

IQ 15 Best Practices Guide



TABLE OF CONTENTS

Executive Summary	11
Goals of this Document	11
Tips For Using This Document	11
Recommended Reading	11
Introducing IQ 15	11
What's New in IQ 15	12
Product Goals	12
Overview of Major New Features	13
Summary of New Features Introduced in Each Release of IQ 15	13
Engine Parallelism in IQ 15	17
Hardware Sizing and Server Configuration for IQ 15	18
Server-Level Changes for IQ 15	19
Catalog Changes in IQ 15	19
New Recompile Clause for Alter Procedure/Alter Function	19
Tip: Recompile Stored Procedures When a Schema Changes	19
Repeatable DDL	19
Custom Collations Have Been Depreciated in IQ 15	20
Changes To View Dependencies and Altering Schemas	20
Changes to the Request Log (zrlog File)	20
The Request Log Content Has Changed	20
The New Request Log Monitoring Procedures	21
Tip for Reading Long Statements	21
Storage Changes for IQ 15	22
Overview of DBspaces and DBstores	22
DBstore Changes	22
Stores Can Now be Comprised of Multiple DBspaces	22
Tip: Use the Read-Only/Read-Write Feature for DBspace Administration.	23
Table Partitioning in IQ 15 (Using The Very Large Database Option)	23
Tip: Optimize Database Object Placement	23
Understanding Query Optimizer Changes In IQ 15	24
Query Engine Improvements	24
Improved Tokenization	24
Multicolumn HG Indexes	24

Component Integration Services	24
Transaction Out-of-Space Behavior	24
Multiplex Changes for IQ 15	25
Multiple Reader and Writer Nodes	25
Tip: You Can Have Multiple Writer and Reader Nodes	25
The Coordinator Node	26
Correctly Sizing the Coordinator Node	26
Obsolete Multiplex Features	26
Simulating Local Store Functionality in IQ 15	27
Local Stores Have Been Depreciated in IQ 15	27
How To Simulate a Local Store in IQ 15	27
Inter-Node Communication (INC) and SQL Remote	27
Tip: Set The Max Number of User Connections (-gm option)	28
Multiplex Inter-Node Communication (MIPC)	28
Dynamic Collision	29
Tip: Minimize Query Disruptions	29
Multiplex Do's	29
Multiplex Don'ts	30
Distributed Query Processing (DQP) and Logical Servers	31
Distributed Query Processing (DQP)	31
Logical Servers	31
Sizing the New Shared Temporary DBspace	31
Full Text Search in IQ 15	32
Migration and Upgrade Tips for IQ 15	33
Recommended Reading	33
IQ Versions	33
MS Windows and the 32-Bit Version of IQ	33
Backup Before an Upgrade or Migration	33
Backup After an Upgrade or Migration	34
Establish a Performance Baseline for Your New install	34
Test Your New Install	34
Run SP_IQCHECKOPTIONS Before and After an Upgrade	35
Verify Database Consistency Before and After a Migration	35
Migrating from IQ 12.7 to IQ 15 (Unload and Reload the Catalog)	35

IQ Catalog Changes	35
Use the IQUNLOAD Utility to Migrate from IQ 12 to IQ 15	36
Tips for the IQUNLOAD Utility.....	36
Use the Schema Unload Mode Before Performing the Load	36
Use the “-IQRO” Switch When Performing an UNLOAD/RELOAD of the Catalog	36
Sample Task List for Performing a Multiplex Migration from IQ 12 to IQ 15	36
Known Issues For The IQUNLOAD Utility	38
Drop the Global Temporary Tables	38
Reset the DATE_FORMAT and DATE_ORDER	38
Drop Any JAVA Classes or JAVA Procedures	38
Failing to Start the IQ Server in READ-ONLY Mode May Result in Errors.....	38
Resolving the “Parse Stack Overflow” Issue	39
The Parse Stack Overflow Error	39
Fix the Stack Overflow Error by Using IQUNLOAD to Increase the Catalog Page Size	39
Other Solution Options	39
Upgrading from One Version of IQ 15 to Another (Alter Database Upgrade).....	40
Use the “-IQRO” Switch When Performing the Alter Database Upgrade Command	40
The “-IQRO” Switch	40
Example of Using the “-IQRO” Switch in a Simplex Upgrade.....	40
Example of Using the “-IQRO” Switch in a Multiplex Upgrade.....	41
The Max Connections Error	42
Resolving the Max Connections Error.....	42
Rebuilding Pre-IQ 15.4 FP Indexes for Enhanced Compression	43
Licensing Changes For IQ 15	44
Legacy Licensing	44
New Licensing Options for IQ 15.....	44
The Evaluation Version vs the Evaluation License	46
The Evaluation Version for IQ	46
The Evaluation License for IQ.....	47
SySAM2	48
SySAM is Now Required for All Licenses.....	48
New Features	48
Enhancements.....	48
License Files.....	48
Overview of SySAM Licensing Checks	49
Scan the Log for SySAM Errors (And Take the Appropriate Action)	49

Tip: Scan the Log for SySAM Errors	50
Troubleshooting “Product Edition” and “License Type” Errors	50
SySAM Debugging Options.....	51
Troubleshooting AIX Multi-Processor Issues	51
Operational Management.....	52
IQ Administration Tools	52
Sybase Administration GUI’s.....	52
Stored Procedures for IQ Management.....	52
OS-Level Tools for IQ Management	53
Sybase Central (SC).....	53
Sybase Control Center (SCC)	54
SCC Roadmap	54
Overview of SCC	54
History of SCC for IQ.....	54
SCC Version Restrictions for IQ	55
SCC Administration Features for IQ 15.3	55
SCC Administration Features for IQ 15.4	55
New SCC Features	55
Tip: Backup the SCC Repository	56
Using The Sybase Control Center.....	57
Summary of Procedures for Installing and Starting SCC.....	57
1. Installing the SCC Server.....	57
2. Starting the SCC Server	57
3. Installing the IQ Server’s SCC Agent.....	59
4. Starting the IQ Server’s Local SCC Agent.....	59
5. Connecting to the SCC Server	59
Summary of Procedures for Registering and Authenticating an IQ Server to SCC.....	60
1. Registering Your IQ Server to SCC.....	60
2. Authenticating Your IQ Server to SCC	60
3. Registering Your IQ Server’s SCC Agent	60
4. Authenticating Your IQ Server’s SCC Agent	60
DBISQL and Open Client Utilities	61
The Open Client SDK	61
Open Client Utilites	61
Deprecated Utilities	61
Reinstated Utilities.....	61

IQISQL Versus ISQL	61
Interactive SQL (DBISQL)	62
DBISQL	62
DBISQL-Java Versus DBISQL-C	62
DBISQL-C, DBISQL-Classic, and Interactive ISQL-Classic Have Been Deprecated	62
DBISQL-Java is the New Standard	62
DBISQL-Java Features.....	63
Interactive SQL (JAVA-Based).....	63
Command-Line Mode (DBISQL-JAVA)	63
Returning Multiple Result Sets in DBISQL.....	63
Windows GUI.....	63
UNIX Command Line	64
UNIX GUI	64
DBISQL-Java Changes	64
Tip: Do Not Mix SQL Dialects	64
Interactive SQL Restrictions	64
Network Connectivity	65
New Client Access API's	65
Connectivity Tips	65
ODBC Driver Manager on UNIX.....	66
Network Connectivity	66
NAS Support	66
Client Side Loading	67
The IQ_BCP Utility has been Replaced.....	67
New Features	67
Client Driver and Protocol Support.....	67
Security.....	68
Performance Advantages for Client Side Loading	68
Performance Enhancements for Client Side Loading	68
Performance Tips	69
Parallelism	69
New Database Options For Tuning Parallelism.....	69
Tip: Keep Your Old Query Run Times and Query Plans.....	69
New Optimizer Configuration Parameters for IQ 15	69

Additional Performance Recommendations.....	69
Data Modeling Recommendations	70
Proper Data-type sizing.....	70
IQ Unique and Minimize_Storage.....	70
Null Values.....	70
Unsigned Data types	70
LONG VARCHAR and LONG BINARY	70
Large Object Storage.....	71
VARCHAR vs. CHAR.....	71
When to Use Indexes	71
Temporary Tables.....	72
Cursors	72
Security Recommendations	73
Authorities.....	73
User Login Policies.....	74
IPV6	74
Sybase Central Authentication.....	74
Advanced Security Option.....	75
DataBase Maintenance Recommendations.....	76
Backup and Recovery	76
Monitoring the Main DBspace and Temp DBspace Usage	77
Stored Procedures for Monitoring DBspace	77
How To Use the “dropconn” Stored Procedures for Monitoring DBspace	77
Message Log Management Using -iqmsgsz and -iqmsgnum	78
Monitoring the Catalog’s Disk Space	78
Monitoring Free Space Using the New “sa_disk_free_space” Stored Procedure	78
Tip: Monitor Your Free Space	79
Auditing.....	79
Tip for Enabeling Auditing.....	79
Temporary Files.....	79
Tip: Manage Your Temp Files.....	79
Database Connection Enhancements	80
Scan the Log for SySAM Messages.....	80
IQ Database Options That Help Optimize Performance	80
The “Query Plan” Option	80

The “Force_No_Scroll_Cursors” Option	80
The “Append_Load” Option.....	80
The “Default Disk_S Striping” Option	81
The “xp_cmdshell” Option	81
The “MINIMIZE_STORAGE” Option.....	81
The “Packetsize” Option And The “INSERT...LOCATION” Operation	81
Backup and Restore Samples and Scenarios.....	82
Backup Methods	82
IQ Server-Level Backups.....	82
Virtual Backup Advantages	82
Backup Tips	83
Backup Restrictions	83
IQ Restore.....	83
Backup Verification and Diagnostics	84
Diagnostics Checklist for Troubleshooting Problems in IQ.....	85
Diagnostics for All Problems	85
Diagnostics for Slow Server Performance.....	86
1. IQ Server-Level Diagnostics for Slow Server Performance	86
2. Operating System Diagnostics for Slow Server Performance	87
Diagnostics for Slow Query Performance	87
Diagnostics for Non-Fatal Stacktraces	88
Diagnostics for a Crash or Fatal Stacktrace.....	88
Diagnostics for a Hung Server	89
Diagnostics for a Connection Issue	89
Diagnostics for a Backup and Restore Issue	89
Diagnostics for SySAM Issues.....	90
User Defined Functions.....	91
High-Performance, External C/C++ UDFs (Internally Executed UDFs).....	91
Partner UDFs (The IQ_UDF Option)	91
In-Database Analytics (The IQ_IDA Option).....	91
Interactive SQL UDFs (Internally Executed)	92
Externally Executed UDFs (Java, Perl, Etc)	92
Scalar and Aggregate Features for UDF V3 vs V4	93
API Version 3 (V3).....	93
API Version 4 (V4).....	93

Failover Considerations for the IQ_IDA Option	93
Tip: Consider Your Failover Logic	93
UDF Programming Practices	94
UDF Do's for External C/C++ UDFS.....	94
UDF Don'ts for External C/C++ UDFS	94
UDF Performance and Tuning tips	95
UDF Debugging.....	95
UDF Internationalization	96
UDF Time Series	96
Database Options	97
Purpose of Database Options	97
Where to Find Documentation of Database Options	97
Range of Behavior	97
Compatibility	97
Error Handling	97
Concurrency and Transactions.....	97
Performance and Optimizer Behavior	97
Query Behavior	98
Diagnostics	98
How to Set a Database Option	98
Syntax	98
Examples	98
How to Reset or Delete a Database Option	98
Examples	98
Rules for Resetting Database Option Values	98
How to Control Database Options	99
Option Scope.....	99
Option Duration	99
Precedence.....	99
Dynamic Versus Static Options	99
Tip: Do Not Change Options When a Cursor is Open	100
Tips on Setting DB Options.....	100
Set User	100
Set Public.....	100
Set Temporary (without the PUBLIC option)	100
Set Public Temporary	100

Permanent (An Implied Setting)	100
Using SP_IQCHECKOPTIONS to Display Your Database Options	101
Tip: Run sp_iqcheckoptions BEFORE and AFTER an upgrade to verify options settings	101
Database Options that Affect Performance.....	101
Scrollable Cursor.....	101
Query Temp Space	101
Query Plan Setting.....	102
Minimize Storage Settings	102
Compatibility Settings	102
Table Load and Unload Settings.....	102
Load_Memory_Mb	102
Row_Count =	102
Database Options for Collecting Query Performance Data	103
Tip: Set the Following Database Options to Collect Query Plan Information:	103
New and Changed Database Options for IQ 15	104
Listing of Deprecated Options for IQ 15	104
Listing of New Options for IQ 15.3 and 15.4	104
Listing of Enhanced Options for IQ 15.4	106
Listing of Options with New Defaults for IQ 15.4	107
Listing of Options with New Defaults for IQ 15.1, 15.2 and 15.3	107
Listing of New Options for IQ 15.0, 15.1, and 15.2	109
Appendices	117
APPENDIX 1: Listing of Deprecated Options for IQ 15.0, 15.1, 15.2, 15.3 and 15.4	117
APPENDIX 2: Sp_dropConnOnMainUsed Procedure	119
APPENDIX 3: Sp_dropConnOnTempUsed Procedure	121

EXECUTIVE SUMMARY

GOALS OF THIS DOCUMENT

This document presents the best practices for IQ 15. 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 15'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 15. 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 15 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) for each version of Sybase IQ you plan to install. Each of these documents contains information that is critical to understanding IQ's latest features and to achieving a successful installation of Sybase IQ in your environment.

INTRODUCING IQ 15

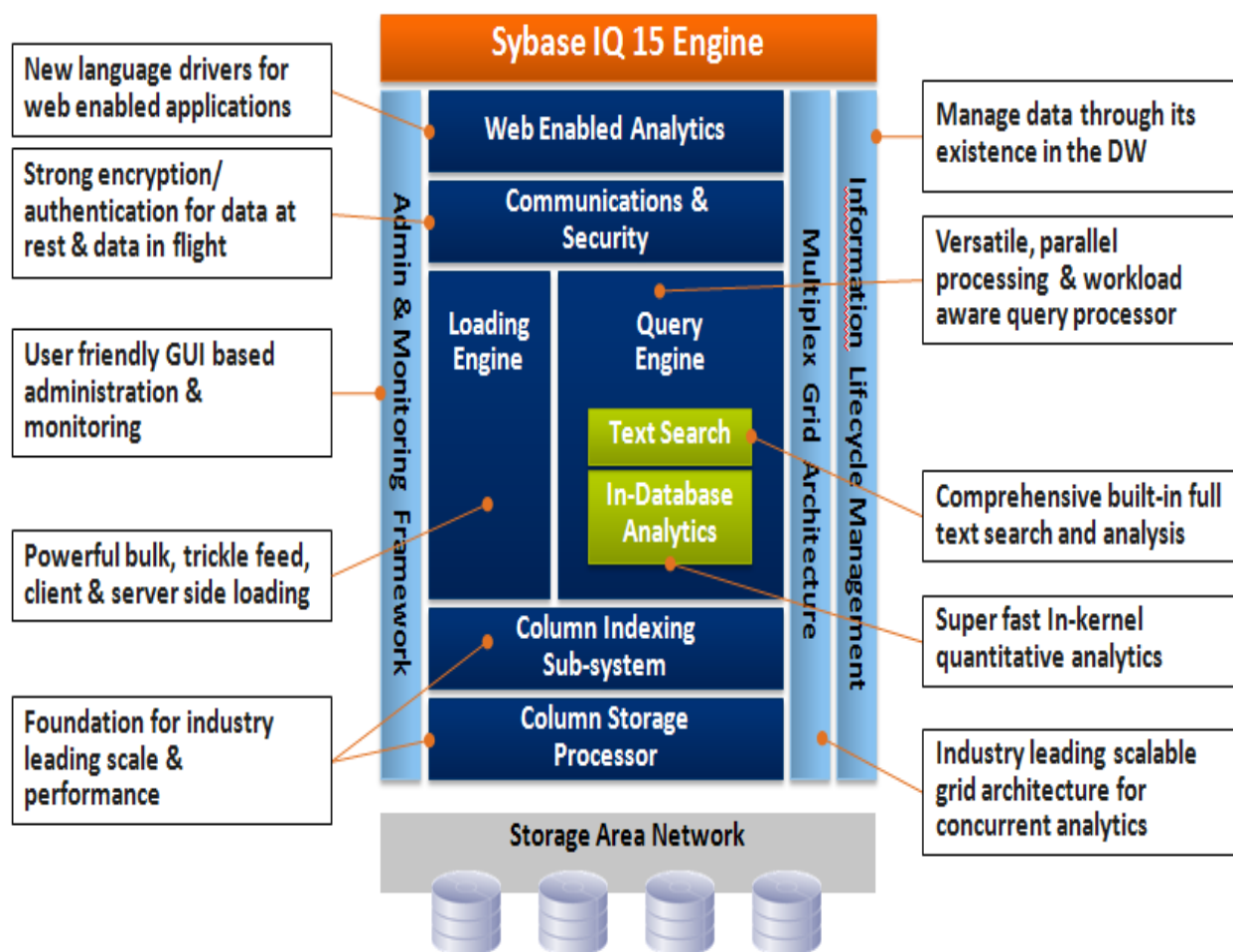
- Sybase 15 is the culmination of a series of releases designed to offer an analytics platform that provides both sophisticated analytics, and performance scalability in multiple dimensions.
- IQ 15 was first released in March of 2009 and it was the first step towards improving the parallelism of IQ running on SMP architectures.
- IQ 15.1 was released several months later and introduced high-performance analytics through user defined functions (UDFs) deployed by Sybase Certified Partner libraries.
- IQ 15.2 was released in June 2010 and provided an infrastructure for supporting text analytics and SQL operators to construct search expressions.
- IQ 15.3 was released in June 2011 and introduced a new "shared everything" MPP (massively parallel processing) architecture to distribute queries across multiple host servers.
- IQ 15.4 was released in November of 2011, and is the latest release of IQ. This version improves UDF performance and enables any customer to create and use high-performance analytics.
- For more details on the features introduced in each version of IQ 15, please read the "New Features Summary" for each of these versions of IQ.

