses/license.lic:/opt/sybase/SYSAM-2_0/licenses/examplea.com.lic:/opt/sybase/
SYSAM-2_0/licenses/examplea.com.lic:/opt/catalog/EX_PD_HFBF1/*.lic
Failed to obtain license(s) for IQ_CORE feature from license file(s) or server(s).
No such feature exists. Verify that a license containing the feature name IQ_CORE is
available or generate and deploy the license from the Sybase Product Download Center.
License feature name: IQ_CORE
License search path: /opt/sybase/SYSAM/licenses/license.dat:/opt/sybase/SYSAM-2_0/
licenses/SYBASE.lic:/opt/sybase/SYSAM-
2_0/licenses/license.lic:/opt/sybase/SYSAM-2_0/licenses/examplea.com.lic:/opt/sybase/
SYSAM-2_0/licenses/examplea.com.lic:/opt/catalog/EX_PD_HFBF1/*.lic:
FLEXnet Licensing error:-5,414
For further information, refer to the Sybase Software Asset Management website at http://
www.sybase.com/sysam
DBSPAWN ERROR: -82
Unable to start specified database: autostarting database failed
```

Being the destination for stdout may result in uncommon messages in the .stderr log occasionally. For example, IQ 15.4 produced these messages:

```
MIPC Failed to update connectivity status for server id : 2
MIPC Failed to update connectivity status for server id : 2
MIPC Failed to update connectivity status for server id : 2
```

They indicate a problem with network communications between nodes in an IQ Multiplex installation. The issue may be due to network problems.

Request Log messages

User-level error messages are logged in the Request log, along with all command activity. Common errors are logged as follows:

```
code: -1000266 “Table ‘dbo.ReCreator‘ not found.  -- (db_iqutility.cxx 7212)“

code: -131 “Syntax error near ‘(end of line)‘ on line 1“
```

The code values are SQLCodes. Besides the SAP Sybase IQ Error Message guide, another source of information about these message codes is the SQL Anywhere Error Messages guide. See the topic "SQL Anywhere error messages sorted by SQLCODE".

Less common errors will be reported similarly. Here is what the Request log shows for the Non-Fatal IQ Internal error used as an example above in the .iqmsg log section:

```
code: -1006003 ”IQ Internal error. Please report this to Sybase IQ support. mincost.
Transactions_2011Q4.ASIQ_IDX_T1984_I14_HG 9 ??? -- (s_ohcIter.cxx 809) ”
```

Note that an "out of connections" condition is logged only in the Request log, not the .iqmsg log or .srvlog. Connections are managed by SQL Anywhere and, like the .srvlog, the Request log belongs to the SA engine. Messages look like this:

```
** ERROR conn: 1078055 code: -102 ”Database server connection limit exceeded“
** DONE conn: 1078055 CONNECT
** ERROR conn: 1078056 code: -102 ”Database server connection limit exceeded“
** DONE conn: 1078056 CONNECT
```

The –gm startup configuration parameter sets this limit.

When there are no messages

Sometimes a problem with the IQ server can occur without leaving messages in the .iqmsg log. In such circumstances, inspect the other IQ logs. If the server has halted, look in the .srvlog and the .stderr log. At the end of the .srvlog, an "assertion failed" message may be found, as described earlier in the section regarding messages in the .srvlog. When IQ terminates with no messages, often the cause is in the SQL Anywhere engine. Check the Request log to see the last command executed. An enormous command, such as can be submitted by code generators, can exceed stack space and cause the SA engine to stop, bringing down IQ.

www.sap.com