SYBASE®

An **SAP**® Company

Installation and Configuration Guide

Sybase IQ 15.3

Sun Solaris

Contents

Preparing for Installation

Read this section before you install Sybase® IQ.

About Sybase IQ

Sybase® IQ is a high-performance decision-support server designed specifically for data warehousing.

Sybase IQ is part of the Adaptive Server® product family that includes Adaptive Server Enterprise and SQL Anywhere. Component Integration Services within Sybase IQ provide direct access to relational and nonrelational databases on mainframe, UNIX, or Windows servers.

Supported Server Platforms

Sybase IQ is compatible with these platforms and operating systems.

- Sun Solaris 10 (SPARC T) for 64-bit systems – minimum patch level 14 January 2008
- Sun Solaris 10 (x64) for 64-bit systems – minimum patch level 11 January 2008. Patch 119961-04 also required

Supported Client Platforms

Sybase IQ 15 includes a network client for all supported server platforms.

All Sybase IQ UNIX and Linux installations include ODBC drivers needed for the client and InteractiveSQL. Sybase IQ Network Client for Windows includes 32-bit ODBC Windows drivers, which allow you to connect to Sybase IQ from third-party 64-bit Windows applications, and 64-bit Windows applications written in C++.

Licensing Requirements

All product editions except the Evaluation Edition require a license. Optional features are sold and licensed separately.

Whether you install a licensed version of Sybase IQ or the Evaluation Edition, you have 30-day access to all features and options. To convert the Evaluation Edition, or to use an option beyond the 30-day evaluation period, you must purchase and install an appropriate Sybase Software Asset Management (SySAM) license.

The e-mail message or Web key you receive when you purchase a Sybase product provides specific product and licensing information. Before installing and configuring any product edition, know which features you are licensed to use.

Installation Media

Sybase IQ includes installation media for server and client installations.

Table 1. Sybase IQ Installation Media.

Media	Description
Sybase IQ Server Suite DVD	Installs Sybase IQ server components and other support tools.
Sybase IQ Client Suite CD	Installs the components required for client connections to a Sybase IQ server.

Planning Your Installation

Sybase IQ installation process depends on whether you are installing Sybase IQ for the first time, upgrading from a previous version, or migrating your databases to a new hardware platform.

Note: Sybase Control Center is a Web-based tool for managing and monitoring Sybase servers, including Sybase IQ single-node and multiplex servers. If you plan to use SCC to manage your Sybase IQ servers, see the *Sybase Control Center Installation Guide* during the planning phase of your deployment for information on hardware requirements.

Installing Sybase IQ For the First Time

Follow these steps to install Sybase IQ for the first time.

- Read the release bulletin for the latest information. See the special installation instructions section for any installation instructions not included in this document.
- Complete all preinstallation tasks. Apply any required operating system patches, check to see that there is sufficient disk space and memory, and update any necessary configuration changes.
- Determine the license model you want to use. SySAM supports served and unserved license models. See the *Sybase Software Asset Management Users Guide* to determine which model is best suited for your environment.
- Use the product and licensing information you received from Sybase to log in to the Sybase Product Download Center and generate a license key for your software.
- Install Sybase IQ server components, activate your license, then start and test your server.
- Install and configure Sybase IQ client components.

Upgrading From an Earlier Version

The upgrade process depends on the version of Sybase IQ currently installed on your system.

Sybase IQ 15

- Choose an appropriate installation directory. You cannot install Sybase IQ in the same parent directory as an earlier version. Uninstall the version currently installed on your system, or install Sybase IQ in a different parent directory.
- Upgrade your client tools. Although you can use the current version of the client tools to connect to an earlier 15.x server, some options, tools, and objects may not be available.

Sybase IQ 12.6/12.7

- Upgrade Sybase IQ, if necessary. All database migration paths assume that you are migrating from Sybase IQ 12.6 ESD #11 or Sybase 12.7 ESD #5. If you are migrating from a version earlier than 12.6, you must upgrade your software before you migrate your database.
- Review the migration procedures and plan your migration. Sybase provides utilities that you can use to migrate your 12.x data to Sybase IQ 15.3. Understanding how these utilities work can help you plan your migration more effectively.
- Follow the work flows for the type of databases you want to migrate. There are separate migration paths for simplex and multiplex databases.