WHAT'S NEW IN IQ 15

PRODUCT GOALS

IQ 15 has been one of the biggest releases in Sybase IQ's history. Prior to developing this release, Sybase 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, the latest version of IQ 15 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 15 has to offer:



OVERVIEW OF MAJOR NEW FEATURES

- IQ 15 has introduced a number of major features designed to improve performance, scalability, maintenance, security and analytics. These included:
 - Increased CPU parallelism.
 - Increased server parallelism using logical servers and new multiplex architecture:
 - Logical Servers: In IQ 15.3, Logical Servers enable the grouping of multiple computer resources into one logical entity. A logical server may consist of one or more servers in an IQ multiplex configuration. Multiple logical servers can be created to serve different groups of applications or users. If distributed query processing is enabled, Sybase IQ distributes the query execution to those nodes that are members of the logical server.
 - Multiple Writers and Readers: IQ 15's new multiplex architecture allows multiple writer and reader configurations. It allows read-write transactions from multiple servers in the multiplex. The primary server, or coordinator, manages all global read-write transactions and maintains the global catalog and metadata. The table version log (TLV log) is also maintained by the coordinator and stores information about DDL operations and communicates information about new table versions to the secondary servers (readers or writers).
- Distributed Query Processing (DQP): With the introduction of distributed query processing in IQ 15.3, query processing can now be distributed in parallel across multiple groups of computer resources using IQ's logical server functionality.
- Multiple DBspaces: In order to better handle tiered storage, partitioning, and object placement on disk, Sybase IQ 15 introduced multiple DBspaces. You can now create any number of DBspaces to hold data objects. These objects include tables, columns, indexes, and partitions.
- Enhanced Security: Security enhancements include changes to network encryption support and user login management. Support of FIPS encryption, Kerberos authentication, and column encryption is included in the separately licensed Sybase IQ Advanced Security Option. Support of IPv6 is included without a separate license.

SUMMARY OF NEW FEATURES INTRODUCED IN EACH RELEASE OF IQ 15

Below, is a brief summary of the features introduced in each release of IQ 15. For more details, please read the "New Features Summary" for each release of IQ 15.

- **IQ 15.0 GA** was first released in March, 2009. It provided the first steps towards enhanced parallelism of IQ running on SMP architectures. Its features included:
 - Improved query performance using enhanced parallelism, improved tokenization, better subquery performance and reduction in cache and temp space
 - Direct loading of data from clients.
 - Table loads for large single (fact) tables using parallel loads of HG and WD indexes.
 - Support for loading partitioned tables.
 - Enhanced range partitioning.
 - Enhanced security using FIPS, Kerberos and IPv6 support.
 - Enhanced space management for the IQ main store and IQ temporary store.

- **IQ 15.1** was released in July of 2009 and introduced high-performance, in-database analytics for Sybase Certified Partners. The highlights of this release include:
 - High-Performance external V3 C/C++ user-defined functions for deployment by Sybase Certified Partners.
 - Time series and forecasting functions.
 - Enhanced scalar date and time functions.
 - New aggregate functions.
 - FIPS supported on Windows x64.
- **IQ 15.2** was released in June of 2010 and provided an infrastructure for supporting text analytics. This included large object storage and retrieval capabilities, indexes designed to quickly locate and score terms, and SQL operators to construct search expressions. Highlights of this release included:
 - Enhanced performance and scalability for “full text” searching and Large Object management.
 - Improved CIS performance.
 - Improved ODBC driver manager.
 - Improved Security.
 - SQL function support for:
 - Microsecond date and time functions.
 - SQL:2008 OLAP functions.
 - New time series and forecasting functions.
 - Statements and Options enhancements for DIVIDE_BY_ZERO_ERROR option, ENABLE_LOB_VARIABLES option, MAX_PREFIX_PER_CONTAINS_PHRASE option, TEXT_DELETE_METHOD option, new and changed SQL statement syntax.
 - Sybase IQ data access APIs.
 - Utility enhancements for the command line initialization utility (iqinit), Server startup (the -xd switch), and thread infrastructure improvements.
 - Sybase Control Center is a new Web-based GUI tool for managing and monitoring IQ. For more details, see [Sybase Control Center 3.2](#) and [Sybase Control Center 3.2.3-Sybase Control Center For Sybase IQ](#).
- **IQ 15.3** was released in June of 2011 and introduced a new MPP (massively parallel processing) architecture to distribute queries across multiple servers. The main features of this release included:
 - Distributed Query Processing (DQP): This feature improved performance in IQ multiplex configurations by distributing query processing across multiple computer resources in one or more logical servers (such as nodes in a cluster or one or more separate servers). DQP occurs automatically for most queries that qualify for parallelization.
 - Logical Servers: Enables the grouping and presentation of a subset of multiplex computer resources as one logical entity. Multiple logical servers can be created to serve different groups of applications or users. If distributed query processing is enabled, Sybase IQ distributes the query execution to those nodes that are members of the logical server. You can also dynamically add and remove nodes from a logical server to accommodate the individual application resource needs.

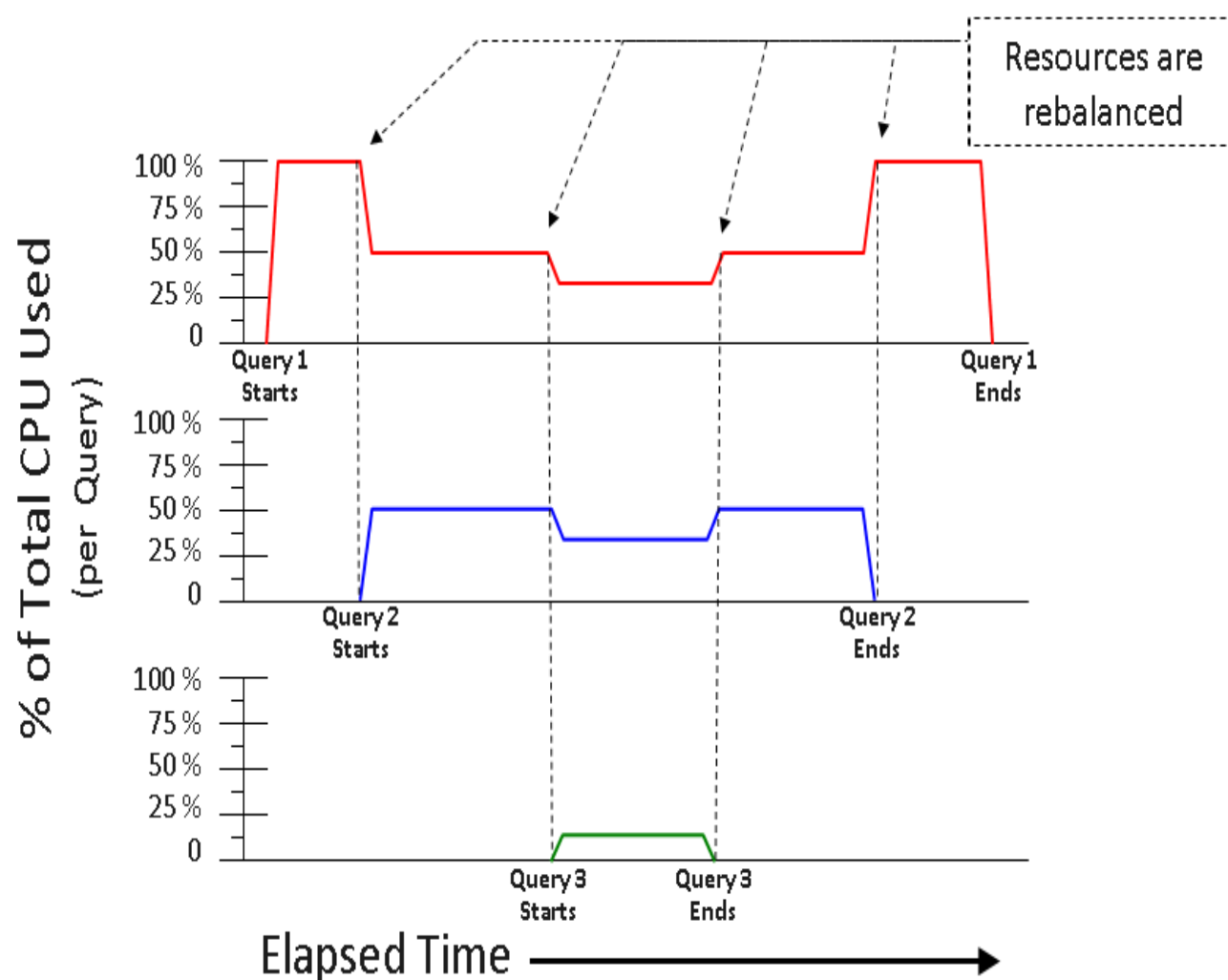
- Multiplex Interprocess Communication (MIPC): MIPC is a fully meshed communication framework that supports distributed query processing and high availability for IQ multiplex servers.
- Support for the BIGTIME and BIGDATETIME datatypes when using ASE's Component Integration Services (CIS) and when using the INSERT...LOCATION command.
- Built-in support for Web services, which allows handling of standard SOAP and HTTP requests.
- Enhanced Parallelism within Queries.
- Enhanced Predicate Evaluation.
- Ruby Driver Support.
- Sybase IQ InfoPrimer: Previously known as Sybase ETL (Extract, Transfer and Load). This product is now packaged and delivered as a separately licensed option to Sybase IQ 15.3 and must be installed separately using its own installer.
- **IQ 15.4** was released in November of 2011 and is the latest version of IQ. Its primary goal was to strengthen IQ's advanced analytics functionality. This included extending the performance and usability of in-database analytics by introducing the InDatabaseAnalytics option to all IQ customers. These new features include the following:
 - Enhanced Analytics:
 - The InDatabase Analytics Option: This is a new license type that enables customers to build, deploy and run their own external V4 C/C++ user defined functions (UDFs) within IQ.
 - Table parameterized functions (TPFs): These are a subset of table UDFs that can accept either table or scalar input parameters (unlike table UDFs, which accept only scalar).
 - For more details, please see the [Sybase IQ 15.4 User-Defined Functions](#) and the *UDF section* later in this document.
 - Enhanced Compression of fixed length data types (CHAR and BINARY), and variable-length data types (VARCHAR, VARBINARY, LONG VARCHAR, and LONG VARBINARY). This compression applies to both encrypted and non-encrypted columns.
 - Increased performance of Java External Environment UDFs.
 - Spatial Data Support:
 - SQL Anywhere is the row store component of Sybase IQ that manages the catalog, handles database connections, security, and SQL parsing. SQL Anywhere 12 now incorporates geospatial data types. Therefore, in IQ 15.4 you can now join spatial data stored in SQL Anywhere with other data in IQ. Currently, scalability is limited. In the future, we plan to incorporate geospatial data types natively within IQ. However, the current implementation performs well in our tests: The joining of 100 million SQL Anywhere geospatial data points with IQ data took only a few seconds.

- Spatial Data improvements for IQ 15.4 include the following:
 - Load times for shapefiles, polygons and multipolygons, and collections containing polygons belonging to round-Earth spatial reference systems has been improved.
 - Introduction of support for the following spatial set operations:
 - ST_Union and ST_Intersection applied to complex geometries.
 - Spatial predicates such as ST_Contains and ST_Intersects.
 - Spatial predicates where one of the geometries is a point.
 - Spatial predicates applying ST_WithinDistance or ST_Distance to indexed round Earth geometry columns.
 - Interactive SQL support of SHAPEFILE and SRID clauses for the INPUT statement.
 - A new stored procedure, “st_geometry_load_shapefile” has been added to allow loading of an ESRI shapefile. Note: You must upgrade your database to access this new stored procedure.
 - A new database option, “st_geometry_interpolation” controls the interpolation tolerance for the St_CircularString constructors.
 - Enhancements to ST_WithinDistanceFilter: The spatial predicate "ST_WithinDistanceFilter" is now supported for geometries in round-Earth spatial reference systems.

ENGINE PARALLELISM IN IQ 15

Version 15 of Sybase IQ brought about a fundamental change in both load and query operations. The engine has evolved to perform significantly more operations in parallel than before. The parallelism in IQ 15 attempts to strike a balance between single user performance and overall system concurrency. This is achieved through a dynamic reallocation of resources at runtime. To illustrate, the first user's query will get all the cores needed for the load or query operation. The second user's query will not wait for the first user to complete, but rather, the engine and optimizer will pull resources away from the first user so that both users can equally consume the CPU resources. The third user's query will experience the same behavior; with all three operations consuming roughly one-third of the system each. When an operation comes to completion, the resources are freed up and given back to any other currently executing task that can benefit from more CPU resources.

The following diagram illustrates this process:



HARDWARE SIZING AND SERVER CONFIGURATION FOR IQ 15

The increased parallelism and storage management capabilities in IQ 15 allow it to take advantage of more CPU and memory resources than in IQ 12.7. Plus, it allows IQ to utilize tiered storage effectively.

For more 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 IQ White Paper: [“A Practical Hardware Sizing Guide for Sybase IQ 15”](#).

Also, see the later sections of this document for a few additional sizing tips such as “Sizing the New Shared Temporary DBspace”, “Proper Data-Sizing” and “Correctly Sizing the Coordinator”.

SERVER-LEVEL CHANGES FOR IQ 15

CATALOG CHANGES IN IQ 15

- Sybase IQ uses 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 15 were such that the IQ catalog needs to be rebuilt for IQ 15. Therefore, if you are upgrading from IQ 12.7 to IQ 15, 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. For more details on migrating from IQ 12.7 to IQ 15, please see the section on Migration Tips within this document, and the [Sybase IQ Installation and Configuration Guide](#) for your version of IQ and operating system.

NEW RECOMPILE CLAUSE FOR ALTER PROCEDURE/ALTER FUNCTION

- In IQ 15, the alter procedure/alter function now includes a "recompile" clause. Use this clause to force the recompilation of a stored procedure.

TIP: RECOMPILE STORED PROCEDURES WHEN A SCHEMA CHANGES

- The recompile function is useful when the schema of a table has changed. When recompiled, the object definition stored in the catalog is re-parsed and the syntax is verified.
- For more information, please see the [ALTER PROCEDURE statement](#).

REPEATABLE DDL

- When creating an object, you can now use the "CREATE OR REPLACE" command. This enables you to repeat the same "create" statement without changing your code. Therefore, you no longer need to precede a "create" statement with the "DROP...IF EXISTS" statement. For example:

- You can do this:

```
CREATE OR REPLACE PROCEDURE ....  
BEGIN  
...  
END ;
```

- Instead of this:

```
DROP <object_name> IF EXISTS  
CREATE PROCEDURE ....  
BEGIN  
...  
END ;
```

CUSTOM COLLATIONS HAVE BEEN DEPRECATED IN IQ 15

- The creation of custom collations is NO longer supported in IQ 15.
- If you have created a database with a custom collation, you can use the iqunload migration utility to upgrade the database to IQ 15 and preserve the custom collation.

