SYBASE®

MIGRATING FROM SYBASE ADAPTIVE SERVER® ENTERPRISE TO SYBASE IQ

THE ENTERPRISE. UNWIRED.

WHITE PAPER

TABLE OF CONTENTS

1 Introduction 3

2 Architectural Overview 6

3 Schema Design: OLTP vs. Data Warehouse 9

4 Administration 13

5 Data Types 18

6 SQL Differences 21

7 Loading from ASE 24

8 Connectivity 29

9 Transaction Processing 37

10 Security 37

11 Stored Procedures 39

12 Sybase IQ Database Options 42

13 Reserved Word Differences 44

14 Using ASE and Sybase IQ Together 45

1 INTRODUCTION

This document was developed to provide guidance to DBAs who are migrating a Sybase Adaptive Server Enterprise (ASE) database to a Sybase IQ database. It will also provide information about the differences between these databases and may be helpful to users who may be considering separating and off-loading the reporting/query aspects of their application database from ASE (or other relational database engines) to Sybase IQ.

This document was prepared based on information current with Sybase IQ version 12.6 and Sybase ASE version 12.5.3.

SYBASE IQ HISTORY

A little history is relevant as it is important to understand some of the evolution of the Sybase IQ database. Sybase IQ was developed in the early 1990's by Expressway, Inc. (the database engine was called Expressway). Sybase acquired Expressway, Inc. in 1993 in a friendly acquisition. Sybase introduced the Expressway product in 1994 as IQ Accelerator, but renamed it shortly thereafter to Sybase IQ as version 11.0. Version 11.x required a Sybase ASE to run with Sybase IQ to provide connectivity and catalog resources. If you had occasion to use Sybase IQ in version 11 you would have found that it had many restrictions and a limited SQL vocabulary, but with the ASE catalog it had the feel and some of the functionality of an ASE database. With the advent of Sybase IQ version 12.0 the product was renamed to Adaptive Server IQ (ASIQ) and changed substantially with many new features and far fewer restrictions. In version 12 Sybase Adaptive Server Anywhere (ASA) replaced ASE for the catalog and connectivity, and there were major changes to the architecture, performance and scalability.

THE TECHNOLOGY BEHIND SYBASE IQ

The original designers of IQ (Expressway) had specific goals to address the needs for fast query performance against the largest of tables in a relational database environment. Another goal was to provide fast inserting of new records on a fully indexed table. Performance involving the manipulation of individual rows of data was another performance consideration. Sybase IQ today employs these same original design concepts but has been enhanced with other performance features which dramatically improved query speed and data management. Whereas ASE has been engineered to provide a good performance balance between OLTP and DSS over the years, Sybase IQ has focused on meeting the expanding needs of complex query and load performance required by data warehouse applications.

SOME BASIC DIFFERENCES BETWEEN ASE AND SYBASE IQ

Without getting into details (since most are addressed in the body of this document), there are several fundamental differences between these server products. Before creating a Sybase IQ database you should become familiar with some of these inherent differences as it may save you development and implementation time.

ONE SERVER—ONE DATABASE

A Sybase IQ server is designed to support only one database. When you connect to Sybase IQ you are "in the database" so there is no need to "use" a database after you connect. If issued, a "use" command is ignored.

DATABASE OWNER

The "DBA" user owns and administers a Sybase IQ database. The DBA user has the same authority as the "sa", and you will not find an "sa" account. For compatibility there is a "dbo" user provided. This will be discussed in more detail in the security chapter. The DBA user has all the authority required to create, build and maintain a database. These authorities can be delegated to other users.

SERVER CLIENTS

The native connectivity to Sybase IQ is ODBC, JDBC, and OLE DB. ADO.net is also supported. While Open Client™ connectivity is fully supported there are inherent differences with Open Client connections. These differences are discussed in this document and warrant particular attention by DBAs who are more comfortable using Open Client (isql and SQL Advantage) than ODBC clients.

Client tools provided with Sybase IQ include GUI clients dbisqlc written in C++, dbisql (Java™), Sybase Central™, and Open Client (version 12.5). The dbisqlc client uses ODBC connectivity with the Sybase IQ ODBC driver provided. The dbisql (Java) client can use ODBC or JDBC. Sybase Central uses JDBC exclusively.

Many Sybase ASE users are accustomed to using Open Client exclusively to connect to and manage Sybase products. The Sybase IQ GUI clients have many advantages over Open Client, particularly from Windows desktops, and we encourage you to use them. ODBC with the ASIQ driver provides robust connectivity and is used by most of the popular OLAP tools used in data warehousing. Using the Sybase Central Client with Sybase IQ is optional except when you want to enable a Sybase IQ Multiplex database. The Sybase Central Client provided with version 12.6 is (or will be) capable of administering other Sybase products including Sybase ASE, Replication Server and Adaptive Server Anywhere (ASA).

SYSTEM STORED PROCEDURES

Sybase IQ system stored procedures are all named with the prefix “sp_iq”, but there are a few ASA stored procedures available that are prefixed with “sa_”. ASE system procedures “sp_help” and “sp_who” do exist in a Sybase IQ database, but the same functionality exists using other Sybase IQ system procedures. This will be discussed in more detail in the stored procedure chapter. All Sybase IQ system stored procs are written in the “Watcom SQL” language. You can find the source code for these in the $ASDIR/scripts directory. Sybase IQ supports both Watcom SQL and T-SQL language. A chapter in this manual is dedicated to writing stored procedures in Sybase IQ.

DATA IS STORED BY COLUMN

This is one of the principal concepts that Sybase IQ employs for significant performance and storage gains. There are no data pages containing rows of data in a Sybase IQ database. A Sybase IQ “table” consists of a logical group of columns (also known as an Index Only Table) and every column is considered an “index.” Only the columns referenced in a SQL statement are manipulated and read into memory, leaving the other columns on disk. This storage method results in a dramatic reduction of disk I/O. For example, an UPDATE command on a column will only touch the pages holding data for the column, so page splits as a result of an update are not a concern as much as they might be in an ASE or other OLTP databases storing data by rows.

LARGE DATA PAGES

The Sybase IQ Data Page size may be 128K, 256K or 512K and is set when issuing a Create Database command. The default page size of 128K is recommended for most applications. Larger page sizes (256K and 512K) are recommended depending upon the number of rows in the largest table in a database. Larger pages contribute to reduced disk I/O since many column values are stored “on the page.” Since the Sybase IQ Page Size cannot be changed after a database is built, you need to carefully anticipate the growth and predicted future size of your database.

OPTIMIZED DATA STORAGE

Columns of data having a cardinality (number of unique values) less than 65,535 values can be automatically stored as 1 or 2 bytes by setting a database option before creating a table (recommended). In this scheme Sybase IQ creates a "look up" page that is used to "decode" the values when necessary. This storage technique can contribute to dramatic reduction of disk storage, and at the same time improves query performance by further reducing disk I/O. The database option which controls this storage behavior is "Minimize_Storage." and is 'Off' by default. It is recommended this option be set as 'On' before creating any tables in a database.

SQL LANGUAGE

Sybase IQ strictly uses the ANSI-89 standard with entry-level ANSI-92 features and core SQL99 features. There are some fundamental differences in SQL between ASE and Sybase IQ that may affect a few ASE applications. As an example, COMPUTE and COMPUTE BY statements are not available in Sybase IQ; however, the ANSI commands GROUP BY ROLLUP and GROUP BY CUBE provide similar functionality.

OBJECT OWNERSHIP

Unlike ASE, object ownership cannot be changed in Sybase IQ. Before you create any permanent objects in a Sybase IQ database, you should plan out an object ownership plan. Unless otherwise specified in a DDL command, the user issuing the DDL command will own the object created (DDL syntax does allow you to specify the object owner when issuing a CREATE command). A user other than the owner may need to fully qualify any objects referenced in SQL commands. This is different from ASE where the "dbo" typically owns all objects in a database, and users in a database do not need to fully qualify objects. To simulate an ASE environment many Sybase IQ sites typically create a Group to own all objects and have all the users included as members of the Group. Members of a Group do not need to fully qualify objects owned by the Group.

TRANSACTIONS IN SYBASE IQ

Sybase IQ transactions are different from ASE and have some interesting implications. Sybase IQ maintains data consistency in transactions using ANSI Isolation 3. This level is also referred to as "Snapshot Versioning." With Snapshot Versioning consistency is maintained on the table (or object) for the duration of a transaction. Every user that connects to a Sybase IQ database is presented with a "snapshot" of the most current committed versions of data tables. A user's view of all objects in a Sybase IQ database does not change until the user either disconnects (and reconnects) or issues a "commit" command in the database. There can be many users writing (or changing data) in a Sybase IQ database, but only one writer per table is permitted.

With the notable exception of Open Client connections (e.g. isql), transactions are managed by default in ANSI Chained Transaction Mode (this is configurable). Chained transaction mode means that every SQL statement in the database implicitly starts a new transaction which must be explicitly committed. (DDL commands are the only other exception as these commands are always automatically committed.) Even a SELECT command starts a transaction in Chained Mode. Open Client connections, however, are always in Unchained Mode (the default for an ASE database). An Open Client connection issues an implicit Begin Transaction and Commit Transaction for every command unless transaction logic is used in the SQL command. ODBC and JDBC connections are by default in Chained mode and do not issue implicit commits. The dbisqlc and dbisql clients provided as part of the Sybase IQ client software, as well as many third party ODBC clients, can be configured to 'auto pre-commit' and issue a commit before starting a new transaction either through a menu option or as an entry in the Windows registry.

SYBASE IQ MULTIPLEX DATABASE

Sybase IQ Multiplex consists of multiple Sybase IQ servers accessing a single database. In Sybase IQ Multiplex only one Sybase IQ server can "write" or change/modify data in the Sybase IQ Main Store (IQ Store). All other Sybase IQ servers are "Query Servers" that may only read data from the IQ Store. A Query Server may be configured with a "Local Data Store" allowing users to create permanent objects in a dbspace that is configured for use by a specific Query Server. Multiplex can consist of multiple Sybase IQ servers running on one host or multiple hosts each running a Sybase IQ server. A Sybase IQ Multiplex configuration with multiple hosts requires raw disk devices for the IQ Store (where data is stored). Local Data Stores for Query Servers may use raw devices or file systems. Multiplex has a number of performance advantages since database writes are always performed on one server with no performance impact on query servers. There is no direct communication or inter-dependence between servers in a Multiplex and, therefore, no impact if one server is down.

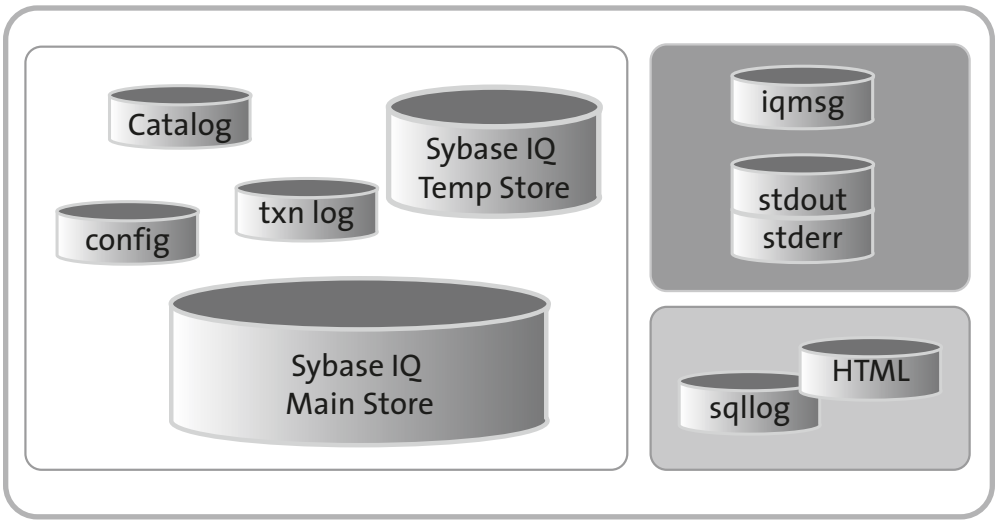
CROSS-DATABASE QUERIES

Customers often have more than one source of data to obtain real-time decision support information. Their business relies on fast cross-database joins between different databases. As stated above, a Sybase IQ server only supports a single database. CIS is provided in Sybase IQ, but there may be performance implications when joining objects between a Sybase IQ database and another server (Sybase IQ or ASE). This subject will be addressed later in this document.