Changing Hardware Platforms

To upgrade from an earlier version and move your Sybase IQ databases to a different hardware platform, perform the conversion before you upgrade your data.

- For information about migrating a database from one platform to another with the same endian structure, see *Testing Installed Products > Converting to a new hardware platform*.
- After you change your hardware platform, follow the appropriate work flow in the "Database Upgrades" section.

Note:

- If you created your Sybase IQ database on a Linux 32-bit version prior to Sybase IQ 12.6 ESD #2, first install IQ 12.6 ESD #2 for Linux 32-bit and create a new data backup before converting to another platform.
- Sybase IQ 12.6 ESD #2 and later versions support migration between Windows and Linux.

Planning for Distributed Query Processing or High Availability

Distributed query processing can benefit from an optional high-speed interconnect. Use scalable interconnect technology to connect multiplex nodes.

Sybase recommends:

- A high-speed network interconnect providing a local network that connects all multiplex nodes. Use an interconnect bandwidth of 1Gb or higher or the highest bandwidth, lowest latency interconnect available.
- Two physically separate network interfaces for each multiplex node:
    - A public network for multiplex interconnection traffic and client traffic.
    - A private network for multiplex interconnect traffic only, excluding external client traffic. Currently, multiplex interconnects support only the TCP/IP standard.

    These two networks improve security, fault-tolerance, and performance.
- A switch that enables high-speed communication between nodes.
- Network cards that reside on different fabrics so that the multiplex survives network failure. Separate the public and private networks physically.
- Private interconnect fabrics that contain only links to machines participating in the multiplex. Private interconnect for all multiplex nodes should connect to the same switch, which connects to no other public switches or routers.
- Redundant network interface cards added to private or public networks if desired. The private and public connection information allows for multiple IP addresses on each.

Preinstallation Tasks

Perform the following tasks before you install Sybase IQ. You may need to refer to your operating system documentation for specific commands.

Increase the Swap Space

The recommended minimum swap space to run Sybase IQ is at least 1GB. Certain operations may cause the memory used by Sybase IQ to grow dynamically. Changing the way Sybase IQ uses buffers can dramatically change the way it uses memory.

Depending on the load, swap requirements may exceed space allotted. Insufficient swap space may result in the system supporting fewer users, and large processes that cannot acquire sufficient swap space may be prematurely killed by the operating system.

See your operating system documentation for information about extending swap space.

Sybase IQ May Require a New License Server

Sybase IQ requires SySAM license server version v11.6.1 or later, which is distributed on the Sybase IQ Server Suite installation media. Using an older SySAM server can cause the Sybase IQ server to become unresponsive. Install the license server before you install Sybase IQ.

To check the version of your current SySAM license server, open a console or command prompt, change to $SYBASE/SYSAM-2_0/bin, and enter:

```
./lmutil lmver lmgrd
```

If you currently use a license server to host SySAM licenses for other Sybase products:

* Uninstall your current license server
* Install the license server distributed on the Sybase IQ Server Suite DVD
* Rehost your licenses

Rehost your SySAM licenses after you install the new license server. See the *Sybase Software Asset Management 2 Users Guide*.

Manage Shared Memory

In some cases, increasing the shared memory segment can lead to better network performance.

Sybase IQ uses shared memory for two purposes:

* When the client and Sybase IQ are on the same system, shared memory is the default communication mechanism, not TCP/IP.
* When you specify the server command-line option **-iqwmem**, shared memory is used to provide wired memory.

If the **-iqwmem** option is used, you need to increase the maximum shared memory segment size parameter in the /etc/system configuration file. Sybase recommends that you specify a configuration for the largest amount of wired memory you expect your system to support.

To create a shared segment to support a maximum physical memory of 3350MB for your system, for example, add the following line to the /etc/system file:

```
set shmsys:shminfo_shmmax=0xffffffff
```

Never set **shmmax** to a value greater than 7/8 of physical memory.