CHANGES TO VIEW DEPENDENCIES AND ALTERING SCHEMAS

- View dependencies have been changed in IQ 15.2. When you have a view referencing another table or view, the link between both objects is now 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 above, 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
```

CHANGES TO THE REQUEST LOG (ZRLOG FILE)

THE REQUEST LOG CONTENT HAS CHANGED

- 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 [SQL Anywhere Server - Database Administration: Request logging](#).
- In IQ 15, the content of the request log file (sometimes called the zrlog file) has changed. If you have a monitoring process on this file, you will need to make the changes described below.

THE NEW REQUEST LOG MONITORING PROCEDURES

1. Generate the Request Log:

- As in IQ 12, 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 new 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 done 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 STATEMENTS

- 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>');`

STORAGE CHANGES FOR IQ 15

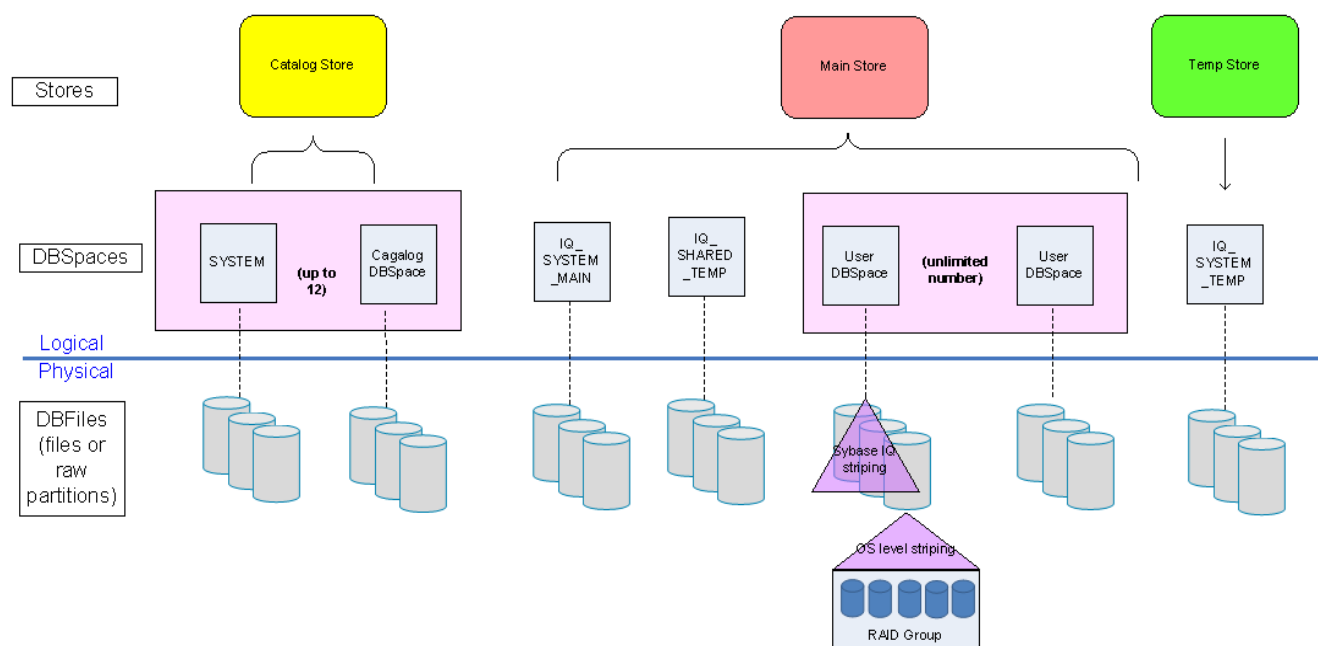
OVERVIEW OF DBSPACES AND DBSTORES

- IQ uses a single database in which data is organized into logical “stores”.
- An IQ store is comprised of a logical container called a DBspace. A DBspace is created by mapping it to a physical database file system (often called a DBfile) that can be either a flat file or a raw device.
- IQ uses three types of logical data stores for storing persistent data:
 1. **The Catalog Store:** This store contains the SYSTEM DBspace and up to twelve additional catalog DBspaces.
 2. **The IQ Main Store:** This store contains the IQ_SYSTEM_MAIN DBspace and other user DBspaces.
 3. **The IQ Temporary Store:** This store contains the IQ_SYSTEM_TEMP DBspace.

DBSTORE CHANGES

STORES CAN NOW BE COMPRISED OF MULTIPLE DBSPACES

- In IQ 12.7, a store was limited to a single file system (either a flat file or raw device). That is, a store could only be comprised of a single DBspace, which then could be mapped to only one physical database file. In IQ 15, this has changed. A store can now be comprised of multiple DBspaces, which can then be mapped to multiple physical database files (either flat files or raw devices).
- With the licensed VLDB option, users may define an unlimited number of DBspaces for user data.
- Without the VLDB option, users may define a single user data DBspace with multiple DBFiles within it.
- In IQ 15.3, there is an additional DBspace called the shared temporary store. It is used to support the new distributed query processing (DQP) feature (see [Distributed Query Processing](#) section).
- Here is a depiction of the storage layout for Sybase IQ 15:



- When a database object is written to a DBspace and the DEFAULT_DISK_STRIPING option is 'ON', Sybase IQ automatically distributes data across all available DBfiles in the DBspace. Users may also set up disk striping across RAID groups at the operating system level.
- The DBspace and file free list are maintained in IQ_SYSTEM_MAIN. Versioning space, some node to node communication, the TLV replay, and other system housekeeping are also performed in this DBspace.
- By default, 20% of IQ_SYSTEM_MAIN is reserved for these housekeeping purposes.

TIP: USE THE READ-ONLY/READ-WRITE FEATURE FOR DBSPACE ADMINISTRATION.

- A user DBspace can be either *read-only* or *read-write*. A DBfile can also be marked as read-only or read-write. With this separation, database administration becomes more efficient due to the following:
- DBAs only need to backup once.
- DBAs can backup and restore read-only data independently of read-write DBspaces.
- DBAs can restore read-only data while the system is operational.
- Data validation can be performed on just the read-write portions of a database.
- DBAs can manually mark a DBspace as being online or offline. If a DBspace becomes inaccessible or unusable, the DBA can tell IQ to ignore this space by marking it “offline”. IQ will ignore this space until is marked “online” again.
- Note: Currently, a virtual backup does not perform a selective backup and restore. It is an all-inclusive, full backup only.

TABLE PARTITIONING IN IQ 15 (USING THE VERY LARGE DATABASE OPTION)

- In IQ 15, the Very Large Data Base Option (VLDB) provides partitioning and placement features. A table can be partitioned by specifying a column partition key, and specifying ranges of values of that key. Data within each range of values is relegated to a partition. At this time, range partitioning is the only partitioning scheme that is supported. A data partition includes the base FP (Fast Projection index in IQ) data only, and not the other indexes that exist along with that data.
- 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.
- A partitioned table can be altered to:
 - Add/drop partitions.
 - Split a partition.
 - Merge adjacent partitions.
 - Un-partition the table.

TIP: OPTIMIZE DATABASE OBJECT PLACEMENT

- After partitioning, an administrator can place a database object (such as a table, table partition, column, or index) within a particular DBspace; or move a database object into a different DBspace. With flexibility of data positioning, frequently accessed data can be assigned to faster storage, and less frequently accessed data can be assigned to cheaper, slower storage. This strategy controls storage costs, while still delivering performance.

UNDERSTANDING QUERY OPTIMIZER CHANGES IN IQ 15

QUERY ENGINE IMPROVEMENTS

IQ 15 performs more query processing in parallel than in previous versions. This means more threads will be allocated to a particular task in IQ 15 than in IQ 12.7. More threads mean more memory, not only from main cache and temporary cache but from the operating system as well.

IMPROVED TOKENIZATION

- FP index tokenization has been implemented for columns with more than 64K distinct values. This is called an FP(3) index, and it is structurally similar to the FP(1) and FP(2) indexes. The maximum size of the FP(3) lookup table is 16777216, and each lookup key requires 3 bytes of storage. The creation of an FP(3) index is permitted only if the space used by the lookup table is less than the current value of the FP_LOOKUP_SIZE database option, and less than the portion of the main cache specified by the current setting of the FP_LOOKUP_SIZE_PPM database option.
- The FP(3) index can offer significant performance improvement for queries using extremely large tables with high unique cardinality

MULTICOLUMN HG INDEXES

This feature boosts query performance, and it is a new multicolumn HG index for ORDER BY queries that reference multiple columns.

COMPONENT INTEGRATION SERVICES

- Sybase IQ uses Component Integration Services (CIS) to query tables on remote servers. Changes in Sybase IQ 15.2 allow queries with proxy tables or IN SYSTEM tables to execute significantly faster than earlier versions when the amount of IN SYSTEM or proxy table data is small compared to the IQ data:
 - Statements that benefit from these changes are:
 - SELECT, SELECT INTO; and,
 - INSERT...SELECT statements only.
- Queries that do not benefit from these changes have the following characteristics:
 - A subquery in a CASE expression.
 - LOB's (columns of LONGBINARY, LONGVARCHAR, LONGNVARCHAR, LONGBITSTRING, or XML) on remote tables.
 - A function not supported in the remote node (for example, "bit or" or user defined functions).
 - Queries with global variables.

TRANSACTION OUT-OF-SPACE BEHAVIOR

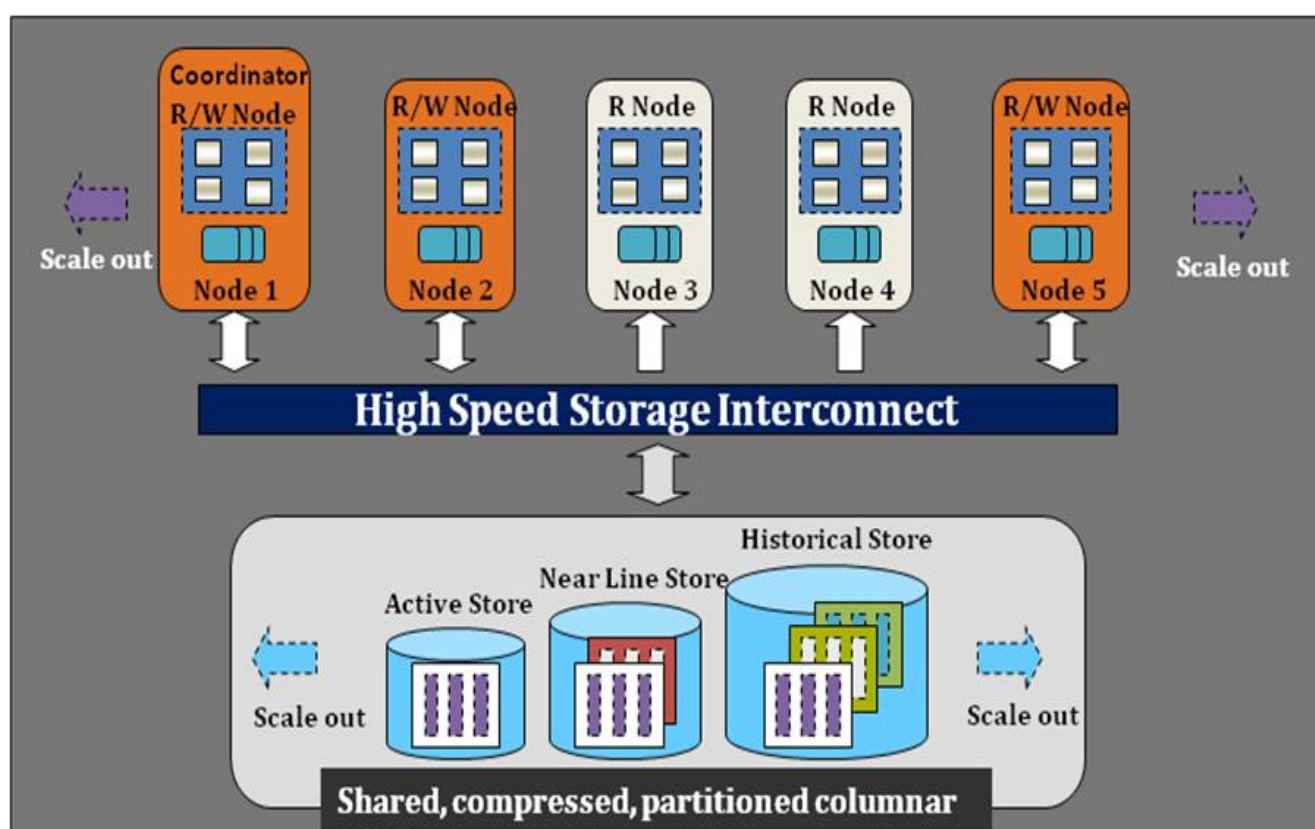
- Space management for the IQ main store and the temporary store has improved. In IQ 15, when a transaction runs out of space, IQ will terminate and roll back the transaction, rather than suspend it. Additional space can then be added to the main store or temporary store as needed.

MULTIPLEX CHANGES FOR IQ 15

A Sybase IQ Multiplex is a group of Sybase IQ servers that connects to a central store for permanent shared data. Each server maintains its own local storage for catalog metadata, temporary data, and transaction logs.

MULTIPLE READER AND WRITER NODES

- In IQ 12, only one node in the Multiplex was allowed to modify shared IQ data. In IQ 15, the IQ Multiplex now includes a single coordinator, plus additional multiple writers and additional multiple readers. This allows for the distribution and parallelization of read-write DML operations (INSERT, LOAD, UPDATE, DELETE) on shared IQ tables.
- Below is a graphical example of this architecture:



TIP: YOU CAN HAVE MULTIPLE WRITER AND READER NODES

As with IQ 12, an object can have only one user change its data or structure at the same time. However, in IQ 15 it is now possible to architect a system with multiple writer and reader 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.

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.
 - With the implementation of distributed query processing in IQ 15.3, the coordinator is also the only node with the ability to add DBFiles to the IQ_SHARED_TEMP DBspace, and administer logical servers.

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.

OBSOLETE MULTIPLEX FEATURES

- The following IQ 12.7 features and dependencies were discontinued in IQ 15:
 - There are no Local Stores in IQ 15.
 - There are no shared IQ store path aliases.
 - There is no SQLRemote dependency.

SIMULATING LOCAL STORE FUNCTIONALITY IN IQ 15

LOCAL STORES HAVE BEEN DEPRECATED IN IQ 15

- One of the important changes in the IQ 15 Multiplex is the absence of query server *local stores*. Therefore, if you are migrating from an IQ configuration with local stores, you will need to transfer the data from your local stores into IQ 15's main shared store. The *iqunload* 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 15.

HOW TO SIMULATE A LOCAL STORE IN IQ 15

- Here is a sample process for simulating "local store" type functionality in IQ 15:
 - 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) AND SQL REMOTE

- In IQ 12, communication between secondary nodes and the writer node was handled by the SQLRemote process. In IQ 15 and later, the communication between the coordinator node and the secondary nodes (readers and writers) is now handled by a new, Inter-Node Communication (INC) layer. 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 15 introduces a new performance feature called Distributed Query Processing (DQP). This feature allows IQ to distribute the processing of a query across multiple server nodes in a multiplex configuration. In order to support streamlined communication among nodes participating in DQP, IQ 15.3 also introduces the Multiplex Inter Node Communication (MIPC) framework. The MIPC mesh is a peer-to-peer inter-node communication infrastructure that supplements the Inter Node Communication (INC) protocol added in IQ 15.0. 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 pairing at this time.
- During our internal testing, we found one particular instance where a private MIPC network provided significant performance benefits over a shared MIPC network: A DQP query running on a two-node configuration using a private MIPC network performed almost the same as a three-node configuration over a shared MIPC network.

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.

MULTIPLEX DO'S

- All multiplex nodes must be at the same ESD level. IQ 15 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 on the IQ_SYSTEM_MAIN DBspace, you must shut down all secondary nodes. Once these 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.
- In 15.3, it is recommended that you ensure that there are no active connections on 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)

- Prior to IQ 15.3, the architecture of Sybase IQ was such that an application would connect to a single node in the Multiplex and run the user's workload against that one node. This allowed the application and database administrators to have tight controls over the IQ Multiplex and direct certain users or applications to specific hosts for true workload segregation. Should that node become unavailable, though, it was up to the application and/or user to connect to another functioning Sybase IQ node.
- IQ 15.3 mitigates the problem by introducing the PlexQ™ Distributed Query Platform, a Massively Parallel Processing (MPP) architecture that accelerates highly complex queries by distributing work to many computers in a Multiplex configuration. Distributed Query Processing (DQP) 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 Sybase IQ, and is enabled by default. In general, any query that qualifies for parallel processing also qualifies for DQP in IQ 15.3 and above when this feature is enabled.
- Refer to the "[Using Sybase IQ Multiplex](#)" user guide, and the Sybase technical white paper "[Scaling Out Query Performance with Sybase IQ 15.3](#)".

LOGICAL SERVERS

- Logical Servers enable the grouping and presentation of a subset of multiplex computer resources as one logical entity.
- Multiple logical servers can be created to serve different groups of applications or users. You can also 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, Sybase IQ distributes the query execution (in parallel) to those nodes that are members of the logical server.

SIZING THE NEW SHARED 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 worse 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. See the next section on "Distributed Query Processing" for more details.

FULL TEXT SEARCH IN IQ 15

- Sybase IQ 15.2 introduced the Unstructured Data Analytics (IQ_UDA) option. This option provides the means to store and retrieve unstructured data objects from the IQ store. IQ's "full text" search capability enables users to search for both words and phrases, perform Boolean and proximity searches, and score results based on relevance. Sybase IQ uses TEXT indices to store positional information for terms in an indexed column. TEXT indexes are created using settings stored in a text configuration object. Here is an example of creating a TEXT index using a SQL command:

```
CREATE TEXT INDEX myTxtIdx ON Customers ( CompanyName ) CONFIGURATION default_char
```

- Sybase IQ 15.2 ESD #2 added support for NGRAM TEXT indices. An NGRAM is a sequence of variable characters that stands for a word or string of words. Fuzzy search capability over an NGRAM TEXT index can be used to search for misspellings or variations of a word.
- Sybase IQ can use external pre-filter and term-breaker libraries to pre-filter and tokenize documents during index creation or query processing. External pre-filter and term-breaker libraries must be provided by a Sybase-certified partner. Currently the only certified partner is ISYS. The ISYS Document Filters for UDA enable a user to create text indices against unstructured data stored in the IQ store. IQ text indices then facilitate the searching of unstructured data using SQL. ISYS document filters support most common document formats such as Microsoft Word (DOC, DOCX), Excel (XLS, XLSX), PowerPoint (PPT, PPTX), Adobe PDF, WordPerfect, Rich Text Format (RTF), Open Document Format (ODF), HTML and many others.
- "Full text" searches are performed using the CONTAINS clause in the FROM clause of a SELECT statement, or by using the CONTAINS search condition (predicate) in a WHERE clause, e.g.:

```
SELECT ID, ct.score, Description  
FROM MarketingInformation  
CONTAINS ( MarketingInformation.Description,  
          'stretch* | comfort*' )  
AS ct ORDER BY ct.score DESC;
```

- For additional resources that discuss Sybase IQ "full text" search:
 - [*"Using Sybase IQ as a Text Search Engine for ASE"*](#).
 - [*"SYS Prefilter 9.7 plug-in running in Sybase IQ 15.2, supporting Full Text Search"*](#).

MIGRATION AND UPGRADE TIPS FOR IQ 15

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 Sybase installation is the review of Sybase'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 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 Sybase software.

IQ VERSIONS

Whenever possible, Sybase recommends that you migrate to one of the most recent versions of IQ. Currently, the two most recent versions of IQ are 15.3 and 15.4. Consider migrating to one of these two versions.

MS WINDOWS AND THE 32-BIT VERSION OF IQ

As of IQ 15.4, IQ only supports 64-bit servers. Therefore, you will only find a 64-bit version of IQ 15.4 on the Sybase Product Download Center (SPDC) for MS Windows. You will no longer find a 32-bit version of IQ 15.4 or above for the MS Windows (or ANY other) OS platform.

BACKUP BEFORE AN UPGRADE OR MIGRATION

- Sybase **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 SYSDBSACES.

- Also, save the lengths of the following files:
 - The IQ_SYSTEM_TEMP DBspace file, typically named dbname.iqtmp
 - 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 Sybase IQ System Administration Guide.](#)

BACKUP AFTER AN UPGRADE OR MIGRATION

Although optional, Sybase 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 can not 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 A MIGRATION

- Sybase 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 12.6 or 12.7 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 [System Administration Guide: Volume 1 > System Recovery and Database](#).

MIGRATING FROM IQ 12.7 TO IQ 15 (UNLOAD AND RELOAD THE CATALOG)

IQ CATALOG CHANGES

- IQ uses a 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 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. And, in IQ 15.4, the engine was SA 12.0.1.

USE THE IQUNLOAD UTILITY TO MIGRATE FROM IQ 12 TO IQ 15

- Between IQ 12.7 and IQ 15, the changes in SA were such that upgrading a database from IQ 12.6/ 12.7 to IQ 15.x is very different than upgrading from one version of IQ 12 to another version of IQ 12. Upgrading from IQ 12.7 to IQ 15 requires an unload/reload of the catalog.
- This is accomplished by using the “iqunload” utility. Note: Other existing DBspaces are unaffected and do not need to be reloaded.
- For more details on using the IQUNLOAD utility, please see the *“Database File Migration”* section of the Sybase IQ Installation and Configuration Guide for your operating system and IQ version. For example, if you are upgrading to IQ 15.3 on Linux, you would go to: [“Upgrading a Sybase IQ 12.6/12.7 Database” section in the Sybase IQ 15.3 Installation and Configuration Guide for Linux.](#)

TIPS FOR THE IQUNLOAD UTILITY

This section presents several tips for using the IQUNLOAD utility to migrate an IQ 12.6 or 12.7 database to an IQ 15 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 Sybase IQ 15.x 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 Sybase IQ 15.x 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.

SAMPLE TASK LIST FOR PERFORMING A MULTIPLEX MIGRATION FROM IQ 12 TO IQ 15

Below, is a summary of the tasks for a Multiplex migration from 12 to 15. The purpose of this list is to emphasize where to use the “-iqro 1” switch. This is NOT a detailed task list.

- Please see the following documents for a detailed list of migration procedures:
 - For multiplex upgrades, see: [Sybase IQ 15.4 > Installation and Configuration Guide for AIX > Database Upgrades > Upgrading a Sybase IQ 12.6/12.7 Database>Migrating Multiplex Databases.](#)
 - For simplex upgrades, see: [Sybase IQ 15.4 > Installation and Configuration Guide for AIX > Database Upgrades > Upgrading a Sybase IQ 12.6/12.7 Database>Migrating Simplex Data.](#)

Task List for Multiplex Migration from IQ 12 to IQ 15

1. [Synchronize and Shutdown Multiplex Query Nodes](#): Check your multiplex server log files for synchronization problems.
2. [Migrate Your Local Stores](#): This step is only required if you created local stores for your 12.x multiplex query nodes. In IQ 15.4, the iqunload utility connects to a Sybase IQ 12.7 database with a Sybase IQ local store and creates scripts that unload its schema and data.
3. [Start the Multiplex Write Server in Single Node Mode](#):
To ensure a smooth multiplex migration, start the write server in single node mode to clean internal state information. To do this, start the writer node in single node mode, shutdown, then start again normally, and then shut down again.
4. [Verify the Integrity Of Your Pre-Migration Database](#): Start your Sybase IQ 12.7/12.6 database, then use sp_iqcheckdb to verify that your database is clean and error-free. Repair any index and allocation problems.
5. [Backup your database](#): Follow your normal procedures for backing up your IQ server.
6. [Use the iqunload Utility to Migrate Your Database](#):
Run the iqunload utility with the appropriate parameters to migrate your databases. Make sure that the database file is not in use, and run the iqunload utility. NOTE: The database and the iqunload utility must be on the same machine to migrate the database, or iqunload returns an error. For example, to migrate the database "mytest" save the output in the unload.out file in the current directory:

```
iqunload -au -c "uid=DBA;pwd=SQL;dbf=mytest" -o unload.out
```
7. [Start and Verify the Coordinator](#):
 - a. Start your IQ 15 write server (with its migrated database) in **READ-ONLY (-iqro) mode**. Also use the **-iqmpx_sn**, and **-gm** switches to start the server in single node mode with only one connection. For example:

```
start_iq @iqdemo.cfg iqdemo.db -iqro 1 -iqmpx_sn 1 -gm 1
```
 - b. Next, verify your migrated database by running the sp_iqcheckdb procedure.
8. [Manually Synchronize the Secondary Nodes](#): Synchronize the secondary nodes from the coordinator node.
9. [Start the Secondary Nodes](#): Start the secondary nodes, then import the schema and data for the local stores into the coordinator.
10. [Set the Failover Node](#): Set the failover mode for the failover node, if you have one.
11. [Complete any necessary post-migration tasks](#). See "Post-migration tasks" for more information.

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 OR JAVA 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 15 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 OVERFLOW ERROR

- During IQ 15 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 SQL Anywhere level, where the SA server manages connections and parses commands before the query is passed to IQ. In IQ 15, 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 15.
- 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. The 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 15 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 15 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 IQ 15 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 can not rebuild the IQ 15 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 SQL Anywhere database implementation, but in Sybase IQ, its effect is as described above.

UPGRADING FROM ONE VERSION OF IQ 15 TO ANOTHER (ALTER DATABASE UPGRADE)

- The changes in SA between different versions of IQ 15 do NOT require an unload/reload of the IQ Catalog.
- For example: Migrating from IQ 15.1 to IQ 15.4 only requires running the ALTER DATABASE UPGRADE command.
- Please review the latest Release Bulletin, Cover Letter and Installation and Configuration Guide for the version of IQ you are migrating to.

USE THE "-IQRO" SWITCH WHEN PERFORMING THE ALTER DATABASE UPGRADE COMMAND

THE "-IQRO" SWITCH

It is critical that you use of 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.

EXAMPLE OF USING THE "-IQRO" SWITCH IN A SIMPLEX UPGRADE

- Follow these steps to upgrade a simplex database from one version of IQ 15 to another. For example, from IQ 15.2 to IQ 15.4:
 1. Disconnect all users from the server.
 2. Back up the database with the version of Sybase IQ you currently use.

3. Shut down the server. If the server aborts or hangs during shutdown, stop and do not proceed to the next step. Instead restart the database with the IQ version you are currently using and shut down the server. Proceed to the next step only on a clean shutdown.
 4. Start the server in READ-ONLY mode (and with one connection): Restart the IQ 15.x server you are migrating with these startup flags: *-iqro* and *-gm 1*. For example:

```
start_iq @iqdemo.cfg iqdemo.db -iqro 1 -gm 1
```
 5. Start Interactive SQL and connect to the database.
 6. Use the ALTER DATABASE UPGRADE statement to upgrade the database.
 7. Run sp_iqcheckdb ('allocation database') to verify that there are no errors.
 8. Shut down and restart the server normally (without the *-gm 1* and *-iqro 1* startup flags).
 9. Back up the database.
- For more details on the upgrade procedure, see the “Installation and Upgrade” section of the Sybase IQ 15.4 Release Bulletin for your platform. For AIX, this would be: [Sybase IQ 15.4: Release Bulletin for IBM AIX: Installation and Upgrade](#).
 - For more details on using the Alter Database command, see the [ALTER DATABASE statement in the Sybase IQ 15 Reference: Statements and Options > SQL Statements](#).

EXAMPLE OF USING THE “-IQRO” SWITCH IN A MULTIPLEX UPGRADE

- Run the following procedure to upgrade a 15.x multiplex database to another version of 15.x:
 1. Disconnect all users from the servers.
 2. Back up the database with the version of Sybase IQ you currently use.
 3. Shut down all nodes in the multiplex.
 4. If the server aborts or hangs during shutdown, stop and do not proceed to the next step. Instead restart the database with the IQ version you are currently using and shut down the server. Proceed to the next step only on a clean shutdown.
 5. Use IQ 15.3 to restart the coordinator with startup flags *-iqmpx_sn 1*, *-gm 1* and *-iqro 1*. For example:

```
start_iq @iqdemo.cfg iqdemo.db -iqmpx_sn 1 -iqro 1 -gm 1
```
 6. Start Interactive SQL and connect to the database.
 7. Use an ALTER DATABASE UPGRADE statement to upgrade the database.
 8. Run sp_iqcheckdb ('allocation database') to verify that there are no errors.
 9. Shut down and restart the coordinator normally (without the *-gm 1* and *-iqro 1* startup flags).
 10. Synchronize and restart all multiplex secondary servers.
 11. Back up the database.
- For more details on this procedure, see the “Installation and Upgrade” section of the Sybase IQ 15.4 Release Bulletin for your platform. For example, on the AIX platform: [Sybase IQ 15.4: Release Bulletin for IBM AIX: Installation and Upgrade](#).

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 affect 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 PRE-IQ 15.4 FP INDEXES FOR ENHANCED COMPRESSION

- In IQ 15.4, an enhanced compression algorithm was introduced for fixed length data types (Enhanced Compression). This new compression routine is automatically applied to all newly added values.
- NOTE: This new compression routine is NOT automatically applied to existing (pre-IQ 15.4) data.
- Therefore, in order to compress pre-15.4 data using IQ 15.4'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 in IQ 15.4 or below.
 2. Move the data to another DBspace using the `ALTER TABLE ... MOVE` statement.
 3. Upgrade the database using the `ALTER DATABASE UPGRADE` statement.

LICENSING CHANGES FOR IQ 15**LEGACY LICENSING**

With the release of IQ 15, a new licensing model was introduced. Consequently, provisions were made to allow pre-IQ 15 customers to continue using their existing licensing model. In order to achieve this, a clear designation was made between customers using new, old, and interim licenses. Therefore, a new license type was created to identify pre-IQ 15 licenses called a 'Legacy License'. For example, when pre-IQ15 customers log into the Sybase Product Download Center (SPDC), they will see something like this:

Version	Description	Date	Available
15.4	Sybase IQ Enterprise Edition (Legacy) 15.4 for Linux x86-64 - 64bit	Jul 9, 2009	Download Log

What does the “Legacy” designation mean? If this was a pre-IQ 15 customer who already had an Enterprise Edition license with the Multiplex License Option, the following entitlements would be generated when creating a new license:

- **Core Entitlements:** The pre-IQ 15 entitlements for the existing number of core processors (CP's) would be automatically converted to the new license with no changes or additional charges. For example, if this site already had a pre-IQ 15 4-core entitlement, they would continue to get a 4-core entitlement for IQ 15.
- **Multiplex Option:** The existing Multiplex License would be converted to the new Sybase IQ Multiplex Grid License with the same number of nodes that were previously licensed at no additional cost. NOTE: If additional secondary nodes are added to the Multiplex at some latter time, then additional IQ_MPXNODE licenses will need to be purchased.

NEW LICENSING OPTIONS FOR IQ 15

In addition to the IQ Core license, IQ 15 includes a number of optional licenses that extend IQ's functionality. The following license options and changes were introduced in IQ 15.0 and above:

IQ 15.0:

- **IQ_ENC:**
 - This option was removed and included in the IQ_SECURITY license. (It was originally Introduced in IQ 12.6 and 12.7).
- **IQ_MPXNODE:**
 - Each secondary node in a multiplex configuration is required to have this license.
- **IQ_VLDBMGMGT:**
 - Allows for more than one user DBspace per server.
 - An IQ_VLDBMGT license is required for every 1TB of data.
 - Note: In IQ 15.1, the VLDBMGT license added table level partitioning, and the IQ_TBSIZE option was removed as a separate license (see below).

IQ 15.1:

- **IQ_UDF** (“In-Database Analytics-Partner Solution”):
 - This option introduced in-database user defined functions (UDFs) for Sybase Certified Partners only.
- **IQ_VLDMGMT/IQ_TBSIZE:**
 - Removed the IQ_TBSIZE option and combined its features into the IQ_VLDBMGMT option.
 - Added table level partitioning.
 - If any of the following conditions were true, then an IQ_VLDBMGMT license was now required:
 - Storage greater than 1TB.
 - More than one user DBspace (in addition to IQ_SYSTEM_MAIN).
 - Table level partitioning.
- **Small Business Edition License:**
 - The license model for this option was changed from being based on the number of cores, to being based on the number of sockets used by the IQ server.
 - Note: A “socket” in this case was defined as a physical connector on a computer motherboard that accepts a single physical chip (CPU). A chip can have one or more cores.

IQ 15.2:

- **IQ_UDA** (Unstructured Data Analytics):
 - Enables the IQ Text Search functionality which allows searching unstructured and semi-structured data.
 - Includes the IQ_LOB license.
 - Applies to IQ TEXT indexes and combines the previous LOB and Full-Text Search options.

IQ 15.3:

- **Distributed Query Processing (DQP):**
 - Improves performance in IQ multiplex configurations by spreading query processing across multiple nodes in the cluster.
 - Unless disabled, DQP occurs automatically for qualifying queries when:
 - The server is part of a multiplex where servers have established MIPC connections. (Servers establish these connections automatically on startup. To verify, run `sp_iqmpxinfo`).
 - The shared temporary DBspace has writable files available.
 - The Logical Server of the current connection has at least one other member node available. Note: Logical Server configuration allows you to control which multiplex nodes participate in distributed query processing.
- **InfoPrimer:**
 - Formerly known as Sybase ETL. This utility is now a separately packaged and licensed option.
- **IQ_MPXNODE:**
 - The Multiplex license now includes the new DQP (Distributed Query Processing) feature as default functionality.

IQ 15.4:

- **IQ_IDA:**
 - This new option enables any customer to build, deploy and run their own V4 C/C++ User-Defined Functions (UDF) within IQ. This option is available for purchase by any Sybase IQ customer. Previously, this functionality was available only to Sybase Certified Partners who purchased the IQ_UDF option (introduced in 15.1).
 - This option also requires the purchase of a multiplex license (IQ_MPXNODE) because runtime execution of UDFs is restricted to multiplex reader nodes only.

- Failover Considerations for the IQ_IDA Option:
 - SYSAM checks the IQ node type when bringing the UDF license online. If it detects a writer or coordinator node, it will not enable the UDF license since runtime execution of UDFs is restricted to multiplex reader nodes under this license.
 - Therefore, check your failover logic and be sure that a reader node NEVER changes role to become a writer or coordinator node in a UDF implementation. If a failover results in a reader node being promoted to a writer or coordinator node, all subsequent UDF queries will fail.
- Certified Sybase Partner Considerations for IQ_UDF:
 - A Partner UDF license is still required to use V3 C/C++ libraries.
 - Partners can continue to use their current IQ_UDF license for existing V3 C/C++ libraries.
 - A Partner UDF license will continue to support running UDFs on simplex nodes.
 - An IQ_IDA license is required to run the new V4 C/C++ libraries.
 - Partners who have a UDF license can also obtain the new IQ_IDA license at no additional cost.
- Sybase IQ Express Edition License (IQ_XE license):
 - This is a new, no-cost license.
 - It supports all the options and features available in the Evaluation Edition. However, unlike the previous evaluation licenses, the XE license does not expire at the end of 30 days.
 - However, this license imposes the following restrictions:
 - The aggregated size of all IQ Main DBspaces is restricted to a size of 5GB or less.
 - You are unable to add any additional license options.
 - Sybase IQ Express Edition is not eligible for production deployment and is not supported by Sybase Customer Service and Support.

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 "Evaluation License":
 - 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.Imp 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

Sybase uses a utility called SySAM (Sybase Software Assets Manager) to install, maintain and verify Sybase product licenses. In version 15 of all Sybase products, we introduced SySAM2. This version significantly changes the way Sybase manages software licenses. Below, is a discussion of SySAM's new features in IQ 15, and how they affect the way you use Sybase IQ. For more details, please review the "Licensing Your Software" chapter in the "Installation and Configuration Guide" for the platform of your choice.

SYSAM IS NOW REQUIRED FOR ALL LICENSES

In pre-15 versions of Sybase software, the use of SySAM was optional for all "core" licenses. SySAM was only required when using optional licenses that extended IQ's core functionality. In version 15, this has changed. SySAM license registration is now required for ALL licenses. This is accomplished using a new version of the SySAM license manager called SySAM2.

NEW FEATURES

ENHANCEMENTS

- SySAM2 provides several enhancements. These include:
 - Use of the "sp_iqlmconfig" stored procedure to control license management configurations instead of running sybinstall with the "-add_license" option.
 - Use of the "sp_iqlmconfig" stored procedure to control license management configurations instead of running sybinstall with the "-add_license" option.
 - The ability to dynamically add licenses while the server is running.
 - The ability to maintain all IQ licenses in one location (by using the "Served License" implementation).

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 new license manager property file that is created when you install a new IQ 15 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 15.4, the iq.default.lmp file is located in the \$SYBASE/IQ-15_4/SySAM directory.

- The .lic file:
 - This is the license file you create on the SPDC (Sybase Product Download Center) Web Site.
 - 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, under certain circumstances in IQ 15, 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 (“.Imp”) 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 “.Imp” 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 .Imp 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 .Imp 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 .Imp file is in your \$SYBASE /IQ-15_4/SySAM directory. You can also use the output from the Imutil (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 “*Troubleshooting SYSAM*” appendix in the “*Installation and Configuration Guide*” for your particular platform. For example: [Sybase IQ 15.4 > Installation and Configuration Guide for AIX: Troubleshooting SySAM](#).

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
```

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 above, 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.