2 ARCHITECTURAL OVERVIEW

Sybase IQ differs architecturally from the ASE server in several important ways. The most conspicuous difference is that Sybase IQ is implemented as a single database, breaking the many databases within a database server paradigm. Instead, Sybase IQ is installed as a single server/single database.

The Sybase IQ instance is really just a collection of files and raw disk devices. Each file performs a specific purpose and all files are required for the proper operation of the Sybase IQ server.



Database Catalog—This file is stored as databasename.db and contains the system tables and views as well as any stored procedures, views, users and groups, events and user-defined data types. The file is also referred to as the "database file."

Database Transaction Log—This file defaults to databasename.log and contains the transaction log for the Sybase IQ database. The name can be set beyond the default through the command-line database create command. This log is used to manage transactions in the Database Catalog (above) only. The file needs to be managed infrequently, using a utility program.

Server Configuration File—This file defaults to params.cfg and contains the start-time Sybase IQ server configuration parameters.

Sybase IQ Main Store—Also called the IQ Store, this file is the main unit of storage for the Sybase IQ database itself. The general naming standard for this file is databasename_01.iq, but the name can be set to any value during database creation or dbspace additions. Most Sybase IQ databases will be composed of several of these files, each representing a dbspace. The dbspace is the logical device within the Sybase IQ server that points to the physical device. Raw devices are recommended.

Additional dbspaces are created as space is needed for database growth. It is not at all uncommon for the Sybase IQ instance to have many dbspaces allocated for the Sybase IQ Data Store.

Transactions of data in the IQ Main Store are managed internally with this dbspace so there is no external transaction log of data transactions for the DBA to manage.

Sybase IQ Temporary Store—This file is the unit of storage for the Sybase IQ temporary workspace used for resolving joins, order by or group by results as well as many other functions within Sybase IQ. Like the IQ Main Store, the temporary workspace can also be made up of several logical dbspaces, pointing to separate physical devices.

SYBASE IQ OUTPUT

STANDARD FILES

Sybase IQ Message File—This file is the main output of the Sybase IQ database-specific messages. The file is named databasename.iqmsg. All database-related operations are echoed into this file, including error messages and warnings, query plans and transaction tags.

Other Message Files—By default, the <IQservername>_nnn.stderr and <IQservername>_nnn.stdout files are used to echo server-related messages. The file names are numbered an increment by one every time the Sybase IQ server is started. These files are created and referenced in the start_asiq server utility that is included with the Sybase IQ database creation. If a custom utility is written to start or stop the Sybase IQ server, these files would not be specifically generated.

SPECIALIZED FILES

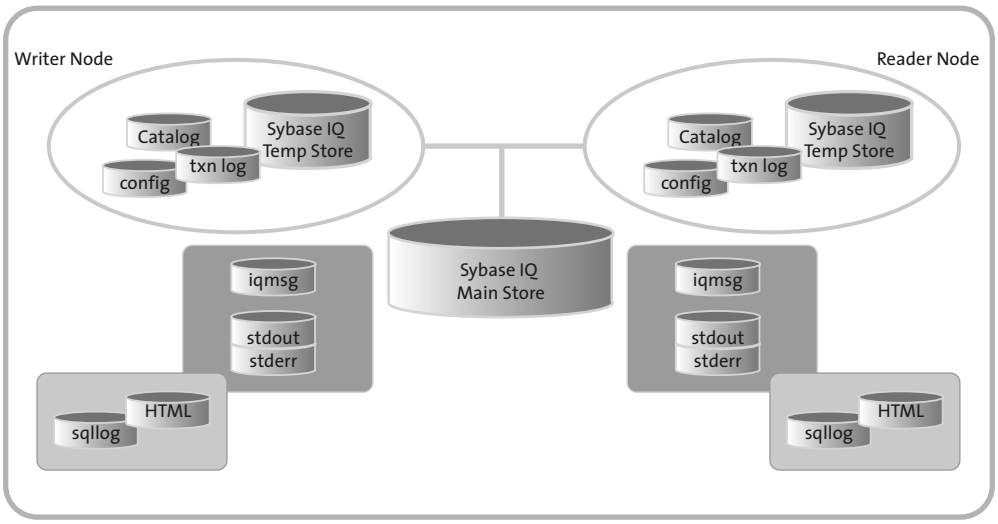
SQLLOG File—This file provides output specific to any queries run against the Sybase IQ database. The verbosity level as well as the file location are configurable in the params.cfg file through the -zr and -zo options.

HTML Query Plans—These files contain html query plans for either all queries run on the server or for an individual query based on a database option. This setting is modified through the query_plan_as_html setting on the server. By default, HTML query plan output files are written in the directory containing the Sybase IQ Message File (db_name.iqmsg.) but may be written to a specific directory on the host using the database option Query_Plan_As_HTML_Directory.

SYBASE IQ MULTIPLEX

The Sybase IQ Multiplex architecture allows for a much more scalable implementation of the database. Through the use of this architecture, many Sybase IQ instances can share the Sybase IQ Data Store simultaneously, allowing many more queries to be executed against the database without any noticeable performance degradation.

The Multiplex installation is composed of many host machines connected to a shared disk through a shared I/O backplane. This is most commonly accomplished through fiber controllers, but could be implemented through other technologies.



The Sybase IQ instances installed on these host machines are designated as either query servers or as a write server. There is only one write server allowed in the multiplex installation, but many query servers can be present. The creation of these servers is accomplished through the Sybase Central.

The files used by the various multiplex servers will have their own copy of all files described above with the exception of the Sybase IQ Main Store. The Sybase IQ Main Store is shared among all the database servers contained within the multiplex.

Sybase IQ Local Store (not shown above) is a dbspace that may be created for exclusive use by a Sybase IQ Multiplex Query Server. This feature allows users connected to that query server to create, store and maintain persistent tables or objects.

SCHEMAS

Since there is only one database per server, the Sybase IQ instance will never resemble that of an ASE server with multiple databases. However, there are methods available to logically segregate a Sybase IQ database into multiple schemas that more closely resemble the multiple databases in ASE.

Using the object ownership of specific users, separate schemas can be created to partition data so that logical "databases" can be formed. The use of these object ownership rules will allow for a separate database schema to be created such that referencing each schema is controlled through the use of object permissions.

For instance, a schema created by the MKTG user, such as a MKTG.customer would be different than a schema created by an ACCT user with an ACCT.customer table. The fully qualified name dbname.ACCT.customer could be used to reference this table but the dbname qualifier would be for documentation purposes only, and is ignored by Sybase IQ, much like the "USE" statement would with ASE.

3 SCHEMA DESIGN: OLTP VS. DATA WAREHOUSE

There are fundamental differences between an OLTP database and a database designed for data warehousing. Some notable differences:

OLTP DATABASE

Handles many predictable transactions per day

Designed for performance when processing individual rows of data

Usually has narrow tables to permit more rows per page, resulting in fewer disk reads

Uses 3rd Normal Form (NF) with many tables and minimal redundancy

Uses a minimal number of indexes

DATA WAREHOUSE

Designed for performance when processing large amounts of data

Queries tend to be unpredictable (ad hoc) but with some standard reports

Queries search many rows and often join several tables

Has fewer and wider tables with some data redundancy (rarely using 3rd NF)

Has many indexes

Handles many unpredictable queries per day

DATA WAREHOUSE DESIGN BASICS

The tables in a data warehouse are usually designed around how users query the data. Many data warehouses employ a Star Schema design consisting of a Fact Table and several Dimension Tables. Other designs may employ wide denormalized tables. If an OLAP tool will be used, the data warehouse design may need to conform to table design specifications required by the query tool. (Note: Sybase IQ has several built-in OLAP functions including cube, rank, rollup and others.)

UNDERSTAND WHAT SYBASE IQ DOES WELL

As in any database, the design should take advantage of the performance features offered by the database engine.

WIDE TABLES

Take advantage of the column-based storage by creating wide denormalized tables. Query performance is typically better when there are fewer joins. Sybase IQ tables can have hundreds or even thousands of columns (some limitations apply for columns defined as char() or varchar() larger than 255 bytes). "Look up" tables typically do not need to be denormalized. Joins with look up tables typically use Sybase IQ Hash Joins, which provide excellent table join performance.

SYBASE IQ INDEXES

Before designing tables, you should have a thorough understanding of all Sybase IQ indexes, especially the Low Fast (LF) and High Group (HG) Indexes. The LF and HG indexes are "enumerated" (optimizer knows the cardinality and distribution for each value) and provide very fast query response for equality searches, some range searches and Group By operations.

When you create a table in Sybase IQ each column will have the FP (Fast Projection) index created by the default. This index stores the data value for the column and is used by many different query operations. The database option "Minimize_Storage" should be set 'On' before creating any tables. This option will create the "optimized" Fast Projection (FP) index on all columns with less than 65,536 values. Optimized FP indexes dramatically reduce storage and will make queries run faster.

You will need to create additional Sybase IQ indexes for optimal query performance. Columns used in search arguments and other query functions typically have 2 and sometimes 3 indexes. With Sybase IQ you typically create these indexes *before* loading data, although indexes can be created after loading data. Sybase IQ has been designed to make it more efficient to load data into a fully indexed table rather than load data and create indexes later.

The Primary Key and Unique constraints automatically create a Unique HG index on the column(s) defined for the constraint. A Foreign Key constraint automatically creates a non-unique HG index on the column(s). If the constraint is complex (more than one column), you will need to create an appropriate index (LF or HG) on each column of the key to support searches on the key components. The Primary Key and Unique constraints are very important to the Sybase IQ query optimizer in resolving table joins and are recommended for optimal query performance.

TABLE DESIGN

Create a Primary Key on your tables whenever possible. Primary Keys are used extensively by the query optimizer for table joins.

Define all columns explicitly as NULL or NOT NULL. The default behavior between ASE and Sybase IQ are opposite. Sybase IQ assumes the column is NULLABLE unless specified as NOT NULL. In ASE the exact opposite is true. (Note: If using Open Client to create a table, NOT NULL will be assumed if not specified.)

DATA TYPES

Choosing data types is critical for optimal performance and efficient storage. The table below illustrates the data types supported on ASE and Sybase IQ. Data types are discussed in greater detail in Chapter 5 of this document.

Type	Sybase IQ	ASE	Notes
bit	✓	✓	
tinyint	✓	✓	
smallint	✓	✓	
integer	✓	✓	ASE does not support unsigned integer
bigint	✓		Sybase IQ supports signed and unsigned bigint
char	✓	✓	ASE char() is page size dependent
varchar	✓	✓	ASE varchar() is page size dependent

long varchar	✓		Sybase IQ uses CLOB data type and may be unlimited size
text	✓	✓	Sybase IQ uses CLOB data type and may be unlimited size
nchar/nvarchar		✓	
unichar		✓	
binary	✓	✓	Sybase IQ max is 255. ASE is page size dependent
varbinary	✓	✓	Sybase IQ max is 32k. ASE is page size dependent
long binary		✓	
image	✓	✓	Image is a synonym for long binary in Sybase IQ
date	✓	✓	
time	✓	✓	
datetime	✓	✓	
smalldatetime	✓	✓	smalldatetime is a synonym for datetime in Sybase IQ
timestamp	✓	✓	Timestamp is binary(8) in ASE. In Sybase IQ is a datetime
numeric	✓	✓	Default precision and scale are very different
decimal	✓	✓	Default precision and scale are very different
float	✓	✓	Storage differs
double	✓	✓	
java types	✓	✓	Sybase IQ stores java types in the System dbspace (catalog)
identity	✓	✓	Implementation differs between ASE and Sybase IQ

If you intend to use Open Client connections to Sybase IQ you cannot use unsigned integer or bigint data types in a Sybase IQ database. (Support for some Sybase IQ data types is dependent on the version of Open Client.)