Local connections through shared memory are not supported for Sun Solaris x64 systems. Use standard network connectivity by adding the parameters **-host** *<hostname>* and **-port** *<portnumber>* to the client connection string.

See the Sybase IQ System Administration Guide for more information on using the **-host** and **-port** parameters.

Set the File Descriptor Limits

To use the Solaris debugger, add a file descriptor for each thread in addition to the file descriptors for each dbspace.

Although the Solaris thread limit is not a problem, using the Solaris debugger on IQ requires a file descriptor for each thread in addition to the file descriptors for each dbspace. You need to adjust the number of file descriptors by means of the C shell **limit** command and by including `set rlim_fd_max=4096` in `/etc/system`.

Verify Network Functionality

Make sure your network devices can communicate before you install Sybase IQ.

Sybase IQ uses networking software whenever the client and server components are installed on different systems. Use the **ping** command to and from another computer to verify that the server where you plan to install Sybase IQ can use TCP/IP to communicate with other devices on your network.

Licensing Your Software

When you purchase a Sybase product, you receive an e-mail message or a Web key that tells you how to log in to the Sybase Product Download Center (SPDC) and dynamically generate license keys for Sybase IQ.

The e-mail message or Web key also provides specific product and licensing information. Use this information when you generate your licenses.

Available Licenses

Available licenses are determined by the product edition and options you purchase.

Enterprise Edition (EE)

Sybase IQ Enterprise Edition requires a CPU license, and supports several licensing options.

Table 2. Enterprise Edition.

Orderable License	Actual License	Required	Optional	Description
CPU	IQ_CORE	•		License restricts the number of cores (not CPUs) which an IQ server can use.
LOMO	IQ_LOB		•	Allows you to create and use CLOB or LOB columns. The number of LOB licenses must match number of cores.
				IQ_LOB license is included for customers who purchase an IQ_UDA license. An IQ_LOB license is also available as a separate purchase option.
				This license is required for 12.6 or 12.7 customers who already have a LOB license (IQ_LOB).

Ordera- ble Li- cense	Actual Li- cense	Re- quired	Option- al	Description
VLDB MO	IQ_VLDBMG MT		•	Allows you to create multiple table spaces and additional IQ dbspaces beyond the defaults al- lowed.
				You are allowed one IQ_SYSTEM_TEMP dbspace, one IQ_SYSTEM_MAIN dbspace, one IQ_SHARED_TEMP, and one additional user defined main store dbspace.
				To create any additional IQ dbspace requires the IQ_VLDBMGMT license. This license re- stricts the ability to create table partitions. An IQ_VLDBMGMT is required whenever an IQ server creates or starts with two or more IQ user dbspaces.
				Each VLDB license allows for 1 TB of storage. You will need to purchase one license for each additional TB of storage in the main store.
ASO	IQ_SECURITY		•	Allows you to use column encryption, ECC Kerberos and FIPS. The number of ASO licen- ses must match cores
				This license is required for 12.6 or 12.7 cus- tomers who already have encryption (IQ_ENC).
MPXO	IQ_MPXNODE		•	Allows you to start secondary multiplex nodes (readers/writers). The number of nodes must always be less than cores, as each multiplex server must have at least one core.
				This license is required for 12.6 or 12.7 cus- tomers who are already running multiplex. This is a new license requirement.
IDBA-PSO	IQ_UDF		•	Allows you to define and run high-performance scalar and aggregate user-defined functions.
				This license is only available with approved third-party libraries.
UDA	IQ_UDA		•	Allows you to use the IQ Text Search function- ality, which lets Sybase IQ search unstructured and semi-structured data. An IQ_UDA also in- cludes an IQ_LOB license.
				This license applies to IQ **TEXT** indexes.

Note: SySAM verifies optional feature licensing at various times. See *Troubleshooting SySAM > License Checkout and Verification* for a table that describes the events that determine when SySAM checks out different licenses.

Small Business Edition (SE)

Sybase IQ Small Business Edition requires a CPU license.

Table 3. Small Business Edition.

Orderable License	Actual License	Re-quired	Optional	Description
CPU	IQ_CORE	•		Licenses the number of sockets (not CPUs) which an IQ server can use.