OPERATIONAL MANAGEMENT

IQ ADMINISTRATION TOOLS

- IQ can be monitored and administered in several ways:
 1. By using one of Sybase's GUI Server Administration tools:
 - Sybase Central (SC): Sybase's long-standing, Java-based administration GUI tool.
 - Sybase Control Center (SCC): Sybase's new, Internet based GUI tool.
 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 Sybase IQ.

SYBASE ADMINISTRATION GUI'S

- Sybase GUI tools consist of Sybase Central and Sybase Control Center. These tools allow you to manually perform most administration functions (such as adding users and database objects) as well as viewing a variety of statistics.
- IQ statistics include:
 - CPU usage statistics.
 - Memory usage statistics.
 - Cache statistics.
 - Thread statistics.
 - Connection statistics.
 - Request statistics.
 - Transaction statistics.
 - Store I/O statistics.
 - DBspace Usage.
 - Network Statistics.

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 MANAGEMENT

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.

SYBASE CENTRAL (SC)

Sybase Central (SC): This is Sybase's Java-based, legacy GUI tool. Here are some important notes about Sybase Central for IQ 15:

- The Sybase Central version between 12.7 and 15 has changed from v4.3 to v6.0.
- Sybase Central now supports JRE 1.6.0.
- The .scRepository is now renamed to .scRepository600.
- The location of the .scRepository600 on Windows is now in the directory where the ALLUSERSPROFILE environment variable is pointing to. On UNIX, it is still in the directory where SC is installed.
- All the jar files pertaining to Sybase Central have "600" appended to their name. For example, sybasecentral.jar is now sybasecentral600.jar.
- The new topology view for Multiplex servers not only displays the servers graphically, but also gives you the option to add/remove servers, include/exclude servers or change the designated failover node.
- Using the new Performance Monitoring tab for all IQ servers, you can customize the statistics you want to view and also vary the collection rate. In a Multiplex setup, you can view the stats for all, some or one of the servers at the same time.
- You can view the 'Server Messages and Executed SQL' for the servers you are connected to in SC. This option gives the user the ability to see exactly what SC is doing behind the scenes, thereby assisting with problem debugging.
- The improved Application Profiling wizard allows you to customize what you want to profile. Application profiling allows you to profile stored procedures, functions, triggers and events. It makes recommendations on how to improve performance and captures database activity.
- Connection Profiles now have Import and Export options.
- The Sybase Central plug-in version for IQ 15.4 is v6.1.
- This new plug-in only supports IQ servers at version 15.2 ESD # 3 and higher.
- This version of SC has been enhanced to support the following new features introduced in SA 12, including:
 - Spatial Reference Systems.
 - Units of Measure.
 - Sequence Generators.
 - Permission Editors.
 - Unique Indexes WITH NULLS NOT DISTINCT.
 - Test Configuration Object External Term Breakers/Prefilters.
 - Wizards show execution SQL.

- In 15.4, the Sybase Central Plug-in only uses ODBC connections and no longer supports jConnect.
- The repository filenames have also been changed in 15.4: On 32-bit computers it is .scRepository610_32. On 64-bit computers it is either .scRepository610_32 or .scRepository610_64 depending on the IQ installation.

SYBASE CONTROL CENTER (SCC)

Sybase Control Center (SCC): This is Sybase's new, web-based GUI for managing ASE, IQ and RepServer (as well some mobility products) within a single tool. Here are some important notes about Sybase Control Center for IQ 15.

SCC ROADMAP

- The goal of SCC is to provide a consolidated interface for managing and monitoring Sybase products from any location within a single thin client delivered via an Internet browser. This includes tools for monitoring real-time availability and performance, plus intelligent tools for identifying performance and usage trends.
- Although SCC for IQ does not currently provide all the functionality provided by Sybase Central, the goal is to achieve this in the coming releases.

OVERVIEW OF SCC

- SCC is a separately installed and configured server that is downloaded from the SPDC site.
- It can be used to monitor and administer ASE and RepServer as well as IQ.
- Using 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.

HISTORY OF SCC FOR IQ

- SCC for IQ was first introduced in IQ 15.2.1. It was built using SCC 3.1 and the functionality of this release was largely limited to monitoring IQ.
- In IQ 15.3, SCC for IQ was built on SCC 3.2 and introduced basic administration of IQ
- In IQ 15.4, SCC for IQ was built on SCC 3.2.2 and it greatly enhanced the administration features for IQ.
- As of IQ 15.4, SCC for IQ does not support all the features provided in Sybase Central. These features are planned for a future release. The features not yet supported by SCC include the following:
 - Add/Remove/Modify objects (tables, views, stored procedures, triggers or events).
 - Add/Remove/Modify Indexes, Join indexes, remote servers.
 - Profile / Debug database procedures.
- Over time, everything you can do in Sybase Central will be available in Sybase Control Center.

- For more details on how to install and configure SCC, please review the online documents: Sybase Control Center is a new Web-based GUI tool for managing and monitoring IQ. For more details, see [Sybase Control Center 3.2](#) and [Sybase Control Center 3.2.3-Sybase Control Center For Sybase IQ](#).

SCC VERSION RESTRICTIONS FOR IQ

- It is important to note the following:
 - You can NOT administer IQ 15.2.1 and IQ 15.3 or higher on the same SCC server. This is because SCC for IQ 15.3 and above requires version SCC 3.2, and IQ 15.2.1 is only supported by SCC 3.1.
 - However, you CAN administer IQ 15.3 and 15.4 on the same SCC server.

SCC ADMINISTRATION FEATURES FOR IQ 15.3

In 15.3, the Administration and Monitoring tasks you can perform in SCC v3.2 include the following:

- Create, start or stop a server.
- Add/Remove/Modify nodes in multiplex.
- Create/Remove/Modify logical servers, add/remove nodes in a logical server.
- Add/Remove/Modify DBspaces/DBFiles.
- Add/Remove/Modify users/groups/login policies/authorities.
- Monitor status of servers at the node level or Multiplex level using a Heat Chart.
- Gather/view statistics on as needed basis at the node level or cumulatively at the Multiplex level.
- Setup alerts for SCC to notify you when a resource needs attention.
- For more details, see the [Sybase Control Center for Sybase IQ \(V3.2\)](#).

SCC ADMINISTRATION FEATURES FOR IQ 15.4

NEW SCC FEATURES

In 15.4, the additional tasks you can perform in 3.2.3 include the following:

- Administration Console – manage existing resources and create new ones. Column-based filtering lets you display only the objects you are interested in. Select and perform operations on several objects simultaneously.
- Authority management – grants and revokes database authorities to a user or group.
- Automatic logout – configure the logout timer to end users' login sessions after a specified period of idleness.
- Database management – creates databases, view database properties, and view and modify database options.
- DBspace and DB file management – add, modify, view properties, or delete a DBspace. Add, modify, view properties, or delete a DB file.
- Functions– Create, alter and drop functions.
- Java EE Functions (External Environment) Support.

- Logical server management – group a subset of physical hardware resources together as a logical entity that appears as a single multiplex server.
- Login policy management – manage the rules for user login.
- Memory management – use environment variables to control Sybase Control Center’s memory use. A new console command, `info -m`, displays memory usage data.
- Multiple object selection - selects and performs operations on several objects simultaneously.
- Multiplex management – change multiplex server configurations, manage secondary and failover nodes, and configure logical servers.
- Procedures – Create, alter and drop procedures /view procedures and functions.
- Server management – creates, start, stop, view properties, change server configuration, and generate administration scripts.
- Shared temporary DBspace management – manage shared temporary stores to communicate and store temporary on-disk structures for distributed query processing.
- SQL execution window.
- Table UDF & TPF.
- Testing scripts – test the execution of alert-triggered scripts to make sure they work as expected.
- Text configuration objects.
- Text index.
- User and group management – add, change, and delete users, or groups containing users.
- Views – Create, alter and drop views.
- NOTE: As of IQ 15.4, the following is not fully supported (but is planned for a future release):
 - Add/Remove/Modify objects (tables, views, stored procedures, triggers or events).
 - Add/Remove/Modify Indexes, Join indexes, remote servers.
 - Profile / Debug database procedures.
 - For more details. see the [Sybase Control Center for Sybase IQ \(V3.2\)](#) .

TIP: BACKUP THE SCC REPOSITORY

- The SCC repository is located within the following directory: `$SYBASE /SCC-3_2/services/Repository`.
- The SCC repository is located within the following directory: `$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.

USING THE SYBASE CONTROL CENTER

SUMMARY OF PROCEDURES FOR INSTALLING AND STARTING SCC

The following is a summary of the steps required to install, start and register an IQ server to SCC. Once this task is completed, you can administer and monitor your IQ server.

NOTE: The following is not intended to be a comprehensive guide for performing these tasks. For complete instructions on installing and configuring the SCC and SCC for IQ, see the following documents:

For details on installing and configuring SCC, please see the [Sybase Control Center 3.2.1 Doc Set](#).
For details on configuring and using SCC for IQ, please see the [Sybase Control Center for Sybase IQ Guide](#).

1. INSTALLING THE SCC SERVER

- To install the SCC server, you must download the software for this product from the SPDC site at: <http://downloads.sybase.com/swd/base.do>
- The SCC server communicates with the IQ Server via the SCC agent.
- To administer an IQ server, you connect to the SCC server using an Internet browser and an IP address that is established during the SCC server installation process.
- NOTE: It is strongly recommended that you install the SCC Server on a host where IQ, ASE or RepServer is NOT running.

2. STARTING THE SCC SERVER

- Once the SCC server is installed, do the following to start the server:
- Login to the UNIX or Windows machine where your SCC server is installed.
 1. Go to the SCC Server's home directory and set the environment:
 - a. **cd \$SYBASE/SCC32**
 - b. **source \$SYBASE/SCC32/SYBASE.csh**
- Verify your version of the SCC Agent:
 1. Enter the command: **\$SYBASE/SCC-3_2/bin/scc.sh -version**
 2. In IQ 15.4, this returns: **Sybase Control Center Server 3.2.2.4514**
- Start the SCC Server:
 1. Go to the SCC home directory: **cd \$SYBASE/bin**
 2. Run the shell script to start the server on the default port: **\$SYBASE/SCC-3_2/bin/scc.sh**
 3. Optionally, you can specify the port and other settings per the install guide. For example, to specify the port, enter:
\$SYBASE/SCC-3_2/bin/scc.sh --port rmi=9991; \$SYBASE/SCC-3_2/bin/scc.sh -start

- During startup, take note of the configuration messages displayed during the boot sequence, such as the following:

```
Starting Sybase Control Center...
Agent Home:      /iq/rell1540/SCC-3_2/instances/myserver.sybase.com
Agent Version:   Sybase Control Center Server 3.2.2.4514
Node:            myserver.sybase.com(10.22.160.53)
RMI Address:     myserver.sybase.com(10.22.160.53)
RMI Port:        9991 -> This is the port the SCC Server "listens" on and is the port used
                    to connect to the SCC server via your Internet browser.
```

- Once started, you will see the following console prompt: `% scc-console>`
- To verify the server, run the console "status" command: `% scc-console> status`
- This will return a listing similar to the following:

```
Agent Home:      /sybase/iq/rell1540/SCC-3_2/instances/myserver.sybase.com
Connection URL:  service:jmx:rmi:///jndi/rmi://myserver.sybase.com:9991/agent

Status:  RUNNING
----- Services -----
Service Id      Service Name      Is Running
-----
Agent           Agent Service      true
```

- In the listing above, verify that:
 - "Service id" = **Agent**
 - "Service Name" = **Agent service**
 - "Is Running" = **true**
- If your values are different than the example shown above, your local IQ server may be unable to communicate with this SCC Server, and registration of your IQ server will fail. In this case, you may need to call Sybase Technical Support for assistance.
- To shut down the SCC Server, enter "shutdown" at the scc-console prompt: `% scc-console> shutdown`
- It is important to note the following:
 - The SCC Server must be running in order for you to administer IQ.
 - For ongoing production, it is probably best to start SCC from a script that restarts automatically.
 - On UNIX, the SCC server can run in the foreground or background depending on your environment.
 - However, the SCC Server must be in the foreground in order to execute the "scc-console" commands such as "status" and "shutdown".

3. INSTALLING THE IQ SERVER'S SCC AGENT

- Your IQ Server's local SCC agent is installed as part of your IQ Server install.
- In 15.3 and 15.4, the agent is located in the directory: \$SYBASE/SCC-3_2/bin
- For more info on installing, see the [Sybase IQ15.4 Installation and Configuration Guide](#).

4. STARTING THE IQ SERVER'S LOCAL SCC AGENT

- An SCC agent must be running on each IQ server to be administered. The SCC agent on the SCC server communicates with the SCC agent on your local IQ server.
- You have several options for starting the SCC agent including specification of the port the agent will listen on.
- To start the IQ Server's SCC agent on the default port on UNIX, do the following:
`$SYBASE/SCC-3_2/bin/scc.sh &`
- To start the IQ Server's SCC agent on a specific port (non-default) port do the following:
`$SYBASE/SCC-3_2/bin/scc.sh --port rmi=9991; $SYBASE/SCC-3_2/bin/scc.sh -start &`
- To shut down the local SCC agent, enter "shutdown" at the scc-console prompt, enter the following:
`scc-console> shutdown`
- NOTES:
 - Leave the SCC agent running whenever Sybase IQ is running. One way to do this is to run the SCC agent as a service that restarts automatically.
 - On UNIX, the SCC agent can be running in the foreground or background.
 - The SCC agent must be running in the foreground to execute the console commands.

5. CONNECTING TO THE SCC SERVER

- Launch your Internet browser.
- Connect to the Sybase Control Center server using the IP address (RMI port) defined in your SCC Server installation. That is, in your Web browser, enter your SCC Server's host name and IP address. For example, enter the following: **https://<scc-hostname>:9991/scc** ---> Where **"9991"** is the SCC Server's listener port (RMI port) that was specified in the SCC Server install.
- This returns the SCC Server login window.
- In the SCC Server login window, log into the SCC server using the account and password defined in your SCC install.
- For example, if you are using the default account and password, do the following:
 - Enter your User-ID: **sccadmin**
 - Enter your password: **<no password>** -> This is the default

SUMMARY OF PROCEDURES FOR REGISTERING AND AUTHENTICATING AN IQ SERVER TO SCC

Below, is a brief summary of the steps required to register and authenticate an IQ server to SCC.

- Once you have successfully installed the SCC components and logged into your SCC server, you must register and authenticate the servers you want to administer with your SCC server. This is accomplished using the SCC GUI tool and includes the following steps:
 1. Registering your IQ Server to the SCC Server ("Registering a Resource").
 2. Authenticating your IQ Server to SCC.
 3. Registering your IQ Server's SCC Agent.
 4. Authenticating your IQ Server's SCC Agent.
- The procedures for registering and authentication your IQ server to SCC are too detailed for this document. Therefore, the following provides an overview of the steps required for this procedure. For complete details on performing this task, please see the [Sybase3.2 Control Center for Sybase IQ Guide](#).

1. REGISTERING YOUR IQ SERVER TO SCC

- From SCC's Resource Explorer Window, choose the "Register" option, and do the following:
 1. Enter your IQ Server name: **my_scc_server_box**
 2. Enter your IQ Server Type ("Resource Type"): This will be either **"IQ Multiplex"** or **"IQ Server"**.
- From SCC's Resource Explorer Window, enter the following information in the "Connection Information Window":
 1. Enter the UNIX host name for your IQ Server: **my_iq_server_box**
 2. Enter the UNIX port number your IQ Server: **15400**

2. AUTHENTICATING YOUR IQ SERVER TO SCC

- From SCC's Resource Explorer Window, choose the "Authenticate" option, and do the following:
 1. Enter your IQ Server Username: **DBA** -> the default user name
 2. Enter your IQ Server Password: **sql** -> the default password

3. REGISTERING YOUR IQ SERVER'S SCC AGENT

- From the Administrative Console's "Register Agent" Window (within the Resource Perspective Window), choose the "Register" option, and do the following:
 1. Enter your SCC Agent's Host Name: **my_iq_server**
 2. Enter your SCC Agent's Port: **9991** -> This is the default. If you changed it, then this will be different.

4. AUTHENTICATING YOUR IQ SERVER'S SCC AGENT

- From the Administrative Console's "Register Agent", do the following:
 1. Enter your IQ Server's SCC Agent User Name: **"uafadmin"** -> The default
 2. Enter your IQ Server's SCC Agent Password: **->the default is blank**
 3. Enter your IQ Server's SCC Agent Port: **9991** -> This is the default. If you changed it, then this will be different.

DBISQL AND OPEN CLIENT UTILITIES

THE OPEN CLIENT SDK

The Open Client SDK is now included in the IQ 15.3 install.