Chapter 4 of the Sybase IQ Reference Manual lists data types and their storage requirements. Here are a few tips (but remember the Open Client restrictions mentioned above):

Note the amount of storage used by the Numeric/Decimal type. In Sybase IQ the storage costs increase dramatically when precision exceeds 18. Sybase IQ and ASE use

different sizes for default precision and scale. ASE default is precision 18 and scale 0. Sybase IQ default precision is 126 and scale 38.

Choose Date over Datetime. If queries never use the time component of a Datetime column, use a Date data type. The Date type only uses 4 bytes versus 8 bytes for a Datetime. Also make sure that you create a Date index on the column. In Sybase IQ there is no difference in storage between Smalldatetime and Datetime, as both will use 8 bytes.

Choose Char over Varchar when the column width is less than 256 bytes. A Varchar field requires an additional byte to store the length. The exception to this rule is for character fields that are concatenated in queries. For those columns you must use varchar because the results are blank padded to the full length of the field.

Data types for join columns should be as small as possible and match data types exactly. Numeric data types are preferred over character types, and unsigned integers are preferred over signed integers (query optimizer does not need to check the sign of the value).

REFERENTIAL INTEGRITY

Referential Integrity (RI) can be implemented and enforced in a Sybase IQ database. RI is beneficial for query performance particularly when a "partitioned" fact table is employed (discussed below). The query optimizer uses RI to determine column cardinality between the keys of a "parent" and "child" table or tables. There is an estimated 5% (max) performance degradation during loading when RI is enforced, but you can also temporarily disable RI while loading data by using the "Disable_RI_Check" database option.

For deletes of a Primary Key, Sybase IQ only supports the RESTRICT action on Foreign Keys, whereas ASE permits SET NULL, CASCADE, DEFAULT and RESTRICT actions.

DEFAULT CONSTRAINTS ON COLUMNS

Default constraints are not currently supported in Sybase IQ. The work around is to use an update command after loading to set the value for a column where value is NULL.

TRIGGERS

Triggers are not currently supported in Sybase IQ. Some triggers may be implemented as Events in Sybase IQ.

PARTITIONING A (FACT) TABLE

You will discover, as table size starts to exceed 100 million rows, your incremental load times to the table will continue to increase. HG index processing during loads (and deletes) is the cause for this behavior. For these large tables there is a strategy to horizontally partition the single large table into several tables with the same columns, but you should only consider this technique for tables having several hundred million rows. Partition the tables into increments of approximately 100 million rows and create a View using the UNION ALL operator to present all the partitions as a single table to users. With one important exception, the optimizer will fork queries through the view and process the query in parallel against each "leg" of the view. The exception is a query that uses a SELECT DISTINCT operator. Such queries CANNOT be run in parallel with a view and usually result in poorer performance than against a single table.

This design technique provides faster and predictable Load (and Delete) performance. This strategy is particularly recommended for very large tables (hundreds of millions of rows or more) and where data is rolled in and out over time. In the case of Deletes you may be able to use a TRUNCATE table command instead of delete. Truncate table will outperform a Delete operation.

4 ADMINISTRATION

ADMINISTRATION DIFFERENCES

The administration of Sybase IQ is similar to but much simpler than Sybase Adaptive Server Enterprise. In concept, the tasks are identical, i.e. installation, configuration, administration processes, etc. However, there are subtle differences that must be considered. The following sections describe by administration task the most important differences.

INSTALLATION AND CONFIGURATION

Sybase IQ installation is very straightforward. Sybase supplies an installation utility (sybinstall) on the delivered CD that will perform most installation tasks for initial installation as well as the application of any SWR patches.

There are two installation tasks that must be performed manually. One of these tasks is required before you execute the automated install and one of the tasks after you perform the automated install. Both tasks relate to setting host Environment variables (except Windows).

PRE-INSTALL TASK

Before running sybinstall, the Sybase environment variable should be set to point to the root directory under which you will install Sybase IQ. This is the starting point for the initial Sybase IQ installation, and all other environment settings will be based on this path.

In addition, the sybinstall utility should set the following environment variables as a normal part of the install. These variables will be found in scripts located in $SYBASE/ASIQ-12_6. You will need to copy these scripts to set variables for the Sybase user profile.

Variable	Setting	Purpose
ASDIR	${SYBASE}/ASIQ-12_6	Location of Sybase IQ root directory
PATH	${ASDIR}/bin	Location of Sybase IQ binaries
ASJRE	${SYBASE}/shared/jre-1_22	JRE location for Open Client 12
SYBASE_JRE	${SYBASE}/shared/jre-1_22	JRE location for Open Client 12

A library path is set to point to the location of the shared libraries:

Platform	Variable	Setting
AIX	LIBPATH	${ASDIR}/lib
HP UNIX®	SHLIB_PATH	${ASDIR}/lib
Solaris	LD_LIBRARY_PATH_64	${ASDIR}/lib
Windows	PATH	%ASDIR%/win32

The OCS directory is set to point to the location of the Open Client shared libraries:

```
SYBASE_OCS="OCS-12_5" SYBASE_OCS = "ocs-12_0"
```

The $SYBASE naming convention will allow the Sybase IQ installation to be shared with an existing ASE since the ASDIR convention is to place all files in an ASIQ-12_6 directory structure, with no risk of overwriting the existing ASE files.

This applies to all but the Open Client libraries. In situations where both Sybase and ASE will share a host, it may be necessary to manually combine the shared libraries into a single SYBASE_OCS directory. This should be done by renaming the existing SYBASE_OCS directory prior to installation, then copying the files from the SYBASE_OCS directory from the Sybase IQ install back into the renamed directory, without overwrite, followed by a rename to put the SYBASE_OCS directory back to its original location.

POST-INSTALL TASK

Upon the completion of sybinstall, ensure that all environment variables are set properly. The easiest way to accomplish the test is to start the Sybase IQ demo database. The command to start the database should be run from the $ASDIR/demo:

```
start_asiq @asiqdemo.cfg hostname_asiqdemo.db
```

As the Sybase IQ demo database starts, a boot log will be echoed back to the console from where the server was started.

If the installation is correct, the Sybase IQ server should start and run as a background task.

OTHER DIFFERENCES

ASE will create a server during the install process, requiring the creation of master, model and tempdb databases. Since the architecture of Sybase IQ is based on the one-database-one-server paradigm this step is not necessary.

Based on this architecture, these same functions are provided through an equivalent Sybase IQ component. The master database is represented in Sybase IQ as the Database Catalog (.db file). The tempdb database is represented by the Sybase IQ Temporary Store, while the model database is a separate Sybase IQ instance known as utility_db. Simply running “start_asiq -n enginename” on any platform will start the utility_db. This instance provides for the minimum Sybase IQ functionality and can be used to create new databases or restore an existing Sybase IQ database from backup.

BASIC SYBASE IQ SERVER COMMANDS

The basic administration functions are significantly different between Sybase IQ and ASE. Automating the starting and stopping of the servers or monitoring the running Sybase IQ instances may be a departure from the way these processes are handled in ASE.

One item of particular note is that the ASE “sa” login will roughly correlate to the Sybase IQ “DBA” login. All administrative tasks within Sybase IQ should be performed as the DBA login. Another difference is that while the ASE “sa” login initially has no password, the Sybase IQ “DBA” login has an initial password of “SQL”.

STARTING THE SYBASE IQ SERVER

A startup script has been provided to start the Sybase IQ server; the start_asiq executable shell is found in $ASDIR/bin and simplifies the process of server start. This script generally equates to

the startserver command in ASE, so a higher-level script may be developed to provide any custom functionality used for the ASE server for general housekeeping such as message log truncation, etc.

```
start_asiq @params.cfg databasename.db
```

STOPPING THE SYBASE IQ SERVER

To stop a Sybase IQ server, you have a choice of several methods:
- From a command line, use the stop_asiq command line utility
- From a user connection, use the STOP DATABASE command
- From a command line, dbstop -y -c dsn=IQ_SERVER_DSN.

The stop_asiq is the most commonly used method; stop_asiq by default will prompt the user to confirm stopping one or more Sybase IQ servers. There are optional parameters to stop_asiq—-force one and -force all—which will stop the server without user interaction.

The stop database command roughly correlates to the ASE shutdown command. There is no equivalent to a shutdown with nowait for Sybase IQ. Shutdown for Sybase IQ is essentially a "kill -15" for the asiqsrv12 O/S process. Many customers have developed automated scripts to stop the server through external processes using this method.

Any current connections/transactions will be "rolled back," meaning that any allocated temporary work pages or any unnecessary active version pages may have to be deallocated prior to the server coming down. If necessary, the "kill -9" should be used as a last resort.

WORKING WITH DATABASES

CREATING A DATABASE

Database creation in Sybase IQ is significantly different than ASE. Remember that the Sybase IQ follows the one-database-one-server paradigm. Therefore, when you create a new database you are creating a Server and Database combination.

There are two ways of creating a new database: either through Sybase Central wizard or using a command script while connected to a Sybase IQ server. At this time, the Sybase Central version shipped with Sybase IQ is the only version capable of creating, managing and administrating a Sybase IQ database. This tool contains specialized options for managing a Sybase IQ database which are generally self-explanatory. The create database utility is a wizard-driven tool which guides you through the various decisions that need to be made while creating a Sybase IQ database. When using Sybase Central to create a database the Sybase Central Agent must be running on the host. (If the Agent is not running you will need to start the Agent manually. The executable is located in $SYBASE/bin directory as S99SybaseIQAgent. For Windows the Agent is set up as a Service to start automatically.)

Creating a database from the command line requires that a current Sybase IQ instance is up and running, since the create function is essentially a copy. This works much like the ASE model database except that custom configurations are not inherited in the new database. Any current Sybase IQ instance can be used to perform this function but the most commonly used databases are either utility_db or the Sybase IQ demo database.

The unit of allocation for disk storage in Sybase IQ is the dbspace. The dbspace works much like the ASE devices in that there are specific commands to create and drop them and they can be built on either raw partitions or file systems.

When you create a database, four dbspaces are automatically created by default:

dbspace Name	Contents	Default O/S file name
IQ_SYSTEM_MAIN	Main (permanent) Sybase IQ Store file	dbname.iq
IQ_SYSTEM_TEMP	Temporary Sybase IQ Store file	dbname.iqtmp
IQ_SYSTEM_MSG	Message log file	dbname.iqmsg
SYSTEM	Catalog Store file	Dbname.db

The SYSTEM dbspace contains the system tables which hold the schema definition as you build your database. It also holds a separate checkpoint log and rollback log.

In addition to these database files, the database server also uses a temporary file to hold information needed during a session. This temporary file is not the same as the Temporary Sybase IQ Store, and is not needed once the database server shuts down. The file has a server-generated name with the extension *.tmp. Its location is determined by the TEMP environment variable, or the ASTMP environment variable on UNIX.

WORKING WITH DATABASE DEVICES

The fundamental building block for Sybase IQ databases is the dbspace. However, a dbspace is different from ASE in that it is composed of one contiguous segment, rather than the multiple segments (log, data, default) of an ASE device.

When you create a new dbspace, it has no contents. When you create tables and indexes and load them, Sybase IQ distributes the data as equally as possible among any existing dbspaces that are not already full. This technique optimizes performance.

Sybase IQ provides utilities to allow the DBA to manage dbspaces in the Sybase IQ Store. These include the following (consult the Sybase IQ documentation for details):

Marking a dbspace as Read Only. A dbspace may be marked as “read_only” to prevent any new writes to that dbspace.

Relocating Objects in a dbspace. Objects in a dbspace may be relocated to other dbspaces to facilitate the removal of a dbspace from a database.

Enlarging a dbspace. It is possible to enlarge a dbspace provided the DBA reserved space for the “freelist” when the dbspace was initially created.

New dbspaces are added either through Sybase Central or through the create dbspace command. When working with Sybase IQ Multiplex databases you must use Sybase Central to add any new dbspaces.

WORKING WITH TABLES

There are no major differences in working with Sybase IQ tables; however, there are some ASE features that are not implemented in Sybase IQ. The major difference is that Sybase IQ does not support the ASE Default property for columns.