Note:

- Server checks out licenses equal to the number of sockets. One license allows one socket, and you can have up to four licenses. The total number of cores should never exceed 4 (MAXCP).
- Total concurrent IP connections <= 25 (MC)
- Total main store <= 250GB (MS)

Single Application Edition (SA)

Sybase IQ Single Application Edition requires a CPU license.

Table 4. Single Application Edition.

Orderable License	Actual License	Required	Optional	Description
CPU	IQ_CORE	•		Licenses the number of cores (not CPUs) which an IQ server can use.

Note: Other limits set by the SA license

- Total cores per server <= 64 (MAXCP)
- Total number of user tables <= 25 (MT)
- Total number of columns <= 100 (MV)
- Total number of mpx nodes <= 1 (MSN)

Evaluation Edition

The Evaluation Edition does not require a license. The Evaluation Edition is an unlicensed server, which provides full access to all features and options available in the Enterprise

Edition. To run an unlicensed database beyond a 30-day evaluation period, however, you must purchase and install an appropriate license.

Before You Generate Your License

Follow these steps before you generate your license.

Read the SySAM Documentation

Review the SySAM documentation for specific information about SySAM licensing, license models, and procedures.

- *Sybase Software Asset Management (SySAM) 2 Users Guide* introduces asset management concepts and provides instructions for establishing and administering SySAM 2 licenses.
- *SySAM 2 Quick Start Guide* tells you how to get your SySAM-enabled Sybase product up and running quickly.
- *FLEXnet Licensing End User Guide* explains FLEXnet Licensing for administrators and end users and describes how to use the tools which are part of the standard FLEXnet Licensing distribution kit from Sybase.

Decide On the License Model You Want To Use

A license model refers to the way you set up your license files for your licensed programs. License files can reside on a central, networked license server in a served license model or on the machine running the licensed program in an unserved license model.

Decide on the license model you want to use before you generate your license file. For information about served and unserved licenses, and which model is best suited for your environment. See the *Sybase Software Asset Management Users Guide*.

Get Your Host ID

If you plan to install a new license server or use an existing license server, you need the license server host ID.

FLEXnet Licensing uses different machine identifications for different machine architectures. If you have a copy of the FLEXnet licensing utilities installed, you can use the `lmhostid` utility to print the exact host ID that FLEXnet Licensing expects to use on any given machine. `lmutil` is included in your product distribution, or you can download a copy from *FLEXnet Publisher download page*

You can also use a system-level command to find the host ID. For platform-specific commands and examples, see *Hostids for FLEXnet Licensing-Supported Platforms*. For additional information, see the FAQs at the Sybase Product Download Center (SPDC).

Note: Some platforms derive the host ID from the network adapter address. If your machine has multiple network adapters, the command you use to get the host ID may return one host ID

---

for each network adapter. If this occurs, choose only one from the list of host IDs. Sybase recommends that you use the value associated with the primary wired Ethernet adapter. Do not use values associated with internal loopback adapters.

Determine the Host Name

You must specify the host name of the license server for served licenses. Although the host name is optional for unserved license models, providing the host name for all license types can simplify future license administration tasks.

Choose a Port Number

If you plan to install a license server or use an existing license server, you need to know the TCP/IP port number that the license server will use to serve your licenses.

If you do not specify a port number, the license server uses the first available port in the range of 27000 to 27009. If you are using three-server redundancy, specify a port number outside of this range.

Generating a SySAM License

To generate a SySAM license, login to the Sybase Product Download Center. Refer to the Web key or email you received from Sybase for information about your SPDC account.

Prerequisites

- Before you generate your license, you need the host ID and host name of the server where you plan to install Sybase IQ.
- Served licenses require a license server. If you do not plan to use an existing license server, you must install one before you install Sybase IQ.