OPEN CLIENT UTILITIES

In IQ 15, several Open Client utilities were deprecated and then reinstated. These are as follows:

DEPRECATED UTILITIES

- In IQ 15.0 through 15.2, the following IQ-specific Open Client utilities were deprecated:
 - IQISQL
 - IQDSEEDIT
 - IQDSCP (UNIX only)
 - IQOCSFG (Windows only)

REINSTATED UTILITIES

- In IQ 15.3, these Open Client utilities were reinstated:
 - IQISQL.
 - IQDSEEDIT
 - IQDSCP (UNIX only)
 - IQOCSFG (Windows only)

IQISQL VERSUS ISQL

- ISQL is a command-line Interactive SQL utility that uses the Adaptive Server Enterprise Open Client API.
- IQISQL is IQ's version of isql, and it is basically the same utility with a different name.
- Functionally, iqisql remains deprecated in IQ 15.3. However, it has been retained for purposes of backwards compatibility. For syntax and parameters, see the [ASE 15.5 Utility Guide](#).
- Neither isql nor iqisql 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 VERSUS 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 15.
- 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 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, Sybase 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 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 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=tcip{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 .isqlPreferences11.
 - 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 15, otherwise you could get errors like:

```
Syntax error near 'PROCEDURE' on line 6
SQLCODE=-131, ODBC 3 State="42000"
```

INTERACTIVE SQL RESTRICTIONS

- In IQ 15.3, 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, 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

NEW CLIENT ACCESS API'S

- In the past, client access to Sybase IQ was limited to JConnect (Sybase's implementation of JDBC), ODBC, and Open Client.
- In IQ 15, there are now a number of new data access APIs. These include the following:
 - ADO.NET
 - Perl (provided by the SQL Anywhere Perl DBD::SQLAnywhere DBI module)
 - Python (provided by the *sqlanydb* interface)
 - PHP (provided by the SQL Anywhere PHP module)
 - OLE DB
 - Native Ruby Driver
 - Ruby/DBI Driver. Note: Support for the Ruby driver is provided only for Windows 32-bit and Linux32(x86) platforms.
- These new APIs make it easier to build and deploy database applications in multiple programming environments.

CONNECTIVITY TIPS

- It is important to note that some older versions (pre-OC 15) of Open Client do not support bigint, unsigned int and unsigned bigint.
- Source code, sample projects, and OS-specific binaries for Perl, Python, and PHP are installed in the "**%IQDIR15%\SDK**" directory on Windows and the "**\$IQDIR15/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 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 [SQL Anywhere Data Source Utility \(dbdsn\)](#) .

ODBC DRIVER MANAGER ON UNIX

- Sybase IQ now provides SA’s *libdbodxx* shared object. In IQ 15.3, it is called *libdbod11* and in IQ 15.4 it is labeled *libdbo12*. This can be used on all supported UNIX platforms as an ODBC driver manager.
- For more details on 15.4 see:
 - For ODBC samples: *"SQL Anywhere 12.0.1 > SQL Anywhere Server - Programming > ODBC support > Building ODBC applications.*
 - For creating an ODBC.INI file: [SQL Anywhere 12.0.0 > SQL Anywhere Server - Database Administration > Starting and connecting to your database > SQL Anywhere database connections > Creating ODBC data sources.](#)

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.

NAS SUPPORT

- Sybase IQ does not currently support NAS (Network Attached Storage).
- This support is under review, and will be implemented on a vendor by vendor basis.

CLIENT SIDE LOADING

- In a client side load, the client application opens the file and then sends the data packets across the network. Each packet contains a portion of the file. The packets are consumed by the server in memory without recreating the file on the server side.
- Client side loading has undergone significant enhancements in 15. The following lists these changes.

THE IQ_BCP UTILITY HAS BEEN REPLACED

- It is important to note that the iq_bcp utility has been deprecated and replaced by the “LOAD TABLE USING FILE” statement.

NEW FEATURES

- The “LOAD TABLE USING CLIENT FILE” command now supports all load options, including LOB support. It is a true bulk loader and its performance approximates a server side LOAD TABLE (plus the time required to transfer the data across the network).
- In IQ 15, the LOAD TABLE command also handles client side loads. Therefore, when using the LOAD TABLE statement, the input data source file can exist on either the client or the server side. When the load file is on the client, issue the LOAD TABLE command using the “USING CLIENT FILE” option.

CLIENT DRIVER AND PROTOCOL SUPPORT

- Client side loading is provided to clients that use the following IQ/SQL Anywhere database drivers:
 - ODBC
 - JDBC using the iAnywhere driver
 - OLEDB
 - ADO.NET
 - Perl
 - PHP
 - Python
- Client side loading is NOT supported by the TDS protocol used in Open Client and jConnect for JDBC.

SECURITY

- To improve the security of client side loading, use the Transport Layer Security (TLS) protocol to encrypt data across the network.
- Additionally, the database administrator can control or disable client side loading with various layered security mechanisms that are available.

PERFORMANCE ADVANTAGES FOR CLIENT SIDE LOADING

- With a server side load, IQ must wait for the file to be fully written to the remote server before it can begin loading the data.
- With a client side load, data loading happens immediately.
- The time required for a client side load should be approximately the same as a server side load plus the network latency.
- Increasing the network packet size can be used to improve network transmissions.

PERFORMANCE ENHANCEMENTS FOR CLIENT SIDE LOADING

- Load performance on a single machine has improved by the parallelization of High_Group (HG) and Word (WD) index creation.
- In a Multiplex environment, loads can be executed in parallel against different tables by using multiple writer nodes. Read more about this in the following section: [MULTIPLEX CHANGES](#).
- Client side loading avoids cluttering the OS file systems that IQ server runs on. It also avoids file access issues on the host server that can occur when the average user lacks the privileges needed to write a file on the host's file system.

PERFORMANCE TIPS

PARALLELISM

NEW DATABASE OPTIONS FOR TUNING PARALLELISM

- To help manage the additional memory requirements of parallelism in IQ 15, IQ now provides several new 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 15 introduces several new 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.

TIP: KEEP YOUR OLD QUERY RUN TIMES AND QUERY PLANS

- If you are migrating from pre-15 versions of IQ, keep your old query performance timings and plans available as a baseline for comparison to your new IQ 15 query performance timings and plans.

NEW OPTIMIZER CONFIGURATION PARAMETERS FOR IQ 15

- IQ 15.x offers many optimizer configuration parameters which can help improve query performance. Please refer to the IQ manuals (Reference: Statements and Options), for details on these various options.
- Make sure you understand IQ's configuration parameters before making any major change because server-wide configurations can cause undesirable impact on all queries. Instead, consider setting options temporarily for individual queries.
- Also, there are a number of optimizer hints available in the query plan. Be sure to make note of them, and implement the advised changes if query performance is unsatisfactory.

ADDITIONAL PERFORMANCE RECOMMENDATIONS

- Additional performance recommendations can be found within this document in the following sections:
 - "Enhancements For Client Side Loading".
 - "IQ Database Options That Help Optimize Performance".
 - "Data Modeling Recommendations".

DATA MODELING RECOMMENDATIONS

This section provides guidance on tuning your database schema for 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 AND MINIMIZE STORAGE

- 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.

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 and TEXT 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 (64K ASCII hex for Varbinary).

- The maximum width of Long Varchar is 64K (128K ASCII hex for Long Varbinary).
- 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

- Large objects can be stored in binary or text based objects.
- This option extends long binary data type from a maximum size of 64K 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

- Because storage in IQ is fixed width, use CHAR wherever possible.
- VARCHAR types add a slight storage overhead. For example, a VARCHAR(100) will require 101 bytes of storage: 100 bytes for the 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.

SECURITY RECOMMENDATIONS

From its inception, Sybase IQ has offered a secure information environment for its users: authentication, authorization, data encryption, and auditing. IQ 15 includes new capabilities for securing the data in your database.

AUTHORITIES

Earlier versions of Sybase IQ supported only two authorities for performing database administrator tasks: DBA and RESOURCE. RESOURCE authority gives a user permission to create and modify database objects. DBA authority enables the user to carry out any activity in the database: create tables, change table structures, create new user IDs, revoke permissions from users, and so on.

DBA authority gives a user the “keys to the kingdom”. A user with the authority to do a backup will also have the power to do all administrative tasks in the database. All power - and potential for failure – is bundled into a single authority.

Sybase IQ rectifies this vulnerability by providing a granular set of authorities, so a user has permission to do only the tasks that he needs to do.

The following table lists the authorities you can assign to users:

Authority	Tasks that can be performed
BACKUP	Backup databases and transaction logs.
DBA	All administrative tasks.
MULTIPLY ADMIN	Multiplex administration tasks – create and drop servers, and change configuration settings.
OPERATOR	Checkpoint databases, drop connections, back up databases and monitor the system.
PERMS_ADMIN	Manage data permissions (except DBA or REMOTE DBA), groups, authorities, and passwords.
PROFILE	Profiling, tracing and diagnostic operations.
READCLIENTFILE	Load data into server from client machine.
READFILE	Use OPENSTRING clause in a SELECT statement to read a file on the server.
REMOTE DBA	Limited set of DBA permissions for a synchronized mobile database user.
RESOURCE	Create and modify database objects.
SPACE_ADMIN	Manage DBspaces.
USERADMIN	Manage users, external logins and login policies.
VALIDATE	Database, table, index and checksum validation.
WRITECLIENTFILE	Unload data from server onto client machine.

USER LOGIN POLICIES

- In Sybase IQ 15, you can create a login policy that defines the rules to be followed when establishing a user's database connection. The following settings are governed by a login policy:
 - Password life time.
 - Password grace time.
 - Password expiry on next login.
 - Locked.
 - Maximum connections.
 - Maximum failed login attempts.
 - Maximum days since login.
 - Maximum non-DBA connections.
 - When you create a user in the database, you assign a login policy to that user. You can modify login policies, and reassign users to different login policies.

IPV6

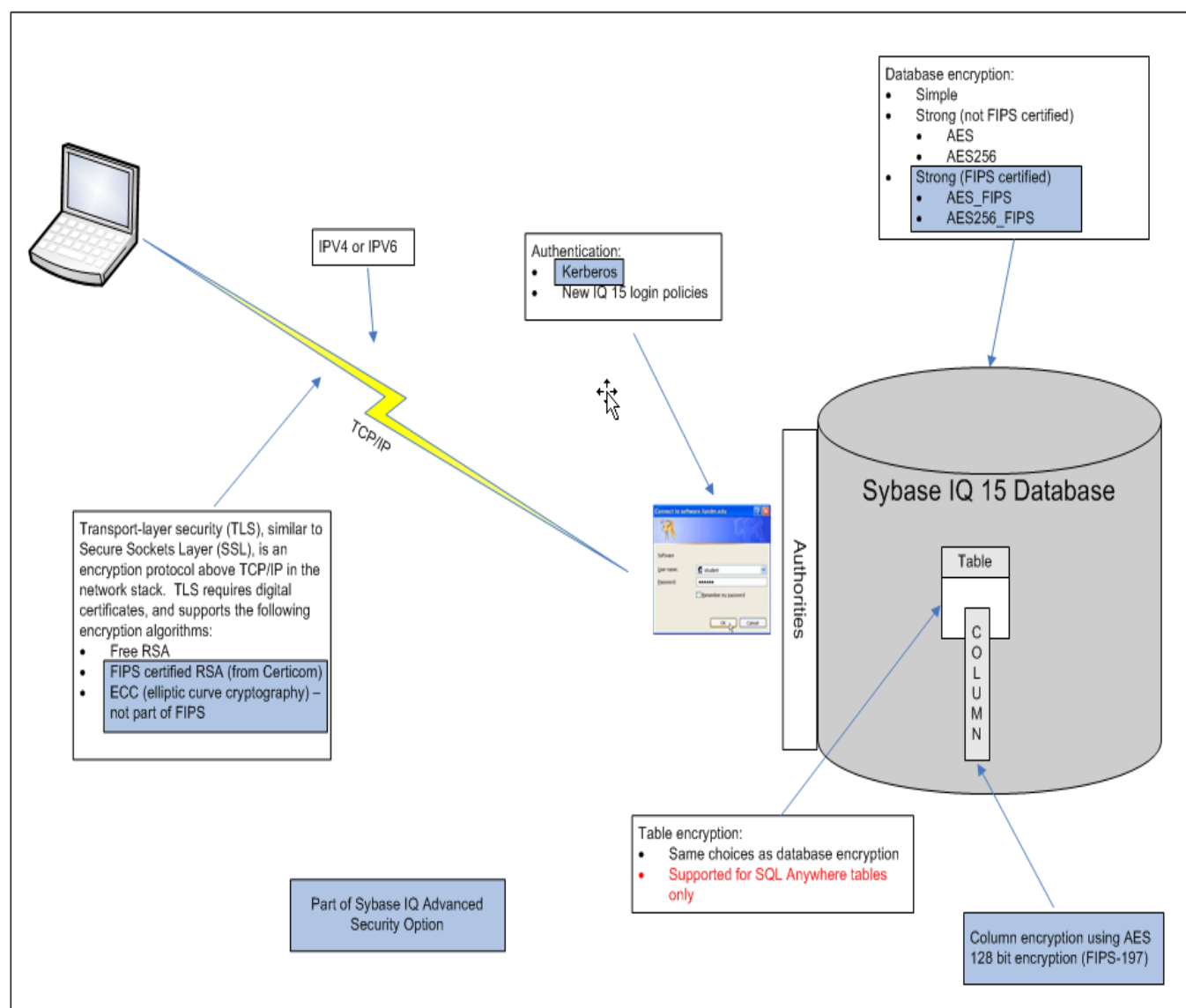
Sybase IQ now supports Internet Protocol version 6 (IPv6), which contains addressing and control information to route packets over the Internet. IPv6 supports 2^{128} unique IP addresses, which is a substantial increase over the number of addresses supported by its predecessor IPv4.

SYBASE CENTRAL AUTHENTICATION

- The Sybase IQ agent enables a Sybase Central client to perform remote administrative functions for the Sybase IQ server. Sybase Central communicates with the IQ agent to:
 - Start/stop servers
 - Access log files
 - Perform system functions
 - An authentication layer has been implemented between Sybase Central and the IQ agent to allow only authenticated clients to access the agent's Remote Method Invocation (RMI) methods to administer the database.

ADVANCED SECURITY OPTION

If you need more secure authentication capabilities, or stronger encryption algorithms, then IQ offers the Advanced Security Option. This licensed option for Sybase IQ includes support for FIPS encryption, database column encryption, and Kerberos authentication. The following picture shows the security features that are included in the base product, and the features that are included with the Advanced Security Option:



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. PowerDesigner can be used to generate and back up the database schema for an IQ database.
- 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 15 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 15 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 Sybase Technical 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.

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%.

MESSAGE LOG MANAGEMENT USING -IQMSGSZ AND -IQMSGNUM

- There are two new message log management parameters called -iqmsgsz and -iqmsgnum.
- 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 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
 - TRANSLLOG
 - TRANSLLOGMIRROR
 - 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 FOR ENABELING 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

- In IQ 15, 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 IQTMP15 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. IQTMP15 (or the SATMP environment variable for SQL Anywhere's temporary files)
 2. TMP
 3. TMPDIR
 4. TEMP

DATABASE CONNECTION ENHANCEMENTS

- Temporary connections are now named in 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 in IQ 15.4.
- In 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

- Sybase 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 anytime 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 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 “APPEND_LOAD” OPTION

- There are several tradeoffs to consider when turning the APPEND_LOAD option ON vs. OFF. With the APPEND_LOAD option ON, data loads will run faster. With the APPEND_LOAD option OFF, there will be less fragmentation.
- Advantages of turning the “Append_Load” option ON:
 - With the “APPEND_LOAD” option on, data will run faster because it uses fewer resources during the load, such as:
 - Fewer pages are read in from IQ main store.
 - Fewer pages are written out to the IQ main store. And, fewer pages modified means less versioning.

- Also note, the following occurs when Append_Load is ON:
 - New rows will go to the end of the table on new pages.
 - New pages from the free list will be filled with new data and the holes left by deletes will be compressed out when stored on disk. When these pages are read into the IQ main cache, they will be uncompressed and will require more space than full pages for the same number of rows.
- Advantages of turning the “Append_Load” Option OFF:
 - IQ will fill in the holes left when rows are deleted.
 - Versioning is higher because more existing pages will be modified.
 - Since having too many pages that are partially full could be viewed as a form of fragmentation, using APPEND_LOAD OFF will reduce this “fragmentation” by helping to reuse the holes left by the deletes.

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 “MINIMIZE_STORAGE” OPTION

- Consider using the “MINIMIZE_STORAGE” option for most of IQ data storage. When doing this, FP(1), FP(2) and FP(3) indexes that use lookup tables will be created instead of flat FP indexes. These indexes improve query performance by using less space and decreasing I/O usage.

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 SAMPLES AND SCENARIOS

BACKUP METHODS

- An IQ database can be backed up using one of the following methods:
 - An IQ Database Backup.
 - An IQ Virtual Backup with an Archive Backup (for log files).
 - An OS-Level Backup (Note: A virtual backup is preferred over this method).

IQ SERVER-LEVEL BACKUPS

- 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.

VIRTUAL BACKUP ADVANTAGES

- With a 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 TIPS

- 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".
- In IQ 15, 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 15, 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.

BACKUP RESTRICTIONS

- Backup and restore commands can only be executed on the coordinator in a Multiplex environment.
- In IQ 15.3 and above, note the following backup restrictions:
 - A selective backup is not allowed.
 - An incremental backup may not be a virtual backup.
 - A virtual backup is an all-inclusive full backup.