Temporary tables work in a similar manner to ASE for session-level (#temp) tables. However, Sybase IQ also provides ASCII syntax for creating the temp tables using declare local temporary table.

In addition to non-persistent temporary tables, Sybase IQ does provide for global temporary tables. A global temporary table allows multiple users to share the table schema. Each user maintains their distinct data much like the tempdb.tablename in ASE except that the data within a global temporary table is partitioned so that a user can only see the data inserted by that user.

WORKING WITH INDEXES

The index structure of Sybase IQ is significantly different from ASE; however, the theory behind the index usage and creation is not very different from any other database server. Indexes are created to improve performance for filtering and aggregating data.

Sybase IQ Indexes have the following characteristics:

- A default index (FP) is automatically created on each column by the create table statement.
- A Unique Index (similar to an ASE index) is automatically created on any Primary Key or Unique Key constraint statements. A non-unique index is created on columns defined as a Foreign Key.
- The DBA creates the additional Sybase IQ Indexes based upon Data Type, Column Cardinality, and expected usage in queries.

To achieve maximum query performance, however, you should choose one or more additional index types for most columns that best represent the cardinality and usage of column data:

Index	Description
Compare (or CMP)	Stores the binary comparison (<, >, or =) of any two distinct columns with identical data types, precision, and scale.
DATE	An index on columns of data type DATE used to process queries involving range searches on date quantities.
Datetime (or DTTM)	An index on columns of data type DATETIME or TIMESTAMP used to process queries involving range searches on datetime quantities.
High_Group (or HG)	An enhanced b-tree index to process equality and group by operations on high-cardinality data of more than 1,000 distinct values.
High_Non_Group (or HNG)	A non value-based bitmap index ideal for most high-cardinality DSS operations involving ranges (other than Date & Datetime) or aggregates.
Low_Fast (or LF)	A value-based bitmap for processing queries on low-cardinality data, recommended for up to 1,000 distinct values.
TIME	An index on columns of data type TIME used to process range queries involving time quantities.
Word (or WD)	Used to index keywords by treating the contents of a CHAR or VARCHAR column as a delimited list.

OTHER DIFFERENCES

Sybase IQ supports stored procedures, views and some referential integrity. There are no major differences between ASE and Sybase IQ with respect to views. Referential integrity is supported in Sybase IQ through the primary key, foreign key and unique constraints.

Stored procedures are covered in another section of this document.

BACKUP AND RECOVERY

Backup and recovery is similar to ASE, with the exception that there is no Backup Server with Sybase IQ. Sybase IQ provides the ability to back up to Tape and/or Disk.

Sybase IQ supports the same type of backups that ASE supports. These include:
- Full Backup
- Incremental since last Full Backup
- Incremental since last Backup (of any type)

In addition, Sybase IQ also has the ability to perform what is called a "Virtual Backup." The Virtual Backup is an O/S level backup that allows a fast database copy through O/S level commands rather than through the database. This backup type is maintained in the Sybase IQ engine so that incremental backups can be performed even though the full backup was not performed by the engine itself.

5 DATA TYPES

Detailed information on the Sybase IQ data types can be found in Chapter 7 of the Adaptive Server® IQ Reference Manual.

BIT DATA TYPE

The BIT data type is used for storing Boolean values: 1, 0 or NULL. Because of the nature of Sybase IQ, this is the most efficient data type available and its use for switches and flags is strongly recommended.

Sybase IQ will only permit a true bit value (0 or 1) to be stored in a BIT data type; non-zero values are stored as 1 (TRUE). ASE and Sybase IQ will both implicitly convert integer data types to BIT.

For other data types ASE supports implicit conversion of BINARY to BIT while Sybase IQ does not (but will be supported in version 12.6). In a situation where this conversion is necessary, the use of the cast function is recommended. For example, select cast (binarysourcecol as BIT) from … where the value of the binarysourcecol is either 0 or 1.

INTEGER DATA TYPES

Integer data types are used for storing exact numeric values, where accuracy is required for arithmetic operations. Sybase IQ supports four integer data types, each able to store progressively larger numbers and capable of reducing storage requirements for exact numeric data.

DATA TYPE	Capacity	Storage Size	Available Range
BIGINT	signed 64-bit	8 bytes	-9223372036854775808 to 9223372036854775807 (signed) 0 to 18446744073709551615 (unsigned)
INTEGER	signed 32-bit	4 bytes	-2147483648 to 2147483647 (signed) 0 to 4294967295 (unsigned)
SMALLINT	signed 16-bit	2 bytes	-32768 to 32767
TINYINT	unsigned 8-bit	1 byte	0 to 255

The main difference between ASE and Sybase IQ with respect to the integer data types is the support for 64-bit and unsigned integers. In the current ASE release, unsigned integers are not supported. Sybase IQ supports both UNSIGNED INT and UNSIGNED BIGINT and the use of these data types is recommended where negative numbers are not expected.

More importantly, Sybase IQ also supports the use of 64-bit exact numeric values through the BIGINT data type. Keep in mind that 64-bit integers are not supported in both ASE and Sybase Open Client. This is usually not a migration issue since these data types will not exist in the current application. However, if this large exact numeric data type is implemented in Sybase IQ, issues can arise in any Open Client, CTLIB or DBLIB application that references this column, such as the use of proxy tables in ASE through a CIS connection to a Sybase IQ database. In order to avoid this issue, the BIGINT value must be cast as NUMERIC or DECIMAL in the application to avoid truncation of the column as it is returned.

CHARACTER DATA TYPES (CHAR, VARCHAR, TEXT, CLOB)

The character data types are used for storing strings of letters, numbers and symbols.

During insert operations, ASE will trim trailing blank spaces from VARCHAR values but Sybase IQ does not. Sybase IQ will pad all VARCHAR values up to 255 bytes. Since any VARCHAR column is blank padded up to 255 bytes, VARCHAR data types will use an extra byte to hold the length. The CHAR type does not require the extra storage byte and is recommended for use over varchar for this reason (the extra byte starts to add up when you are storing hundreds of millions of records).

Sybase IQ will allow up to 32k -1 to be stored in the CHAR or VARCHAR data type. This roughly emulates the TEXT data type in ASE, but is limited to 32k. For unlimited large character objects in Sybase IQ use the CLOB data type. The CLOB type can store data as large as 2 PB depending on the Sybase IQ Page Size. (CLOB types require a special license option. Consult the "Large Objects Management in Sybase IQ" manual for more information.)

ASE CHAR and VARCHAR sizes depend on the logical page size; 2k, 4k, 8k and 16k page sizes result in maximum sizes of 1962, 4010, 8106 and 16298 bytes, respectively.

ASE supports NCHAR, NVARCHAR, UNICHAR, UNIVARCHAR data types N for multi-byte character sets.

ASE supports Unicode in CHAR data if the server charset is UTF8, while Sybase IQ supports Unicode in the CHAR and VARCHAR data types instead of having a separate data type for them.

BINARY DATA TYPES (BINARY, VARBINARY, IMAGE)

The BINARY data types are used for storing raw binary data, such as pictures, in a hexadecimal-like notation of infinite length.

In ASE, the maximum size of BINARY and VARBINARY depends on the logical page size:
- 2k page size - max size of a column can be as large as a single row or 1962 bytes
- 4k would be 4010 bytes, 8k would be 8016 bytes and 16k would be 16298 bytes.

Sybase IQ supports up to 255 bytes for BINARY, 32k for VARBINARY and 64k for LONG BINARY and IMAGE. A special license is required to use the LONG BINARY data type (see the "Large Objects Management in Sybase IQ" manual for more information).

Sybase IQ supports ASE binary display formats; i.e. if '123' is entered in a BINARY field the Sybase IQ display format is by bytes, i.e. '123'; the ASE display format is by nibbles, i.e. '0x616263'.

Sybase IQ does not currently support implicit conversion of BINARY to BIT (available in version 12.6).

DATE, TIME, DATETIME, AND TIMESTAMP DATA TYPES

The Sybase IQ data types DATE, TIME, DATETIME and TIMESTAMP are all used for storing the various date formats. DATETIME, SMALLDATETIME and TIMESTAMP are synonymous and use 8 bytes of storage. Sybase IQ also supports a DATE data type (4 bytes) and TIME data type (8 bytes).

In Sybase ASE, the timestamp data type does not correspond to a valid datetime but is an 8-byte binary number designed to support the dbLib and CT-Lib browse mode. Some users have implemented it as a means of application optimistic concurrency detection or as a unique primary key. In ASE, the timestamp cannot be compared directly, but must be compared with tsequal() TSQL function. Additionally, no user direct manipulation (insert or update) of the timestamp value is allowed. In Sybase IQ, the timestamp data type equates to datetime; therefore, users with ASE-based applications that contain ASE timestamp data will need to store the data in Sybase IQ as binary(8) if the data is necessary and take into consideration the insert/update of the data as required by the application.

ASE defaults to displaying dates as 'MMM-DD-YYYY' while Sybase IQ defaults to the ISO 'YYYY-MM-DD' format. (This is behavior for ODBC connections, but TDS connections will behave like ASE.) This behavior can be changed through database or server options.

NUMERIC DATA TYPES

The NUMERIC data type is an exact numeric data type; its accuracy is preserved to the least significant digit after arithmetic operations. Its maximum absolute value is the number of nines defined by [precision—scale], followed by the decimal point, and then followed by the number of nines defined by scale. The NUMERIC data type is the same as the ASE DECIMAL data type and DECIMAL is a valid data type name in Sybase IQ.

The main differences between ASE and Sybase IQ involve the default precision and scale for the stored values within these data types. Because of the 64-bit capabilities of Sybase IQ, the default numeric precision can be set to 126 with a scale of 38. In ASE, the default precision is 18 with a scale of 0. (There are also width limitations for TDS connections to Sybase IQ that will prevent you from using larger numeric data types.) For Sybase IQ it is strongly recommended that you specify precision for these data types in Create Table commands to get optimal storage (see Chapter 7, Sybase IQ Reference Manual for storage requirements for numeric/decimal types).

APPROXIMATE NUMERIC DATA TYPES

The FLOAT and DOUBLE are approximate numeric data types and are subject to rounding errors after arithmetic operations. ASE differs from Sybase IQ in how the FLOAT data type is interpreted and when to implicitly create a 4-byte or 8-byte data type based on the size of the value to be stored.

By default, Sybase IQ FLOAT values are interpreted by the server as REAL values using 4 bytes of storage. Unless the column or variable is defined with the correct precision, any high-order bytes will be truncated if the value to be stored is larger than 16 significant digits.

Sybase IQ offers the FLOAT_AS_DOUBLE option to control data width. Since ASE treats its own FLOAT values as DOUBLE, enabling this option instructs Sybase IQ to treat FLOAT values in the same manner as ASE.

JAVA DATA TYPES

Sybase ASE supports Java data types in the database while Sybase IQ supports Java data types for tables only in the SYSTEM catalog.

6 SQL DIFFERENCES

As Sybase IQ matures, the SQL differences between it and Sybase ASE are shrinking. The SQL differences can be split into three main categories: system options, runtime SQL, and transaction control.

- There are runtime SQL differences between the two products. However, these changes are fairly minor in the day-to-day operation of most data warehouses. They do come into play, however, with stored procedures and batch SQL processes.
- There are a few differences with respect to transactions and how the two engines treat them. While not technically a SQL difference, it does warrant being in this section due to the impact it has on the SQL being written.

SYSTEM OPTIONS

System options refer to those options that are set to a value that contradicts the ASE behavior. In order to force Sybase IQ to behave more like ASE these options would be changed from their default values.

Important Note: All ASE compatibility options are set for you, if you connect through TDS by the stored procedure sp_tsql_environment. You can edit this stored procedure to specify the settings you want.