Task

1. Start your browser and log into the *Sybase Product Download Center (SPDC)*.
2. On the Product List page, choose Sybase IQ.
3. On the Product Information page, choose the product version appropriate for your platform.
4. On the License Information page, choose the appropriate location, then read and accept the license agreement, if necessary.
5. On the Product Download page, click the License Keys link.
6. On the License Information page, choose the appropriate license(s), scroll to the bottom of the page, and click Select to Generate.
7. On the Generate Licenses page:

- • Choose Served or Unserved license, then click Next.
- • Specify the number of licenses to generate, then click Next.
- • Identify the host machine where you intend to install Sybase IQ. You must identify the host ID and number of licenses. The host name is optional. Click Generate.

8. Click Download License File and save your license to a temporary location on the local file system. You must save the license file with a `.lic` extension.

Installing a New License Server

The Sybase IQ Server Suite installation media includes SySAM Network License Server utilities you can use to install a new license server.

Prerequisites

- • Stop any servers currently running in the installation directory.

Task

1. Insert the installation media into the drive.

   If the Sybase IQ installer does not start automatically, mount the drive, then enter:

   `<DVD_mount_directory>/setup.bin`

2. On the Welcome screen, click Next.
3. Choose an installation directory, then click Next.
4. Choose Custom as the installation type you want to perform, click Next.
5. Unselect all installation options except SySAM License Utilities.
6. Select License Server, click Next.
7. Review the installation summary, then click Install.

Starting a License Server

Copy your license to the license directory, and start the license server from a console or command line.

1. From a console or command line, copy the license file you saved from SPDC to the `$SYBASE/SYSAM-2_0/licenses` directory on the network license server.
2. Change to `$SYBASE/SYSAM-2_0/bin`, then refresh or restart the license server:

Options	Description
sysam start	Start the license server.
	Perform this step if the license server is not running. After you perform this step, repeat step 2 to verify that the server is running.
sysam reread	Make the license server read the new license file.
	Perform this step if the license server is already started.

3. To verify that the license server is running, enter sysam status.

Note: The SySAM License Server Installer installs both IPv4-only and IPv4/IPv6 dual-stack license server binaries during installation and configures use of IPv4-only binaries.

Use the **sysam configure** [IPv6 | IPv4] to configure the appropriate version of the license server. For example, **sysam configure IPv6** will configure use of the IPv4/IPv6 dual stack binaries. For more information, see *SySAM Users Guide*.

Server Installations

Read this section for step-by-step server installation instructions.

Server Components

Server components are grouped by feature. Items marked as default are installed in typical installations. Optional items can be installed in custom installations.

Table 5. Sybase IQ Server Components.

Feature	Components	Default	Notes
Sybase IQ	Sybase IQ Server	•	Installs server software, client tools, ODBC drivers, and Web application development tools.
	Sybase IQ Client	•	
	Interactive SQL	•	Additional Connectivity Language Modules support character sets for Eastern Europe, Turkey, and Greece.
	Sybase IQ Web Drivers	•	
	Sybase IQ ODBC Driver	•	
Sybase Control Center	Sybase Control Center Agent	•	Sybase Control Center is a Web-based administrative console for Sybase products. Sybase Control Center Agent is required to manage Sybase IQ server tasks.
Sybase Central	Sybase IQ Plug-in	•	Sybase Central is an administrative console for Sybase products. The Sybase IQ Plug-in is required to administer tasks for Sybase IQ.
jConnect	jConnect 6.0 for JDBC	•	jConnect is a Type 4 JDBC driver, which is entirely Java-based: • jConnect 6.0 is JDBC 3.0 compliant • jConnect 7.0 is JDBC 4.0 compliant
	jConnect 7.0 for JDBC	•	
SySAM	SySAM License Server		Installs FLEXnet Licensing utilities to support your licensing model.
	SySAM License Utilities	•	

Plug-in Compatibility

Sybase Central is an administration console for Sybase products. Not all versions of Sybase Central are compatible with Sybase IQ 15.

The Sybase IQ server 15 plug-in for Sybase Central is not compatible with 12.x servers. You can use the Sybase 15 plug-in with 15.x servers; you cannot use it to connect to 12.x servers.

Table 6. Sybase IQ Plug-in Compatibility.