IQ RESTORE

- Disaster Recovery (DR):
 - In 15.2 ESD1 and above, restoring to the DR site is performed by starting the utility_db with the DR server name.
 - In 15, the DR server is no longer required to have the same name as the production server.
 - 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 IQ 15 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 15 includes several new 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 calling Sybase Technical 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 or Multiplex?
 - Is this Simplex or Multiplex?
 - If Multiplex, which node: coordinator, reader or writer?
 - 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
 - .srvlog
 - .stderr
 - -zo output file after setting “-zr all” or “-zr sql”
- Provide the Operating System (OS) information : version, patch level, 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, Sybase Central Plug-in, 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 iqisql, provide the output from the following:
 - sp_iqstatus
 - select * from sysfile
 - select * from sysiqdbfile
 - select * from sysoptions order by 2,1
 - Monitoring output:

```
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 2-3 “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 UNIX, 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+) = 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 slow down 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, generated or an ad-hoc query?
- What is the SQL for the query?

- 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 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 Query_Plan_As_HTML_Directory = '<html_plans_directory>' ;
```

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 repro?

DIAGNOSTICS FOR 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: stktcr*.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'`

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 2-3 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 RESTORE 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.

DIAGNOSTICS FOR 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](#).
- Provide the contents of <IQ dbname>.Imp 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. The values returned by a UDF can be either a single value (a Scalar Function), or rows of data (a Table Function).
- UDFs can run either within the IQ server or external to the IQ server. As of IQ 15.1, IQ provides three types of UDFs:
 1. External C/C++ UDFs that run within the IQ server and provide the highest level of performance.
 2. External Language UDFs that run outside of IQ and are written in languages such as Java, PERL, or client-side embedded C/C++ .
 3. Interactive SQL UDFs that run within IQ.

HIGH-PERFORMANCE, EXTERNAL C/C++ UDFS (INTERNALLY EXECUTED UDFS)

PARTNER UDFS (THE IQ_UDF OPTION)

- In IQ 15.1, Sybase introduced support for high-performance, in-process, external C/C++ user-defined functions. Although referred to as “external” UDFs, these UDFs execute within the IQ server process. This style of UDF supports functions written in C or C++ code and can be compiled and linked into a dynamically linkable library. Once defined and linked, these UDFs can be used directly within queries or other SQL statements.
- Because these UDFs run within the IQ server process, they offer greater performance. However, if improperly implemented, they can also corrupt the IQ server process itself by incorrectly writing into internal memory spaces. Therefore, to reduce this risk of server failure, the initial release of internally executed UDFs was restricted to Sybase Certified Partners and required the purchase of the IQ_UDF license.

IN-DATABASE ANALYTICS (THE IQ_IDA OPTION)

- In IQ 15.4, Sybase introduced a new license called the InDatabase Analytics (IQ_IDA) option. This option is available to all Sybase IQ customers and partners without certification. It enables any customer to build, deploy and run their own external V4 C/C++ User-Defined Functions (UDF) within IQ.
- For non-certified partners, this option is available for an additional cost.
- In order to preserve the integrity of IQ’s memory space, the execution of these UDFs is restricted to read-only nodes of a multiplex configuration. This reduces the risk of a UDF writing into IQ’s memory space and corrupting data, or bringing down the server.
- In 15.4, the execution of UDFs on a simplex server or on a multiplex writer node remains restricted to Sybase Certified Partners and requires the purchase of an IQ_UDF license.
- It is important to note that external C/C++ UDFs are different than the Interactive SQL UDFs introduced in earlier versions of Sybase IQ. These UDFs remain unchanged and are briefly described below.

INTERACTIVE SQL UDFS (INTERNALLY EXECUTED)

- Interactive-SQL UDFs are created using standard SQL statements (T-SQL or Watcom-SQL), and then executed within the IQ server. These UDFs are available to all IQ users without an additional license.
- These UDFs are limited to the functionality provided by the SQL languages supported by IQ and may lack the performance of external C/C++ UDFs. However, this style of UDF is readily available, and can provide useful functionality.
- Interactive SQL UDFs have been available prior to IQ 15 and do not require an additional license. For more details on UDFs, please see the following:
 - [Sybase IQ 15: User Defined Functions Guide.](#)
 - [Sybase IQ 15.3: User-Defined Functions Guide: Creating and Executing User-Defined Functions.](#)

EXTERNALLY EXECUTED UDFS (JAVA, PERL, ETC)

- Sybase IQ also provides an external runtime environment option that allows users to write UDFs in different programming languages. These UDFs execute outside the IQ server process, in an external environment such as a JVM for Java.
- Sybase IQ provides support for the following runtime UDF environments:
 - CLR (Microsoft .NET Common Language Runtime).
 - ESQL and ODBC (C/C++ Embedded SQL or ODBC server-side requests).
 - Java.
 - Perl.
 - PHP
- Each external environment has its own set of APIs for processing arguments and returning values back to the server. For example, the Java external environment uses the JDBC API.
- In the Java external environment, you can also create a Java Table UDF. This is a Java function that returns a table (rows and columns) of output values instead of a single value.
- External UDFs do not require a license, and do not need to adhere to the strict API of internally executed C/C++ UDFs.

SCALAR AND AGGREGATE FEATURES FOR UDF V3 VS V4

API VERSION 3 (V3)

In IQ 15.1, 15.2 and 15.3, partners utilized version 3 (v3) of the UDF API. Version 3 provides two types of UDFs:

1. Scalar: Accepts a single value (scalar) input argument and returns a single value output argument.
2. Aggregate: Accepts an input argument with a set of values (aggregate), and returns a single value output argument.

API VERSION 4 (V4)

In IQ 15.4, Sybase introduced version 4 (v4) of the UDF API. This API supports new UDF signatures and also enables distributed processing of UDFs within Sybase IQ. The types of UDF processes that v4 supports include the following:

1. Scalar: Same as V3.
2. Aggregate: Same as V3.
3. Table: Accepts a single value (scalar) input argument, and produces a table (rows and columns) of output values.
4. Table parameterized function (TPF): Accepts an input argument that is either scalar or a table, and produces a table of output values.

FAILOVER CONSIDERATIONS FOR THE IQ_IDA OPTION

TIP: CONSIDER YOUR FAILOVER LOGIC

- Unlike the IQ_UDF option, UDFs 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 PROGRAMMING PRACTICES

UDF DO'S FOR EXTERNAL C/C++ UDFS

- If a UDF needs to be modified at runtime, Sybase recommends following a set of procedures for safely unloading the old UDF library and reloading the new library. [See the IQ 15.4 User-Defined Functions Guide](#) for more information.
- To facilitate the construction of UDFs, Sybase IQ includes a C-based API. This API comprises a set of predefined entry points for UDFs, a well-defined context data structure, and a collection of SQL callback functions that provide a communication path from the UDF back to the server.
- 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 15.4, it is recommended that you upgrade to the v4 API. This version allows you to distribute queries incorporating UDFs across a 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.
- 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.
- 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.
- When using UDFs in your Multiplex environment, be sure that the configured failover logic never results in a reader node being promoted to either a writer or coordinator. That is because only reader nodes can run UDFs and such a promotion will cause all subsequent UDF executions to fail.
- Read the [Sybase IQ 15.4 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 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. Sybase 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.
- Certified Sybase 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, Sybase strongly recommends that UDFs are installed and executed on read-only multiplex nodes only.
- Read the [Sybase IQ 15.4 User-Defined Functions guide](#) for practices to avoid, such as performing complex, or memory-intensive operations that are repeated every invocation.

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.

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 involving v3 and v4 UDFs are evaluated. 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. Enable memory tracking to help you locate memory leaks in your UDFs. 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 v3 and v4 APIs also provide callback functions enabling the UDF to write to the .iqmsg message file, so you can log your own information as appropriate.
- Also, Microsoft Visual Studio 2008 developers can use the Microsoft Visual Studio Debugger to step through the user-defined function code.

UDF INTERNATIONALIZATION

- To support multiple languages, UDFs should be internationalized. If a UDF accepts string keywords as input parameters, you should place these keywords in external files. The same holds true for any exception text and log messages used by the UDF. This allows localization of UDFs without making code changes.

UDF TIME SERIES

- Time series is a sequence of time points, measured at successive times and normally spaced at uniform intervals. Time series forecasting uses a model to forecast future data points based on past data points. Sybase IQ includes SQL functions for time series forecasting and analysis. These functions in turn declare user-defined functions (UDFs) that execute within the IMSL™ C Stat and C Math external libraries.
- The time series forecasting and analysis functions include both OLAP-style aggregates and supporting scalar functions. These functions are available only with RAP.

DATABASE OPTIONS

PURPOSE OF DATABASE OPTIONS

- Database options provide a method for controlling many aspects of IQ's behavior.
- Unlike 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 Sybase Technical Support.

WHERE TO FIND DOCUMENTATION OF DATABASE OPTIONS

- In most cases, documentation for IQ's database options can be found in IQ's Alphabetical List of Options found in the IQ Reference and Statements Guide (see below). There are also several options that apply to SQL Anywhere, and these can be found in the Alphabetical List Of Options for SQL Anywhere (see below). Note: Only a few SQL Anywhere options are applicable to IQ.
- IQ's Alphabetical List of Options : [See Sybase IQ 15.4 - Reference: Statements and Options: Database Options: Alphabetical list of options for more details.](#)
- SA's Alphabetical List of Option: [SQL Anywhere 12.0.1 – Database Administration-Configuring your database-Database options-Alphabetical list of options.](#)

RANGE OF BEHAVIOR

Database options control the following types of behavior:

COMPATIBILITY

- Controls the extent to which the Sybase IQ behaves similar to 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 AND 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 VERSUS 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 Authority.

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 SP_IQCHECKOPTIONS BEFORE AND AFTER AN UPGRADE TO VERIFY OPTIONS 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 CUSOR

- 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' ;
 - 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"**.

QUERY PLAN SETTING

- **SET OPTION public.Query_Plan='Off' ;**
 - This option prevents printing of the query plan to the .iqmsg log file.

MINIMIZE STORAGE SETTINGS

- **SET OPTION public.Minimize_Storage='On' ;**
 - This option prevents printing of the query plan to the .iqmsg log file.
 - This option restricts how much space data access methods can consume.
 - It is recommended that this option be set to 'On' before creating any tables in a database.
 - **NOTE:** Turn this option OFF when loading very wide tables (1000 columns or more).

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"**.
- **APPEND_LOAD = "On" ;**
 - This option controls how new rows are inserted into existing tables.
 - Table loads tend to be faster if APPEND_LOAD is set ON. If your application rarely deletes rows, it is recommended to set this option ON permanently.
 - The default setting is OFF. This causes all new rows to be inserted into any empty row id's in the table (as a result of earlier deletes) before appending.

LOAD_MEMORY_MB

- Beginning in IQ 15.2, all delimited and binary data load operations use IQ temporary cache for load memory. Fixed width loads continue to use load memory as specified by LOAD_MEMORY_MB. See ["A Practical Hardware Sizing Guide for Sybase IQ 15"](#) for sizing guidelines.

ROW_COUNT =

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

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' ;`

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 15

Below, is a listing of the database options that were either added or whose defaults were changed in IQ 15.4 and 15.3. Also, see the following appendix for a listing of database options that were deprecated in IQ 15.

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

1. For IQ-Level Options: [Sybase IQ 15.4 - Reference: Statements and Options: Database Options: Alphabetical list of Options](#).
2. For SA-Level Options: [SQL Anywhere 12.0.1 – Database Administration-Configuring your database-Database Options-Alphabetical list of Options](#).

LISTING OF DEPRECATED OPTIONS FOR IQ 15

For a summary of options that were removed from IQ 15.0 and above, see Appendix 1.

LISTING OF NEW OPTIONS FOR IQ 15.3 AND 15.4

Below, is a listing of the database options that were added in IQ 15.4 and 15.3.

blocking_others_timeout [SA-Level Option]

Description: Specifies the amount of time that another connection can block on the current connection's row and table locks before the current connection is rolled back. This option can be used to prevent a low priority task from blocking other connections for longer than the specified time.

Allowed values: Integer, in milliseconds

Default: 0

Remarks: Only supported by "Command Sequence" connections (not supported by TDS or HTTP connections or by events). This option can only be set as a temporary option that will remain in effect for the duration of the current connection. When this option is set to 0, other connections can block on the current connection for an indefinite period of time.

Default Proxy Table Row Count

Description: This is a new database option that enables users to override the default estimate for the number of rows to be returned from a proxy table. Correctly specifying this value can significantly improve performance. Too small or too large can degrade performance.

Default: 200000

Allowed: 0 to 4294967295

Version: This option was added in 15.4

Default Table UDF Row Count

Description: This option was introduced in 15.4 to provide the query processor with an estimate for the number of rows that a table UDF will return. For a Java UDF table, this is the only way to pass this information. For a C or C++ table UDF, consider publishing this information in the `describe` phase using the `EXTFNAPIV4_DESCRIBE_PARM_TABLE_NUM_ROWS` parameter. Correctly specifying this value can significantly improve performance. Too small or too large can degrade performance.

Default: 200000

Allowed: 0 to 4294967295

Version: This option was added in 15.4

dqp_enabled

Description: This option was introduced in 15.3 to support the new distributed query process (DQP) functionality. It provides a means to disable DQP, since by default, this option is on.

Default: On

Version: This option was added in 15.3.

Reserved Keywords

Description: This option allows users to designate individual keywords to be disabled by default. These can only be set for the PUBLIC group. User and temporary settings are not allowed.

Default: Empty String

Limitations: You cannot turn on the keywords SET, OPTION, and OPTIONS. The following determines whether a word is identified as a keyword (in order of precedence):

- It appears in the SQL Anywhere list of reserved words
- It is turned on with the `RESERVED_KEYWORDS` option
- It is turned off with the `NON_KEYWORDS` option

Version: This option was added in 15.4

sr_timestamp_with_time_zone_format [SA-Level Option]

Description: Sets the format for timestamp with time zone values that are retrieved from the database

Allowed values: A semicolon delimited set of key=value pairs (key1=value1;key2=value2;...)

Allowed keys: "relative-tolerance-percent", and "absolute-tolerance".

Default: yyyy/mm/dd hh:nn:ss.Sssss +hh:nn

Remarks: The Message Agent replicates timestamp with time zone information using this option.

st_geometry_interpolation [SA-Level Option]

Description: This option allows control of interpolation when spatial calculations involve circular arcs.

Allowed values: A semicolon delimited set of key=value pairs (key1=value1;key2=value2;...)

Allowed keys: "relative-tolerance-percent", and "absolute-tolerance".

Default:

- If neither key is specified, relative-tolerance-percent defaults to 0.3, which gives 40 points in a complete circle, and absolute-tolerance defaults to the tolerance specified by the SRID
- If only one key is specified, the other key is set to its minimum value.
- If both keys are specified, the minimum number of points is produced such that all restrictions are satisfied.

Remarks:

Regardless of the setting for this database option, there will be at least 3 points per curve segment. Furthermore, no two consecutive points on the interpolated ST_LineString are within tolerance of each other.

LISTING OF ENHANCED OPTIONS FOR IQ 15.4

Below, is a listing of existing database options whose functionality was enhanced in 15.4.

Identity insert

Description: This option enables users to insert values into, or to update, an IDENTITY or AUTOINCREMENT column.

Enhancement: In 15.4, this option was enhanced to accept table names containing the characters “#” and “@” when enclosed in single quotation marks.

Default: Not set

Allowed Values: = 'tablename'

max_failed_login_attempts

Description: This option specifies the maximum number of failed login attempts (since the last successful attempt) that can occur before a user account is locked.

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.

LISTING OF OPTIONS WITH NEW DEFAULTS FOR IQ 15.4

Below, is a listing of existing database options that now have different DEFAULT values in version 15.4 than in previous versions of IQ.

fp_prefetch_size

Description: Database-specific option.

Default: In 15.4, the default for this option has increased from 10 to 20.

Allowed Values: 0 – 100

For more details, [Sybase IQ 15.4 - Reference: Statements and Options: Database Options: Alphabetical list of options](#).

LISTING OF OPTIONS WITH NEW DEFAULTS FOR IQ 15.1, 15.2 AND 15.3

Below, is a listing of database options that have DIFFERENT DEFAULT values in IQ 15.1, 15.2 and 15.3 then in IQ 12.7.

- For more details, see: [Sybase IQ 15.4 - Reference: Statements and Options: Database Options: Alphabetical list of options](#).

fp_lookup_size

Description: Specifies the maximum number of lookup pages used in Sybase IQ. Controls the amount of cache allocated to the creation of Lookup FP indexes, particularly FP(3) Indexes.

Default: Starting in IQ 15.0, the default for this option changed from 32767 to 16.

login_procedure

Description: The LOGIN_PROCEDURE names a stored procedure to run when users connect. You can accept the default or specify a different stored procedure. Do not modify sp_login_environment.

Default: in IQ 15.0, the LOGIN_PROCEDURE was changed to call the stored procedure sp_login_environment, which then calls the procedure sp_tsqll_environment. Note: The LOGIN_PROCEDURE replaces the sp_iq_process_login procedure previously used in 12.7. Additionally, IQ 15 introduced a "login policy" which permits specifying a set of rules to be applied whenever a user makes a connection.

main_reserved_DBspace_mb

Description: Allows control of the amount of space Sybase IQ sets aside in the IQ main store for certain small, but critical data structures used during release savepoint, commit, and checkpoint operations. For a production database, set this value to between 200MB and 1GB. The larger your IQ page size and number of concurrent connections, the more reserved space you need.