ALLOW_NULLS_BY_DEFAULT

The default behavior in Sybase IQ is to create a table in which all columns allow nulls by default. In ASE the behavior is to not allow nulls by default. To make Sybase IQ behave like ASE, this option would need to be set:

```
set option public.allow_nulls_by_default = 'off';
```

QUOTED_IDENTIFIER

The default behavior in Sybase IQ is to treat single and double quotes as two distinctly different operations. A single quote is used to denote a string, while double quotes are used to denote object names. In ASE there is no difference. To make Sybase IQ behave like ASE, this option would need to be set:

```
set option public.quoted_identifier = 'off';
```

STRING_RTRUNCATION

When data is changed in a table, Sybase IQ and ASE behave slightly differently when that data exceeds the bounds of the data type, in particular string and binary data types. In ASE there is no difference. To make Sybase IQ behave like ASE, this option would need to be set:

```
set option public.string_rtruncation = 'off';
```

ANSINULL

The default behavior in Sybase IQ is to handle null data values with the “is null” operator only. In Sybase IQ, the results of comparisons with NULL using ‘=’ or ‘!=’ are unknown. This includes results of comparisons implied by other operations such as CASE. In ASE data values that are null can be evaluated with “is null”, “=”, “!=”, and “<>”. To make Sybase IQ behave like ASE, this option would need to be set:

```
set option public.ansinull = 'off';
```

CHAINED

The default behavior in Sybase IQ is to have all operations issued contained within a single transaction until the application/user issued a commit. In ASE the behavior is to treat each command/batch as a separate transaction. To make Sybase IQ behave like ASE, this option would need to be set:

```
set option public.chained = 'on';
```

FLOAT_AS_DOUBLE

The default behavior in Sybase IQ is to interpret all occurrences of the keyword FLOAT as equivalent to the keyword DOUBLE within SQL statements. Since Adaptive Server Enterprise treats its own FLOAT values as DOUBLE, enabling this option makes Sybase IQ treat FLOAT values in the same way Enterprise treats FLOAT values. To make Sybase IQ behave like ASE, this option would need to be set:

```
set option public.float_as_double = 'off';
```

RUNTIME SQL

CALL

Currently, ASE only supports one method of executing stored procedures. This is done via the "execute" syntax. Sybase IQ also supports this syntax; however, another extension has been made if using the Watcom-SQL dialect. The "call" syntax will execute a stored procedure, as well.

```
Syntax 1  [ variable = ] CALL procedure-name ( [  expression] [ ,
...] )

Syntax 2  [ variable = ] CALL procedure-name ( [  parameter-name =
expression ] [ , ... ] )
```

FOR

Currently, ASE does not support loop processing. Sybase IQ does. The syntax of this new functionality is:

```
[  statement-label: ]

... FOR for-loop-name AS cursor-name

... CURSOR FOR statement

... [ {  FOR UPDATE | FOR READ ONLY } ]

... DO statement-list

... END FOR [  statement-label ]
```

IF

The "if" statement logic used in ASE is fully compatible with Sybase IQ; however, there are extensions in Sybase IQ that should be noted. The current ASE syntax for the "if" statement is:

```
IF expression

... statement

... [  ELSE [  IF expression ]  statement ] ...
```

The new syntax that is also supported in Sybase IQ is:

```
IF search-condition THEN statement-list
... [  ELSE IF search-condition THEN statement-list ] ...
... [  ELSE statement-list ]
... END IF
```

In order to use the extended syntax, though, the batch or procedure must be written using the Watcom-SQL dialect.

PRINT

The syntax of the print statement is no different between Sybase IQ and ASE. The one major difference, though, is where the output of the print statement goes, and this will depend on the type of client connection. TDS connections (Open Client and JDBC) will direct output to the client. ODBC connections send output to the database server window.

WHILE (T-SQL) vs. LOOP

The syntax of the while statement is no different between Sybase IQ and ASE. However, there are extensions to loop control that Sybase IQ has made. This is done with the "loop" operator in Watcom-SQL.

The current ASE and Sybase IQ T-SQL syntax is:

```
WHILE expression
... statement
```

The extensions that Sybase IQ has made to the "loop" syntax are:

```
[  statement-label: ]
... [  WHILE search-condition ]  LOOP
... statement-list
... END LOOP [  statement-label ]
```

Variable Persistence

In Sybase ASE variables are persistent for the batch. For instance, when a stored procedure exits, the variable and its contents are destroyed. In Sybase IQ, however, variables can follow the ASE persistence rule or be declared as static until the connection is terminated.

The "declare" syntax in ASE and Sybase IQ is used to declare a SQL variable within a compound statement (BEGIN... END). The "create variable" syntax that Sybase IQ also has is used to create a variable that is persistent for the connection or until the "drop variable" command is used.

Variable Names

In ASE, variable names must all be preceded with the '@' sign. In Sybase IQ, this is optional. There is no requirement in Sybase IQ for variable names other than they not be reserved words.

Where to Declare Variables

In Sybase IQ the variables must be declared immediately after the "begin" statement and before any other commands are issued. In ASE, variables can be declared anywhere in the compound statement.

Transaction Control

There are a few differences with respect to transactions and how the two engines treat them. While not technically a SQL difference, it does warrant being in this section due to the impact it has on the SQL being written.

A few transaction guidelines include:

- Use transaction control around logical units of work, even read only queries
- Commit before a read/write batch is started to ensure latest version of data is available
- Issue commit and rollback after batch completion to release all query resources
  – Rollback will free memory resources in use by previous operations
- For systems with high numbers of connected users, freeing memory resources can aid in query performance.

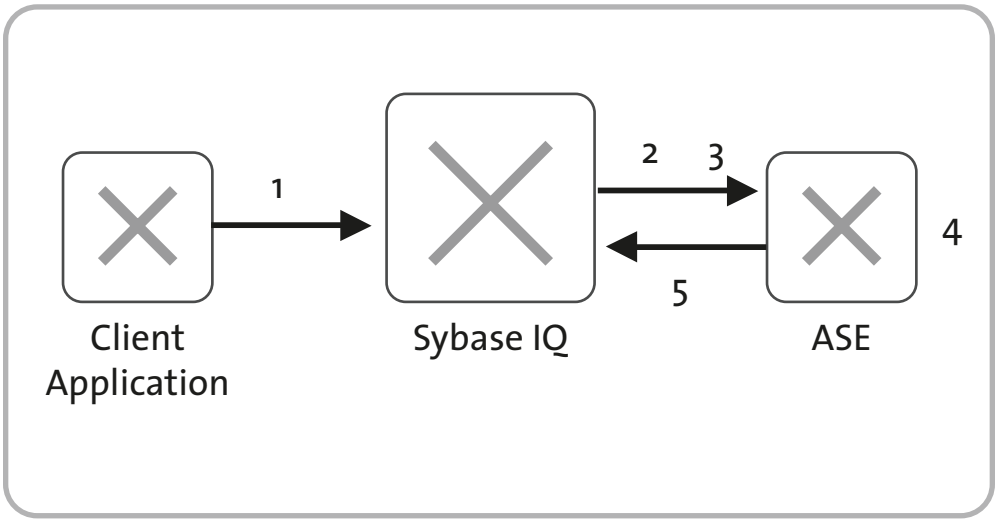
7 LOADING FROM ASE

Loading and extracting data is somewhat different than the ASE (the BCP command is not available). Sybase IQ provides has its own built-in utilities for loading and/or extracting data.

Sybase IQ lets you import or load data from flat files or directly from database tables. You can also enter specified values directly into the database. Export of data to other formats is available from the DBISQL utility and the Sybase IQ data extraction facility.

A Sybase IQ table is a logical table and does not contain data. All the information needed to resolve queries, including data, is contained in the Sybase IQ indexes. When you insert data into the columns in a Sybase IQ table, you are not actually adding data to the columns in the table, but rather to the column indexes. You "build" indexes by inserting data on a table-by-table basis.

LOADING DATA FROM ASE



HIGH LEVEL SEQUENCE STEPS

1. A Client application connects to the Sybase IQ server and issues the Insert From Location command
2. The Sybase IQ server opens an Open Client connection to ASE
3. The Sybase IQ server sends the Select statement from the insert from location command to the ASE Server
4. The ASE Server executes the SQL statement and returns the result set to the Sybase IQ server
5. The Sybase IQ server loads the result set

INSTALLATION AND TESTING

Step 1. Edit the sql.ini or interfaces file on the host where Sybase IQ resides and add an entry for the ASE Server where the source data resides. In this example the Sybase IQ database is named IQDB and the ASE Server is named ASESERVER.

```
File: %SYBASE%\ini\sql.ini

[ IQDB]

master=TCP,192.168.2.2,2638

query=TCP,192.168.2.2,2638

[ ASESERVER]

master=TCP,192.168.2.12,5000

query=TCP,192.168.2.12,5000
```

Step 2. The Sybase IQ Database needs a user ID and password for the ASE server where the source data resides. Here a login on the ASE Server is created and the user is added to the database where the source data resides.

```
D:\ASIQ\crt> isql -Usa -P -SASESERVER -iadd_ase_login.txt -e -
oadd_ase_login.out
```

NOTE: The userid created in Sybase IQ and ASE must be created exactly the same; case matters for both the ID and password! When debugging create a new login on both environments; once it is working you can attempt to use an existing login.

```
File: add_ase_login.txt

use pubs2

go

sp_dropuser iqloader

go

use master

go

sp_droplogin iqloader

go

-- sp_addlogin loginame, passwd, defdb

sp_addlogin iqloader, iqloader, pubs2

go

use pubs2

go

-- sp_adduser loginame, name_in_db, grpname

sp_adduser iqloader

go

grant select on authors to iqloader

go
```

Step 3. Verify that you have connectivity with ASE and that the login has the proper permissions. From the machine and environment where Sybase IQ is running, test the login that was created. It is not mandatory that the ASE user defaults to the database where the data is stored, but that user must be able to issue a "use database" and have permission to execute sql against the tables that you are loading from.

```
D:\ASIQ\crt> isql -Uiqloader -Piqloader -SASESERVER -
itest_ase_login.txt -e -otest_ase_login.out


File: test_ase_login.txt


select db_name()

go

set rowcount 1

go

select * from authors

go



File: test_ase_login.out


1> select db_name()

 ------------------------------

 pubs2


(1 row affected)

1> set rowcount 1

1> select * from authors

 au_id       au_lname                                au_fname

        phone        address

        city                 state country      postalcode

 ----------- ---------------------------------------- --------------
------

       ------------ ----------------------------------------

       -------------------- ----- ------------ ----------

 172-32-1176 White                                    Johnson

        408 496-7223 10932 Bigge Rd.

        Menlo Park           CA    USA          94025


(1 row affected)
```

Step 4. Create the Login for Sybase IQ Database/Server with the same user ID and password as the ASE server.

```
D:\ASIQ\crt>isql -Udba -Psql -SIQDB -e -iadd_login.txt -
oadd_login.out


File: add_login.txt


GRANT CONNECT TO iqloader IDENTIFIED BY iqloader
go


File: add_login.out


1> GRANT CONNECT TO iqloader IDENTIFIED BY iqloader
```

Step 5. Create the table in IQDB and grant permissions to iqloader.

```
File: crt_authors.txt


create table dbo.authors
(
    au_id        char(20)              not null,
    au_lname     varchar(40)           not null,
    au_fname     varchar(20)           not null,
    phone        char(12)              not null,
    address      varchar(40)           null    ,
    city         varchar(20)           null    ,
    state        char(2)               null    ,
    country      varchar(12)           null    ,
    postalcode   char(10)              null    ,
)
go
create hg index au_id_hg on authors(au_id)
go
create hg index phone_hg on authors(phone)
go
create hg index city_hg on authors(city)
go
create hg index postalcode_hg on authors(postalcode)
go
```

```
create lf index state_lf on authors(state)
go
create lf index country_lf on authors(country)
go
commit
go
grant all on authors to iqloader
go
```

Step 6. Load the data from ASE into IQ DB. Connect to the IQ DB server as the iqloader user and load data from the ASE server. The LOCATION 'ASESERVER.pubs2' portion of the command instructs the Sybase IQ server to open a connection through Open Client to a server named ASESERVER and to issue a "USE DATABASE pubs2" command and then execute the SQL statement contained in the curly brackets. The Sybase IQ server connects to the ASE Server as that user and sends the password; it then executes the SQL and the results are passed back to the IQDB server and loaded into the table—in this case "authors".

```
D:\ASIQ\crt>isql -Uiqloader -Piqloader -SIQDB -e -iload_authors.txt
-oload_authors.out


File: load_authors.txt
INSERT INTO dbo.authors
  (
    au_id,
    au_lname,
    au_fname,
    phone,
    address,
    city,
    state,
    country,
    postalcode
  )
 LOCATION 'ASESERVER.pubs2'
  {  select au_id,
       au_lname,
       au_fname,
       phone,
       address,
       city,
       state,
```

```
        country,
        postalcode
    from authors
  }
go
commit
go
select count(*) from dbo.authors
go
```

In most cases Sybase IQ will load the data as fast as ASE will return the data. This method of loading allows for data type conversions and schema changes. The SQL statement passed to ASE can have joins, conversions, case statements, and data type casting.

For efficient bulk loading of tables from flat files into Sybase IQ, use the load table statement. In addition, the Transact-SQL DML (insert, update, delete) commands will also execute in Sybase IQ.

8 CONNECTIVITY

WHAT AN ASE DBA SHOULD KNOW ABOUT SYBASE IQ DATABASE CONNECTIVITY

Sybase IQ is database centric; ASE is server centric. An ASE Server can support multiple user databases, a common and normal practice. The Sybase IQ architecture is quite different than ASE since a Sybase IQ server normally supports a single database. Memory resources like the Sybase IQ Main and Temp Cache are database level resources in Sybase IQ, not server level resources like in ASE. In ASE a connection is made to a server and a database is then accessed via "use database". When connecting to Sybase IQ a connection is made to a database, not a server.

Sybase IQ supports Open Client through a TDS listener built into the Sybase IQ server. Does Sybase IQ support your favorite Open Client based tool? If you use tools like SybPerl and Squish you won't be disappointed. ASE and its System Tables (master, sybsystemprocs, user database system tables) will not work correctly with Sybase IQ. The primary reason is the difference between Sybase IQ and ASE system tables.

ODBC is also considered a native communication protocol for Sybase IQ. Sybase IQ provides a wire level ODBC driver for Sybase IQ. An ODBC driver is available on all Sybase IQ server platforms and various client platforms. The Sybase IQ DBA does need to be concerned about configuring ODBC connectivity.

ODBC

Think of ODBC as being native connectivity to Sybase IQ (ODBC does not use Open Client). Most third party Business Intelligence tools use ODBC to access data in Sybase IQ—this includes multi-tier tools as well as two tier desktop tools. The great news here is that the Sybase IQ ODBC driver is written and maintained by Sybase.

If Sybase IQ or a Query Tool is running on UNIX the DBA will need to understand how to install and configure ODBC on UNIX. Sybase IQ ODBC drivers are available for the following platforms:

HP/UX 64 and 32bit

Solaris 64 and 32bit

AIX 64 and 32bit

Linux 32bit

Windows 32bit

HP Itanium

Sybase IQ is fully 64bit on UNIX; however, most third party UNIX based query tools are still 32bit and require a 32bit ODBC Driver (the drivers are located in $ASDIR/lib). ASE ODBC drivers do not support Sybase IQ since they are expecting the ASE system catalog tables.

OPEN CLIENT

Open Client configuration is slightly different in Sybase IQ than ASE, but it won't take much time to get up to speed on how Sybase IQ connectivity works. Sybase IQ has an interfaces file as might be expected in $SYBASE. The only surprise here is the Sybase IQ database name is used, not the Sybase IQ server name!

An entry in the interfaces file for the Sybase IQ server and mpxdb database looks like this on UNIX:

```
mpxdb
        master tcp ether linux_jb,2638
        query tcp ether linux_jb,2638
```

An entry in the sql.ini file for the DW Sybase IQ server and mpxdb database looks like this on WINDOWS:

```
[ mpxdb]
master=TCP,linux_jb,2638
query=TCP,linux_jb,2638
Test the Open Client Connectivity using isql:
isql -UDBA -PSQL -Smpxdb
1> exit

export DSQUERY=mpxdb

isql -UDBA -PSQL
1> select @@servername, db_name()
2> go
 @@servername    db_name(*)
------------------------------------
 DW                        mpxdb
```

ODBC

A Sybase IQ DBA also configures ODBC on UNIX. The odbc.ini (ODBC INI file) is used by default and is located in the user's home directory.

```
ls -l ~/.odbc.ini
```

```
-rw-r--r--    1 sybase   sybase        658 Dec  2 17:07
/home/sybase/.odbc.ini
```

There are several reasons to use the ODBC INI:

- Avoids hard coding connectivity info in scripts and applications
- Provides password security for utilities and SQL scripts

Just like dsedit and dscp tools for maintaining the interfaces file, Sybase IQ provides tools for maintaining the ODBC INI file (dbping and dbdsn). The file can also be updated using an editor in UNIX. Connectivity is easy once you see a couple of examples.

The Sybase IQ ODBC drivers are located in the $ASDIR/lib directory on UNIX and the $ASDIR/win32 directory for Windows.

On Windows the Data Source Administrator applet is used to create, modify and delete ODBC DSN's. On Windows, the ODBC Driver Manager is provided by Microsoft®.

On UNIX the ODBC INI file has similar functionality to the interfaces file as it contains connectivity information including user ID and password information. The security of this file is protected by standard UNIX file permissions.

```
cat ~/.odbc.ini
```

```
[ODBC Data Sources]
tpch=ASIQ Driver
utility_db=ASIQ Driver

[DW]
Driver=/home/sybase/asiq12/lib/dbodbc7_r.so.1
EngineName=DW
CommLinks=tcpip{host=linux_jb;Port=2638}
DatabaseName=mpxdb
UserID=DBA
Password=SQL
DBG=yes
LOG=/tmp/dw_odbc.log

[utility_db]
Driver=/home/sybase/asiq12/lib/dbodbc7_r.so.1
EngineName=utility_db
```

```
CommLinks=tcpip{host=linux_jb;Port=2638}

DatabaseName=utility_db

UserID=DBA

Password=SQL

DBG=yes

LOG=/tmp/utility_db_odbc.log
```

The dbdsn utility can be used to add, modify or view an ODBC Data Source Name (DSN). ODBC configuration is slightly different on UNIX than Windows. Windows has a concept of System and User DSN's—all this really means is where in the Windows Registry a DSN is located, a Windows service can only access System DSN's.

```
dbdsn -g DW

Adaptive Server Anywhere Data Source Utility Version 7.0.4.3490

DW:

UID=DBA;PWD=SQL;DBN=mpxdb;ENG=DW;CBSIZE=16000;DBG=yes;LOG=/tmp/dw_od
bc.log;LINKS=tcpip{host=linux_jb;Port=2638}
```

Once a DSN entry exists for a Sybase IQ server the dbping utility can be used to verify that the DSN entry is working correctly. dbping has various command line switches as shown below:

The first option checks for the server on the port:

```
dbping -c dsn=DW

Adaptive Server Anywhere Server Ping Utility Version 7.0.4.3490

Ping server successful.
```

If the server is visible on the network then the -d switch can be used to validate that the User ID and Password are correct:

```
dbping -d -c dsn=DW

Adaptive Server Anywhere Server Ping Utility Version 7.0.4.3490

Connected to ASA 7.0.4 (3490) server "DW" and database "mpxdb".

Ping database successful.
```

A word of caution—Sybase IQ connectivity has a built-in ability to locate Sybase IQ servers on the network. Sometimes a connection can be made to Sybase IQ when connectivity information in the DSN is incorrect. The connection is made by the connectivity layer locating a Sybase IQ server on the network using a UDP network broadcast. If the Sybase IQ server is located in a remote location the client connection may fail. Many companies with Wide Area Networks filter UDP traffic across WAN links, which means that UPD packets may get dropped. In this case a client broadcast will never reach the Sybase IQ server and the connection will fail. Adding UDP=OFF to the client ODNC DSN will resolve this issue. The above DSN would look like this:

```
[ DW]

Driver=/home/sybase/asiq12/lib/dbodbc7_r.so.1
```

```
EngineName=DW

CommLinks=tcpip{ host=linux_jb;Port=2638;UDP=OFF}

DatabaseName=mpxdb

UserID=DBA

Password=SQL

DBG=yes

LOG=/tmp/dw_odbc.log
```

When a Sybase IQ server starts up, by default it will broadcast the server name on the network using UDP. If another Sybase IQ server with the same name exists on the network, that server responds to the broadcast and your Sybase IQ server will not start. The following message will be reported in your Sybase IQ server's stderr file:

```
E. 01/07 07:32:24. A database server with that name has already
started
```

The way that Sybase IQ starts up can be modified using command line switches. See Sybase IQ documentation for additional details.

The ODBC Driver Manager also provides some simple debug logging. The DSN's above have the following lines for debugging:

```
DBG=yes

LOG=/tmp/dw_odbc.log
```

DBG can be turned off using "DBG=no". When DBG=yes then connection info is logged to the LOG file specified. Normally the tail command is used to monitor the log file during debugging connectivity issues. Here is a sample log:

```
tail -f /tmp/dw_odbc.log


Tue Dec 02 2003 16:57:09

16:57:09 CONN: Application information:

16:57:09
"IP=192.168.1.102;HOST=linux_jb.sybase.com;OS=Linux_jb.sybase.com
2.4.9-e.24 #1 Tue May 27
16:15:51;PID=0xc8e;VERSION=7.0.4.3490;API=DBLIB"

16:57:09 CONN: Attempting to connect using:

UID=DBA;PWD=***;DBN=mpxdb;ENG=DW;DSN=DW;DBG=YES;LOG=/tmp/dw_odbc.log
;LINKS=tcpip{ host=linux_jb;Port=2638}

16:57:09 Trying to start TCPIP link ...

16:57:09 My IP address is 192.168.1.102

16:57:09 My IP address is 127.0.0.1

16:57:09     TCPIP link started successfully

16:57:09 Trying 192.168.1.102:2638

16:57:09 Found database server dw on TCPIP link
```

```
16:57:09 Client connected
16:57:09 CONN: Connected to the server
16:57:09 [  446381609] CONN: Connected to database successfully
16:57:09 [  446381609] Client disconnected
```

```
dbdsn -h

Adaptive Server Anywhere Data Source Utility Version 7.0.4.3490
Usage: dbdsn [ switches] -l                       list data sources
   or  dbdsn [ switches] -d <dsn>                 delete a data source
   or  dbdsn [ switches] -g <dsn>                 get details of a
data source
   or  dbdsn [ switches] -w <dsn> [ details]      write data source
definition
   or  dbdsn             -cl                      list connection key-
words

Switches (use specified upper- or lower-case letter, as shown):
       -b            brief: print connect string
       -q            do not print banner
       -v            verbose: print connection values in tabular form
       -y            delete or overwrite data source without confirmation

Details:
       -c "keyword=value;..."
                     supply database connection parameters (see -cl
switch)
or individual parameters can be specified using:
       -e            encrypt all network packets
       -o <filename> write client messages to "filename"
                     (by default, errors are written to console)
       -p <size>     set maximum network packet size (>300, < 16000)
                     (default 1024)
       -r            disable multiple-record fetching
       -s <size>     memory for buffers in K (>= 5 and <= 1000)
                     (default 10)
       -tl <sec>     client liveness timeout in seconds
                     (default is server setting - default 120 seconds)
```

```
       -x <list>     comma separated list of network drivers to run
                     ShMem, TCPIP
       -z            display debugging information
       <server_name> connect to named server (first 40 characters
will be used)

Protocol options:

TCP/IP:  Host, Timeout, SendBufferSize, ReceiveBufferSize,
DoBroadcast
         ServerPort, MyIP, ClientPort, BroadcastListener, Broadcast

dbping -h

Adaptive Server Anywhere Server Ping Utility Version 7.0.4.3490
Usage: dbping [ switches]

       Switches (use specified lower-case letter, as shown):
       -c "keyword=value;..."
                     supply database connection parameters
       -d            make a database connection if the server is
found
       -l library    load specified ODBC driver or driver manager
library
       -m            use ODBC driver manager (otherwise use DBLib)
       -o <file>     log output messages to file
       -q            quiet: do not print messages

dbdsn -cl

Adaptive Server Anywhere Data Source Utility Version 7.0.4.3490
Connection keywords:
        <full name>          <short form>         <alternate name>
        Userid               UID
        Password             PWD
        DatabaseName         DBN
        DatabaseFile         DBF
        DatabaseSwitches     DBS
        ServerName           ENG                  EngineName
```

Unconditional	UNC
StartLine	START
ConnectionName	CON
AutoStop	ASTOP
Agent	AGENT
Integrated	INT
EncryptedPassword	ENP
CommBufferSize	CBSIZE
CommBufferSpace	CBSPACE
Debug	DBG
Encryption	ENC
LivenessTimeout	LTO
LogFile	LOG
DisableMultiRowFetch	DMRF
CommLinks	LINKS
AutoStart	ASTART
Charset	CS
ForceStart	FORCE
AppInfo	APP
PrefetchRows	PROWS
PrefetchBuffer	PBUF

ODBC-specific connection keywords:

Description

KeysInSQLStatistics

PreventNotCapable

TranslationName

TranslationDLL

TranslationOption

Driver

LazyAutocommit

DescribeCursor

IsolationLevel

Delphi

SaveFile

InitString

GetTypeInfoChar

9 TRANSACTION PROCESSING

Sybase IQ provides similar transactional capabilities as ASE. However, there are some important differences that will affect the default performance of migrated ASE applications. One of the most important differences is that Sybase IQ is by default set to UNCHAINED transaction mode, which is the exact opposite of the ASE default. Another major difference is the concept of Table-Level versioning. This is unique to Sybase IQ, and not available in ASE. This Table Level versioning causes a dramatic difference in availability of committed transactions.