Default: In IQ 15, the default changed from 4 to 200. However, this default is not used. Instead, Sybase IQ reserves a maximum of 50%, and a minimum of 1%, of the last read-write file in IQ_SYSTEM_MAIN.

max iq threads per connection

Description: This option allows you to constrain the number of threads (and thereby the amount of system resources) that the commands executed on a connection can use. For most applications, use the default.

Default: In IQ 15, the default increased from 72 to 144. For most applications, use the default.

max iq threads per team

Description: Controls the number of threads allocated to perform a single operation (such as a LIKE predicate on a column) executing within a connection. The total for all simultaneously executing teams for this option is limited by the related option, max_iq_threads_per_connection.

Default: In IQ 15, the default increased from 48 to 144.

max query parallelism

Description: Sets an upper bound for parallelism the query optimizer can choose for GROUP BY operations or arms of a UNION, regardless of how many CPUs are available. This option is effective only on GROUP BY operations when PARALLEL_GBH_UNITS is not set. Normally, you would not set this option. However, if you have more than 16 CPUs and you see excessive CPU time spent on system usage, try setting MAX_QUERY_PARALLELISM to a value less than 16. Experiment with this value to determine the right setting for your environment.

Default: In IQ 15, the default increased from 24 to 64.

post login procedure

Description: Specifies a login procedure whose result set contains messages that are displayed by the client application immediately after a user successfully logs in. Can be set for a user-id or the PUBLIC group. You can customize the post login actions by creating a new procedure and setting POST_LOGIN_PROCEDURE to call the new procedure. Do not edit dbo.sa_post_login_procedure.

Default: In 15.0, the default procedure changed from DBA.sp_iq_process_post_login to dbo.sa_post_login_procedure.

prefetch sort percent

Description: Designates a percentage of prefetch resources for use by a single sort object. Increasing this value can improve the single-user performance of inserts and deletes, but may have detrimental effects on multi-user operations. Do not set this option unless advised to do so by Sybase Technical Support.

Default: In 15, the default was changed from 50Mb's to 20MB's.

query plan as html

Description: Generates graphical query plans in HTML format for viewing in a Web browser.

Default: In 15.3, the default changed from off to on. In 12.7 through 15.2, the default was off.

query temp space limit

Description: Specifies the maximum estimated amount of temp space before a query is rejected.

Default: In 15, the default changed from 50 to 20.

string_rtruncation

Description: Determines whether an error is raised when a string is truncated.

Default: In 15.0, the default changed from off to on.

temp_reserved_DBspace_mb

Description: Controls the amount of space Sybase IQ reserves in the Temporary IQ Store for certain small, but critical, data structures.

Default: In 15.2, the default changed from 2Mb's to 200 Mb's. However, this default is not used. Instead, Sybase IQ reserves a maximum of 50%, and a minimum of 1%, of the last read-write file in IQ_SYSTEM_TEMP.

temp_space_limit_check

Description: Checks for catalog store temporary space on a per connection basis.

Default: In 15.2, the default changed from Off to On.

LISTING OF NEW OPTIONS FOR IQ 15.0, 15.1, AND 15.2

Below, is a listing of the database options that were introduced in IQ 15.0, 15.1, and 15.2.

For more details on the following options, please see the documentation notation in italics for each option:

Notation (1):

[See Sybase IQ 15.4 - Reference: Statements and Options: Database Options: Alphabetical list of options for more details.](#)

Notation (2):

[See SQL Anywhere 12.0.1 - Database Administration-Configuring your database-Database options-Alphabetical list of options.](#)

Notation (3):

[See Sybase IQ 15.4 - User-Defined Functions: Building UDFs: Testing User-Defined Functions.](#)

Notation (4):

[See Sybase IQ 15.2-Time Series Guide: Overview: IMSL Libraries for Time Series Forecasting and Analysis Functions.](#)

allow_read_client_file (/)

Description: Enables client-side data transfer. This option must be enabled to read from files on a client computer. For example using the READ_CLIENT_FILE function.

Default: Off

allow_snapshot_isolation [SA-Level Option] (2)

Description: Controls whether snapshot isolation is enabled or disabled.

Default: Off

allow_write_client_file [SA-Level Option] (2)

Description: This option must be enabled to write files to a client computer, for example using the WRITE_CLIENT_FILE function. .

Default: Off

ansi_substring (1)

Description: When ON, a negative or zero start offset is treated as if the string was padded on the left with noncharacters. When OFF, the behavior of the **SUBSTRING** function is the same as in earlier versions of IQ. Where possible, use the **LEFT** or **RIGHT** functions instead.

Default: On

btree_page_split_pad_percent (1)

Description: Determines per-page fill factor during page splits for B-Tree structures. .

Default: 50

conn_auditing [SA-Level Option] (2)

Description: Controls whether auditing is enabled or disabled for each connection when auditing option is On.

Default: On

default_DBspace (1)

Description: Changes the default DBspace for where tables or join indexes are created. Can be set for a group, or user and takes place immediately.

default: (the empty string)

default_disk_striping (1)

Description: Sets default disk striping value for all DBspaces. Used only by CREATE DBSPACE if CREATE DBSPACE does not specify striping.

Default: ON for all DBspaces in the IQ main store.

default_having_selectivity_ppm (1)

Description: Provides default selectivity estimates to the optimizer for most **HAVING**.

Default: 0

default kb per stripe (/)

Description: Sets an upper limit on the amount of data written to a stripe before moving to the next stripe. The default value of 1KB means that each operation writes to a different stripe. To write multiple pages to the same stripe before moving to the next stripe, change the `DEFAULT_KB_PER_STRIPE` setting.

This sets the default size for all DBspaces in the IQ main store.

Default: 1

default like match selectivity ppm (/)

Description: Provides default selectivity estimates (in parts per million) to the optimizer for most LIKE predicates. If the column has either an LF index or a 1- or 2- or 3-byte FP index, the optimizer can get exact information and does not need to use this value.

Default: 150000

default like range selectivity ppm (/)

Description: Provides default selectivity estimates (in parts per million) to the optimizer for leading constant LIKE predicates. If the column has either an LF index or a 1- or 2- or 3-byte FP index, the optimizer can get exact information and does not need to use this value.

Default: 150000

default proxy table Row Count (/)

Description: This is a new database option that enables users to override the default estimate of the number of rows to be returned from a proxy table. Correctly specifying this value can significantly improve performance. Too small or too large can degrade performance.

Default: 200000

Allowed: 0 to 4294967295

Version: This option was added in 15.4

default table UDF row count (/)

Description: This option was introduced in 15.4 to provide the query processor with an estimate of the number of rows that a table UDF will return. For a Java UDF table, this is the only way to pass this information. For a C or C++ table UDF, consider publishing this information in the `describe` phase using the `EXTFNAPIV4_DESCRIBE_PARM_TABLE_NUM_ROWS` parameter. Correctly specifying this value can significantly improve performance. Too small or too large can degrade performance.

Default: 200000

Allowed: 0 to 4294967295

Version: This option was added in 15.4

dqp_enabled (/)

Description: This option was introduced in 15.3 to support the new distributed query process (DQP) functionality. By default, this option is on. Setting it to off disables DQP.

Default: On

enable_lob_variables (1)

Description: Controls the data type conversion of large object variables. Users must be licensed for the Unstructured Data Analytics Option to use large object variables.

Default: OFF

external_udf_execution_mode (3)

Description: Controls the amount of error checking and call tracing that is performed when statements involving external V3 user-defined functions are evaluated. When set to 0, external UDFs are evaluated in a manner that will optimize the performance of statements using UDFs. When set to 1, external UDFs are evaluated to validate the information passed back and forth to each UDF function. When set to 2, external UDFs are evaluated to not only validate the information passed back and forth to the UDF, but also to log, in the iqmsg file, every call to the functions provided by the UDFs and every callback from those functions back into the server.

Default: 0

fp_lookup_size_ppm (1)

Description: Controls the amount of cache allocated to the creation of Lookup FP indexes, particularly FP(3) Indexes. DBA permissions are required to set this option. Can be set temporary for an individual connection or for the *PUBLIC* group. Takes effect immediately.

Default: 2500

http_session_timeout (1)

Description: Specifies the amount of time, in minutes, that the client waits for an HTTP session to time out before giving up.

Default: 30

java_location (1)

Description: Specifies the path of the Java VM for the database.

Default: Empty string.

java_main_userid (2)

Description: Specifies the database user whose connection can be used for installing classes and other Java-related admin tasks.

Default: DBA

java_vm_options (1)

Description: Lets you to specify options that the database server uses when launching the Java VM specified by the *JAVA_LOCATION* option.

Default: Empty string

materialized view optimization [SA-Level Option] (2)

Description: Lets you specify the circumstances under which the optimizer can use stale materialized views. (Data becomes stale when any of the base tables referenced by the materialized view is updated.

- **Disabled:** Do not use materialized views for query optimization.
- **Fresh:** Use a materialized view only if it is fresh.
- **Stale:** Use materialized views even if they are stale.

Default: Stale

max client statements cached (1)

Description: Controls the number of statements cached by the client.

Default: 10

max prefix per contains phrase (1)

Description: Specifies the number of prefix terms allowed in a text search expression.

Default: 1

max priority (1)

Description: Controls the maximum priority level for connections.: High, Above Normal, Normal, Below normal, Low, Background

Default: normal

max query tasks [SA-Level Option] (2)

Description: Specifies the maximum number of server tasks that the database server can use to process a query in parallel:

- 0 allows the database server to use as many tasks as it chooses.
- 1 disables intra-query parallelism
- Any other value sets the maximum number of tasks allowed per query
- The number of tasks the database server can use for all requests is also limited by the number of users set using the **-gn** option and by the number of logical processors available.

Default: 1

max temp space [SA-Level Option] (2)

Description: Controls the maximum amount of Catalog Server temporary file space a connection can use before the request fails with an error message. The `temp_space_limit_check` option must be also be set to On (the default). If the connection exceeds the value, IQ rolls back the current statement and returns an error message: `Max_Temp_Space_Per_Connection exceeded.`"

Default: 0

Allowed Values:

- A value of 0 indicates that there is no fixed limit on the amount of temporary file space a connection can request.
- Any other value specifies the number of bytes.

max temp space per connection (1)

Description: Controls the maximum amount of temporary file space a connection can use before the request fails with an error message. The temp_space_limit_check option must be also be set to On (the default).

- A value of 0 indicates no fixed.
- Any other value specifies the number of bytes.

mpx autoexclude timeout (1)

Description: Specifies timeout for auto-excluding a secondary node on the coordinator node. **Note:** Does not apply to the designated failover node:

- 0 indicates that the nodes will not be auto excluded.

Default: 60

mpx heartbeat frequency (1)

Description: Specifies interval for the heartbeat thread to wake and clean up the connection pool on the secondary node.

Default: 60

mpx idle connection timeout (1)

Description: Specifies the time after which an unused connection in the connection pool on a secondary node will be closed.

Default: 600

mpx max connection pool size (1)

Description: Specifies maximum number of connections allowed in the connection pool on a secondary node.

Default: 10

mpx max unused pool size (1)

Description: Specifies maximum number of unused connections in the connection pool on a secondary node.

Default: 10

oem_string [SA-Level Option] (2)

Description: Stores user-specified information in the header page of the database file.

Default: Empty string

Priority [SA-Level Option] (2)

Description: Sets the execution priority level for requests from a connection. Cannot be higher than max_priority option. Settings: Critical, High, Above Normal, Normal, Below Normal, Low, Background.

Default: normal

query_mem_timeout [SA-Level Option] (2)

Description: Sets the maximum time, in milliseconds that a request waits for a memory grant:

- -1, the request waits for up to 50 times the estimated execution time for the request.
- 0, the request waits forever.
- Positive number: Sets the maximum time that the request waits.

Default: -1

request timeout [SA-Level Option] (2)

Description: Controls the maximum time (wall-clock time) that a single request can run:

- 0, the request waits forever.
- Max: 86400.

Default: 0

Reserved Keywords (1)

Description: This option allows users to designate individual keywords that are disabled by default. These can only be set for the PUBLIC group. User and temporary settings are not allowed.

Default: Empty String

Limitations: You cannot turn on the keywords SET, OPTION, and OPTIONS. The following determine whether a word is identified as a keyword (in order of precedence):

- It appears in the SQL Anywhere list of reserved words
- It is turned on with the RESERVED_KEYWORDS option
- It is turned off with the NON_KEYWORDS option

Version: This option was added in 15.4

secure feature key [SA-Level Option] (2)

Description: Enables features for the connection that were secured using the database server -sf option.

- -sk overrides the -sf option and specifies a key that can be used to re-enable all secured (disabled) features and gives that connection authority to change the features that are secured for all databases running on the database server.
- Can be set as a temporary option only

Default: Null

subquery caching preference (1)

Description: Controls which algorithm to use for processing correlated subquery predicates.

Normally used for internal testing and does not apply to IN subqueries.

Default: 0 = Let the optimizer choose.

subquery flattening percent (1)

Description: Allows the user to change the threshold at which the optimizer decides to transform scalar subqueries into joins. This option only applies to correlated scalar subqueries:

- 0 = The optimizer cost model decides
- 1 to $(2^{32} - 1)$ = The percentage of references at which to flatten.

Default: 100

subquery flattening preference (1)

Description: The percentage of references at which to flatten:

Default: 0 = Allow the IQ optimizer to decide to flatten subqueries.

synchronize mirror on commit (2)

Description: Controls when database changes are assured to have been sent to a mirror server when running in asynchronous or asynctfullpage mode. Set as temporary or for specific applications by examining the APPINFO string in a login procedure. On = each COMMIT causes changes recorded in the transaction log to be sent to the mirror server and acknowledged.

Default: Off

time series error level (4)

Description: Controls error handling for the time series functions that call the IMSL libraries.

Default: 0

time_series_log_level (4)

Description: Controls error logging behavior for the time series functions that call the IMSL libraries.

Default: 0

tsql outer joins (1)

Description: Controls the ability to use the Transact-SQL outer join operators *= and =* in statements and views. Setting this option to "On" allows the use of the deprecated Transact-SQL outer join operators.

Default: Off

updatable statement isolation (1)

Description: Controls whether connections can send query feedback to the statistics governor.

When set to Off, the statistics governor does not receive feedback about the health of statistics or fix any statistics for the specified connection. However, it does continue to monitor the usage of statistics and create or drop statistics based on their usage. Under normal circumstances, it should not be necessary to turn this option off.

Default: 0

APPENDICES

APPENDIX 1: LISTING OF DEPRECATED OPTIONS FOR IQ 15.0, 15.1, 15.2, 15.3 AND 15.4

Below, is a listing of IQ 12 options that were removed in 15.

#	OPTION NAME	IQ 12.7 ESD#11
1	ansi_integer_overflow	Off
2	auto_commit	Off
3	auto_refetch	On
4	automatic_timestamp	Off
5	bell	On
6	char_oem_translation	Detect
7	command_delimiter	;
8	commit_on_exit	On
9	convert_hg_to_1242	OFF
10	default_having_selectivity	0
11	default_like_match_selectivity	15
12	default_like_range_selectivity	15
13	describe_java_format	Varchar
14	disk_stripping	On
15	echo	On
16	enable_ordered_pushdown_insertion	ON
17	enable_thread_allowance	ON
18	flatten_subqueries	OFF
19	float_as_double	Off
20	headings	On
21	input_format	ASCII
22	iqmsg_length_mb	0
23	isql_command_timing	On
24	isql_escape_character	\
25	isql_field_separator	,
26	isql_log	
27	isql_plan	Graphical
28	isql_plan_cursor_sensitivity	ASENSITIVE
29	isql_plan_cursor_writability	On
30	isql_quote	'
31	java_heap_size	1000000
32	java_input_output	Off
33	java_namespace_size	4000000
34	java_page_buffer_size	50
35	local_kb_per_stripe	1
36	local_reserved_DBSpace_mb	200
37	log_detailed_plans	On
38	log_max_requests	100
39	main_cache_memory_mb	16
40	main_kb_per_stripe	1
41	max_work_table_hash_size	20
42	min_nlpdj_filtered_ppm	2500
43	min_nlpdj_table_size	10000
44	min_smpdj_or_hpdj_table_size	100000
45	mpx_global_table_priv	OFF
46	mpx_local_spec_priv	0
47	mpx_options	0
48	nulls	(NULL)
49	on_error	Prompt

50	optimistic_wait_for_commit	Off
51	optimization_logging	Off
52	os_option_crash	0
53	out_of_disk_message_repeat	120
54	out_of_disk_wait_time	30
55	output_format	ASCII
56	output_length	0
57	output_nulls	
58	parallel_gbh_enabled	ON
59	parallel_gbh_min_rows_per_unit	3000000
60	parallel_gbh_units	0
61	percent_as_comment	On
62	query_plan_on_open	Off
63	quiet	Off
64	return_java_as_string	Off
65	ri_trigger_time	After
66	screen_format	Text
67	sort_phase1_helpers	3
68	sqlconnect	
69	sqlstart	
70	statistics	3
71	temp_cache_memory_mb	12
72	temp_kb_per_stripe	1
73	thread_count	0
74	thread_stack	16384
75	thread_swaps	18
76	truncate_date_values	On
77	truncate_with_auto_commit	On
78	truncation_length	256
79	tsql_hex_constant	On
80	upgrade_database_capability	

APPENDIX 2: SP_DROPCONNONMAINUSED PROCEDURE

```
create procedure
//
// This procedure is to monitor 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 3: 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*blocksizeX2/2 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 ;

```



SYBASE, INC.

WORLDWIDE HEADQUARTERS

ONE SYBASE DRIVE

DUBLIN, CA 94568-7902 USA

Tel: 1800 8 SYBASE