The following paragraphs define Transactions and Versioning with Sybase IQ.

SYBASE IQ LOCK MODEL

Sybase IQ uses a radically different approach to locking than ASE. Sybase IQ does not have the same type of locking strategy in that Sybase IQ does not have row-level or page-level lock. Rather, Sybase IQ has an exclusive table-level write-lock. What this means is that only one user is permitted to modify a table at a time. While one user is holding a table-level lock, other users will receive an error message if trying to modify the same table.

SYBASE IQ VERSIONING

Sybase IQ uses snapshot versioning to allow many users to read from the database while it is being updated. A crucial aspect of this versioning method is its ability to isolate users from the effect of other users' transactions. You can think of snapshot versioning as you would a snapshot you take with a camera. When you photograph a snapshot of an object or scene, you get an image of it as it appears at a given moment in time. Likewise, when Sybase IQ takes a snapshot of an object in your database, it retains an image of that object at a given instant in time.

Unlike a camera, though, Sybase IQ does not need to make a copy of the entire object each time the image changes. Instead, it copies only the parts of the image (the database pages) that have changed. Database pages that have not changed are shared among all active versions in the database.

These version pages can build up over time and may take an excessive amount of unnecessary database space. Ensuring that the size of the version pages is under control will be one of the most important administrative tasks required for a Sybase IQ DBA.

The current size of the version pages held can be identified through sp_iqstatus. The outstanding connections which are preventing versions from being released can be identified through sp_iqtransaction. The offending connections will either have to commit or disconnect to free up the version pages and release the space.

10 SECURITY

USER IDS AND PERMISSIONS

Managing Users and Permissions in Sybase IQ is very similar to ASE, with a few exceptions. Many of the same management procedures from ASE also are used within Sybase IQ. However, there are some differences that include:

- Roles like sa, backup, etc. do not exist in Sybase IQ but are part of the Sybase IQ "dba authority"
- System tables cannot be modified by any login including the DBA.

ROLES

Unlike ASE, Sybase IQ only has 3 roles. These include:

- DBA Authority
- RESOURCE Authority
- USER.

When a database is created, a single usable user ID is created. This first user ID is DBA, and the password is initially set to SQL. The DBA user ID is automatically given DBA permissions (also called DBA authority) within the database. This level of permission enables the DBA user ID to carry out any activity in the database: create tables, change table structures, create new user IDs and revoke permissions from users, and so on.

RESOURCE authority is the permission required to create database objects, such as tables and indexes, views, and stored procedures. Resource authority may be granted only by the DBA to other users. The creator of a database object becomes the owner of that object. Ownership of a database object carries with it permissions to carry out actions on that object.

DBA USER ID IN CASE SENSITIVE DATABASES

User IDs and passwords are actually objects in the database. For this reason, if your database was created with the **CASE RESPECT** parameter, you must enter the user ID DBA and its password (SQL by default) in uppercase. For case insensitive databases (the default), you can enter this user ID and password in either uppercase or lowercase.

11 STORED PROCEDURES

SYBASE IQ ADMINISTRATIVE STORED PROCEDURES

Sybase IQ is designed to mimic ASE in terms of the System and Catalog stored procedures in almost all circumstances. The system tables in an ASE database exist in Sybase IQ through a set of system views and contain the same columns found in ASE. For example, the sysobjects view in Sybase IQ contains the same columns and values as that of ASE.

There are exceptions due to the differences in architecture between Sybase IQ and ASE. In these circumstances the stored procedures will call the function sp_tsql_feature_not_supported. Each of the following system or catalog stored procedures will invoke this special function:

```
sp_addalias
sp_addauditrecord
sp_addlanguage
sp_addremotelogin
sp_addsegment
sp_addserver
sp_addthreshold
sp_adddumpdevice
sp_auditdatabase
sp_auditoption
sp_auditlogin
sp_auditobject
sp_auditsproc
sp_bindmsg
sp_changedbowner
sp_checknames
sp_checkreswords
sp_clearstats
sp_commonkey
sp_configure
sp_cursorinfo
sp_dboption
sp_dbremap
sp_depends
sp_diskdefault
sp_displaylogin
sp_dropalias
sp_dropdevice
sp_dropkey
sp_droplanguage
sp_dropremotelogin
sp_dropserver
sp_dropthreshold
sp_estspace
sp_extendsegment
sp_foreignkey
sp_help
sp_helpconstraint
sp_helpdb
sp_helpdevice
sp_helpgroup
sp_helpindex
sp_helpjoins
sp_helpkey
sp_helplanguage
sp_helplog
sp_helpprotect
sp_helpsegment
sp_helpserver
sp_helpsort
sp_helpthreshold
sp_helpuser
sp_indsuspect
sp_lock
sp_locklogin
sp_logdevice
sp_modifylogin
sp_modifythreshold
sp_monitor
sp_placeobject
sp_procxmode
sp_primarykey
sp_recompile
sp_remap
sp_remoteoption
sp_rename
sp_renamedb
sp_reportstats
sp_role
sp_serveroption
sp_setlangalias
sp_spaceused
sp_syntax
sp_unbindefault
sp_unbindmsg
sp_unbindrule
sp_volchanged
sp_who
sp_column_privileges
sp_databases
sp_datatype_info
sp_server_info
sp_table_privileges
```

Because of the differences in architecture or because the functionality doesn't exist in Sybase IQ, most of these procedures simply do not make sense in Sybase IQ.

Some of these procedures would be very useful in the transition between the two systems. These procedures are detailed below.

Both Sybase IQ and ASE draw a distinction between System and Catalog Stored Procedures. The Sybase IQ Catalog Stored Procedures are owned by the "dbo" user and return result sets displaying database server, database, and connection properties. The sa_conn_info is an example of a frequently used procedure.

The Sybase IQ System Stored Procedures are owned by the "dbo" user and are used to perform system administrator tasks in the Sybase IQ Store. All of these procedures begin with "sp_iq*" and roughly correlate to the functionality found in the ASE procedures listed below.

For example, the Sybase IQ equivalent of the ASE "sp_help tablename" procedure is "sp_iqcolumn tablename", each displaying the columns and column properties for a given table.

ASE Procedure	**Sybase IQ Equivalent**
sp_addalias	User alias is not supported in Sybase IQ.
sp_addlanguage	Alternate date formats can be defined on a per-user basis through "set option user.date_format=", p.124
sp_addremotelogin	CREATE EXTERNLOGIN statement, p.429
sp_addserver	CREATE SERVER statement, p.453
sp_bindmsg	N/A
sp_changedbowner	N/A
sp_checknames	N/A
sp_checkreswords	N/A
sp_column_privileges	Detailed in the SYSCOLPERM system table
sp_commonkey	N/A
sp_configure	SET OPTION statement, p.578
sp_cursorinfo	LOG_CURSOR_OPERATIONS option, p.145
sp_databases	N/A
sp_datatype_info	Detailed in the SYSUSERTYPE and SYSTYPEMAP system tables
sp_dboption	N/A - See the SET OPTION statement
sp_depends	Detailed in the SYSTABLE and SYSCOLUMN system tables
sp_displaylogin	sp_iqlistlockedusers, sp_iqlistexpiredpasswordssp_iqlistpasswordexpirations
sp_dropalias	N/A
sp_dropkey	N/A
sp_droplanguage	N/A
sp_dropremotelogin	REVOKE CONNECT statement, p. 556

sp_dropserver	DROP SERVER statement, p.485
sp_estspace	sp_iqestspace, sp_iqestdbspaces
sp_foreignkey	FOREIGN KEY clause in CREATE TABLE, p.405
sp_help	sp_iqtable, sp_iqcolumn, sp_iqview
sp_helpconstraint	sp_iqconstraint
sp_helpdb	N/A
sp_helpgroup	Detailed in the SYSGROUP system table
sp_helpindex	sp_iqindex, sp_iqindex_alt
sp_helpjoins	sp_iqestjoin
sp_helpkey	Detailed in the SYSTABLE system table
sp_helplanguage	N/A
sp_helpprotect	Viewed through Sybase Central
sp_helpserver	sp_servercaps
sp_helpsort	N/A
sp_helpuser	Detailed in the SYSUSERPERM system table
sp_indsuspect	sp_iqcheckdb, sp_iqdbstatistics
sp_lock	sp_iqlocks
sp_locklogin	sp_iqlocklogin, sp_iqlistlockedusers
sp_modifylogin	sp_iqmodifylogin, sp_iqmodifyadmin, sp_iqpassword
sp_primarykey	PRIMARY KEY clause in CREATE TABLE, p.405
sp_procxmode	sp_tsql_environment
sp_recompile	N/A
sp_remoteoption	sp_login_environment
sp_rename	ALTER TABLE…RENAME statement, p.378
sp_renamedb	N/A
sp_role	N/A
sp_server_info	sp_iqstatus
sp_serveroption	sp_iqcheckoptions
sp_setlangalias	N/A
sp_spaceused	sp_iqspaceused, sp_iqspaceinfo, sp_iqindexsize, sp_iqjoinindexsize, sp_iqtablesize
sp_syntax	HELP statement (dbisql only), p.508
sp_table_privileges	Detailed in the SYSTABLEPERM system table
sp_unbindefault	N/A

sp_unbindmsg	N/A
sp_unbindrule	N/A
sp_volchanged	N/A
sp_who	sp_iqconnection, sa_conn_info

Note: The page references in this table refer to the corresponding page in the Sybase IQ Reference Manual.

Sybase IQ supports auditing through the audit features provided in ASA. The ASE system stored procedures sp_addauditrecord, sp_auditdatabase, sp_auditlogin, sp_auditobject, sp_auditoption, sp_auditsproc have equivalent functionality implemented through ASA (see AUDITING OPTION in the "ASA Database Administration Guide").

Since Sybase IQ does not require database devices in the same format as ASE, the following System Stored Procedures are not applicable:

```
sp_adddumpdevice, sp_addsegment, sp_addthreshold, sp_dbremap,
sp_diskdefault, sp_dropdevice, sp_dropthreshold, sp_extendsegment,
sp_helpdevice, sp_helplog, sp_helpsegment, sp_helpthreshold,
sp_logdevice, sp_modifythreshold, sp_placeobject, sp_remap
```

Instead, the following System Stored Procedures have been provided within Sybase IQ to help in managing dbspaces:

- sp_iqspaceinfo will show the dbspace where individual columns or indexes are stored.
- sp_iqdbspace will show detailed information about each dbspace.
- sp_iqalterdbspace will designate a dbspace as read-only or read-write.
- sp_iqestdbspaces will estimate the number and size of dbspaces needed for a given total index size.

Sybase IQ gathers system metrics differently than ASE. The sp_monitor, sp_reportstats and sp_clearstats system stored procedures have been replaced by the Sybase IQ specific sp_iqcontext, sp_iqtransaction, and sa_conn_properties procedures.

In general, any of these system administration functions can be performed from the Sybase Central administration tool provided with Sybase IQ.

12 SYBASE IQ DATABASE OPTIONS

This section addresses setting Sybase IQ database options and discusses the differences between options in ASE and Sybase IQ. Like ASE, Sybase IQ provides database options to customize and control database behavior. Sybase IQ options come with a default setting and may be adjusted to meet the needs of an application. Unlike ASE, however, there are over 100 database options as compared to less than 10 for ASE. To the ASE sa the lengthy list of Sybase IQ database options might appear to be similar to ASE server configuration options, and in a sense that is correct. In Sybase IQ there is only one database running on a Sybase IQ server, so any option that is set may affect the entire Sybase IQ server. The similarity ends there, as the type of options are decidedly different from ASE. Plus Sybase IQ provides more granularity, allowing both users and dbas to control the scope and duration of some database options.

While most options are available to all users, a few are restricted for use by a user with DBA authority. Most options are dynamic but a few will require a reboot of the Sybase IQ server to be enabled. DBA authority is required to set a permanent server-wide option or user specific option.

Chapter 5 of the Sybase IQ Reference Manual addresses database options. Options are listed in tabular form, then explained in detail. You will note that a number of the options listed in the table of options are NOT documented in detail in that section of the chapter. These are undocumented options for Sybase IQ engineering and technical support use and are listed for informational purposes only.

There is a relatively short list of important database options that need to be understood and set for proper database performance (addressed later in this section). DBAs (and users) will need to understand how to properly adjust database options for the appropriate scope and duration. The SET OPTION command is used for this purpose. The full syntax is:

```
SET [ TEMPORARY] [ user_id. | PUBLIC.] option_name = [ option_value]
```

OPTION SCOPE

There are three levels of scope for options: Public, user and temporary. Temporary takes precedence over user_id (user) and Public settings. User level takes precedence over Public. Public is a server-wide setting (DBA use only) and will affect all users. If a user_id or Public is not specified in the Set Option command it will only affect the user issuing the command.

OPTION DURATION

There are three duration levels: Public, Temporary and Temporary until next Sybase IQ server restart. SET PUBLIC.option_name becomes a permanent server-wide option. SET TEMPORARY option_name is effective immediately and persists until changed or a user's session ends. SET TEMPORARY PUBLIC.option_name (DBA use only) is also effective immediately but persists until the Sybase IQ server is restarted, when it reverts to the permanent setting.

Examples

1. After installation you will find the option for generating Query Plans is ON by default and produces a query plan in the Sybase IQ Message File every time a user runs a query. The DBA can turn this option off permanently for all users using:

```
SET OPTION Public.Query_Plan = 'Off';
```

2. John is a user in the database and needs to see query plans. He can override the Public setting by executing this command in a session before running his query:

```
SET TEMPORARY OPTION Query_Plan = 'On';
```

John will generate a query plan every time he executes a command until he either turns the option off or logs off the server. The next time he logs in, he will not generate a query plan since the Public setting for that option applies.

3. Judy is a user who always wants to have a query plan generated. The DBA sets this option permanently for Judy using:

```
SET OPTION Judy.Query_Plan = 'On';
```

Judy can temporarily turn off query plans anytime during a session with

```
SET TEMPORARY OPTION QUERY_PLAN = 'Off';
```

but the next time Judy logs in she will again generate query plans since that option is permanently on for her user_id.

DATABASE OPTIONS AFFECTING PERFORMANCE AND SERVER BEHAVIOR

This section addresses database options that DBAs may consider changing from default.

`QUERY_PLAN - default ON.  Recommendation OFF`

Query plans will be generated for all queries in the Sybase IQ Message File for all commands. Unless query plans are needed on a regular basis this option should be turned off permanently.

`FORCE_NO_SCROLL_CURSORS - default OFF.  Recommendation ON`

By default, all query results are buffered (in the Temp Cache) to permit users to scroll forward and back through all result rows. Since this behavior has an impact on memory, we recommend disabling this option by setting the option ON. If specific users require this functionality the option may be set OFF for those user_ids.

`QUERY_TEMP_SPACE_LIMIT - default 2000 (MB).  Recommendation 0`

This option sets the limit for the maximum amount of memory in the Temporary Cache for a user. Any SQL statement that exceeds that limit will fail with an error message. Recommend setting this option to 0 (zero) to remove the limit for Temporary memory.

`APPEND_LOAD - default OFF.  Consider setting to ON`

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

`LOAD_MEMORY_MB - default 0.  Allowable values 0 to 2000`

Load_Memory_MB controls the amount of heap memory used during a Load Table operation. The default of zero sets no limit on the amount of memory used. For applications running on 32 bit systems or on any machine with a limited amount of memory, this option may need to be set to a value to restrict the amount of memory needed. If a Load Table operation uses too much memory there will be an 'out of virtual memory' error, which kills the Load Table. Set the Load_Memory_MB somewhere between 100 and 500 MB to limit the heap memory. This may require some trial and error. Be aware that loading a table with varchar columns greater than 255 bytes may also require setting this option as the heap memory requirements may be significant.

`ROW_COUNT - default 0.  Note behavior differences from ASE`

This option only controls the number of rows returned from a SELECT statement and has no impact on UPDATE or DELETE commands. The behavior in Sybase IQ is significantly different from the ASE option ROWCOUNT which affects all query operations.

13 RESERVED WORD DIFFERENCES

Sybase IQ and ASE have different universes of reserved words which either must be avoided in certain SQL statements or, in the case of Sybase IQ, must be enclosed in double quotes. While the two server products share a common list of reserved words, the list below represents reserved words in Sybase IQ that do NOT exist in the universe of reserved words in ASE (current to ASE version 12.5).

There is no utility in Sybase IQ to check for reserved words as there is within ASE using the system procedure "sp_checkreswords."

Sybase IQ Reserved Words Not Restricted by ASE

backup, bigint, binary, bit, bottom

call, cast, char, character, comment, connect, contains, cross

date, dbspace, dec, decimal, disable, dynamic

elseif, enable, encrypted, endif, exception, existing, externlogin

first, float, forward, full

identified, inner, insensitive, instead, int, integer, integrated, iq

left, login, long

match, membership, message, mode

natural, no, notify, numeric

options, others, outer

passthrough, publication

real, reference, release, remote, rename, resource, restore, restrict, right

savepoint, schedule, scroll, session, share, smallint, sqlcode, sqlstate, start, stop, subtrans,

subtransaction, synchronize, syntax_error

then, time, timestamp, tinyint, top

unknown, unsigned

validate, varbinary, varchar, variable

within

(References: Chapter 8, ASE Reference Manual—Transact-SQL Reserved Words;

Chapter 6, Sybase IQ Reference Manual—List of Reserved Words)

14 USING ASE AND SYBASE IQ TOGETHER

IMPLEMENTING CIS

ASE and Sybase IQ can be configured together so that some tables (or parts of tables) can be stored in ASE while others are stored in a Sybase IQ database. Historical data, for example, might be stored in a Sybase IQ table with more recent data stored in a similar table in ASE. Data from both sources can be accessed from a connection on the ASE server.

Sybase IQ data tables are accessed from an ASE database using Proxy Tables implemented from ASE's Component Integration Services (CIS). This section addresses how to implement CIS in ASE to access tables of data tables in a Sybase IQ database.

PRELIMINARIES

Before configuring CIS in ASE to access Sybase IQ there are some technical considerations with regard to the data types used in the Sybase IQ database. You must be careful to use data types in Sybase IQ that can be translated by ASE and Open Client (see Table Design section in Chapter 3 of this paper). The data types in a Sybase IQ that are referenced by a proxy table cannot contain bit, unsigned integer or bigint (where int or bigint data values exceed 2.14 billion), varchar types larger than 255, and large binary objects. The alternative is to create new tables in Sybase IQ (or add columns in existing Sybase IQ tables) using data types that can be handled by CIS and proxy tables.

LOGINS AND PERMISSIONS FOR PROXY TABLES

In order for users to access objects from the Sybase IQ database server, logins and permissions will need to be configured. By default, CIS passes the ASE user login and password to the remote server for access. If the user and name and password do not match, or if the user does not have SELECT permissions for the table in the Sybase IQ database, the action will fail. The DBA will need to either set up appropriate logins and permission on the Sybase IQ server or use the ASE system stored procedure sp_addexternlogin to map an ASE login to a valid login on the Sybase IQ server. Below is the syntax for the stored procedure:

```
sp_addexternlogin <servername>, <loginname>, <external_login-
name>, <external_password>
```

ENABLING CIS IN ASE TO ACCESS SYBASE IQ

Follow these steps in ASE to create a Proxy Table that references a table in a Sybase IQ database:

1. **Add the Database Name to the Sybase IQ Configuration File.** On the Sybase IQ server, add the database name for the Sybase IQ database to the configuration file with a second "-n" parameter. This parameter is required for Open Client access to your Sybase IQ server.

2. **Add the Sybase IQ Server/Database to Interfaces.** On the machine running ASE, add the Sybase IQ server database name to the Interfaces file. This name should match the database name for the Sybase IQ server in para 1 above. On the ASE machine test the Open Client connection using isql for connectivity to the Sybase IQ server.

3. **Enable CIS on the ASE Server.** Check the ASE server configuration file to be sure that CIS is enabled for the server. If not, modify the configuration file or execute the command: sp_configure 'Enable CIS', 'On'. In newer releases of ASE the option is 'On' by default.

4. **Add the Sybase IQ Server to sysservers.** Add the Sybase IQ server to ASE sysservers tables using either the Sybase Central wizard or the command line sp_addserver, IQServer_name, ASIQ. Be sure you specify the server type as ASIQ (applies to ASE 12.5 or later).

5. **Create the Proxy Table on ASE.** There are two methods to create a proxy table in ASE.

    a. **Create Proxy Table.** This single line command creates an ASE proxy table using all the columns defined in the remote table on the Sybase IQ server. You cannot use Sybase Central to create a proxy table using this method.

    The create proxy_table command reads the table definition in the Sybase IQ server to create the ASE proxy table. Some data type changes on the proxy table could occur—unsigned integer and bigint data types for Sybase IQ tables will be defined as ASE integer type in the proxy table. (Beware that selecting any data from a Sybase IQ table outside the range of an integer data type will fail with a conversion error on ASE.) The syntax for Create Proxy Table is:

```
Create proxy_table ASEproxyTabName at 'IQserver.owner.IQTabName'
```

For this example, you are creating an ASE proxy table named ASEproxyTabName for a table named IQTabNam owned by owner on the Sybase IQ server named IQserver.

b. **Create Existing Table.** This command allows more flexibility by allowing you to control which columns to define in the proxy table. The syntax is very similar to a Create Table command:

```
Create Existing Table  ASEproxyTabName (
column_name  datatype     null|not null,
…
)on 'default'
external table
at 'IQserver..owner.IQTabName'
```

The proxy table definition may be a subset of columns contained in the Sybase IQ table. Data types defined in the proxy table must be compatible for implicit data type conversion between the two servers (e.g. char to varchar).

QUERY PERFORMANCE USING CIS

Running queries directly on proxy tables in ASE should perform as if the query was run directly in Sybase IQ. In addition, queries involving UNION statements involving both ASE and proxy tables of Sybase IQ data typically perform well.

You should probably avoid performing queries involving joins on large tables involving a proxy table and an ASE base table. Such queries use CIS to perform joins rather than the higher performing ASE query engine. Consider testing joins across servers before implementing for production.

SYBASE®

Sybase Incorporated
Worldwide Headquarters
One Sybase Drive
Dublin CA, 94568 USA
T 1.800.8.SYBASE
www.sybase.com