IQ plug-in	Sybase IQ	Sybase Central toolkit	Agent	JRE
15.3	Compatible only with 15.3 servers.	6.0.0.4787	15.3 agent required	JRE – 6.0 required
15.x	15.x refers to Sybase IQ versions 15.0, 15.1, 15.2. You cannot use the 15.x plug-in to connect to 12.x servers.	6.0 or later required	15.x agent required	JRE – 6.0 required
12.7	12.5 and later fully supported. Earlier IQ versions minimally supported.	4.3 or later required	12.7 agent required	1.4.2 required
12.6	12.6 or 12.5 fully supported. Earlier IQ versions minimally supported.	4.3 or later required	12.6 agent required	1.4.2 required
12.5	12.5 or earlier	3.2 required	12.5 agent required	1.2.2 required (1.3.1 on Linux)

What a Server Installation Does

A typical installation installs the server and other components into a parent directory on the host machine.

Directory Variables
Sybase IQ uses environment variables that point to different installation directories.

Table 7. Sybase IQ Directory Variables.

Variable	Description
$SYBASE	Points to the parent installation directory. This variable is set by the installer.
$IQDIR15	Points to the Sybase IQ installation directory. This variable is set with a shell command after you run the installer.

Parent Installation Directory
The $SYBASE directory contains the files, scripts, and other objects required to support the server.

Table 8. Sybase IQ Parent Installation Directory.

Directory	Contents
charsets	Character sets available for use with Adaptive Server.
collate	Collation sequences available for use with Adaptive Server.
config	Various configuration files.
DBISQL	Interactive SQL.
IQ-15_3	Subdirectories, scripts, and property files required to run Sybase IQ server.
jConnect-6_0	JDBC 3.0 compliant driver for Sybase IQ.
jConnect-7_0	JDBC 4.0 compliant driver for Sybase IQ.
jre	Java runtime environment.
jutils-2_0	Version 2.0 Java-based utilities, including Ribo, a Tabular Data Streams utility.
jutils-3_0	Version 3.0 Java-based utilities, including Ribo, a Tabular Data Streams utility.
locales	Language modules for system messages and date/time formats.
log	Installation log files.
OCS-15_0	Open Client tools and utilities.
SCC-3_2	Sybase Control Center startup and configuration files.
shared	Objects and libraries shared by other components, including Sybase Central and the Java Runtime Environment (JRE).

Directory	Contents
Sybase_Install_Registry	Server registry file that holds information related to the installed products. Do not modify.
sybuninstall	Subdirectories for scripts that uninstall IQ and SySAM.
SYSAM-2_0	SySAM licenses and utilities. This directory contains three subdirectories: `bin`, `licenses`, and `log`.
ThirdPartyLegal	Third party license terms and agreements.

Product Installation Directory
$IQDIR15 is a $SYBASE subdirectory that contains the files, scripts, and other objects required to run the server.

Table 9. Product Installation Directory.

Directory	Contents
bin*<platform>*	Utilities you use to start and manage your server. *<platform>* value is 64 or 32.
demo	Sample data and scripts for the demo database.
install	Installation history file.
java	Different JAR files.
lib*<platform>*	Various library files. *<platform>* value is 64 or 32 depending on your installation.
logfiles	Sybase IQ writes log files to this directory.
lsunload	Local Store Unload utility.
res	Character encoding resource files.
samples	Sample SQL Anywhere tools and scripts.
scripts	Sample scripts and stored procedures.
sdk	Subdirectories for various language utilities.
shared	Objects and libraries shared by other components, including Sybase Central and the Java Runtime Environment (JRE).
Sysam	Local SySAM configuration information.
tix	Tcl/Tk extension library.

Directory	Contents
unload	Various unload scripts, files, and utilities.

Installing Server Software

Server installation media supports interactive, console, and unattended (silent) installs.

Setting Up the Sybase Account

Creating a `sybase` account provides consistent ownership and privileges for all installation, configuration, and upgrade tasks.

The `sybase` user must have permission privileges from the top (or root) of the disk partition or operating system directory down to the specific physical device or operating system file. If you have already installed other Sybase software, the `sybase` user may already exist. If so, verify that you can log in to the machine using this account.

If there is no `sybase` account on your system, see your operating system documentation for instructions about creating new user accounts.

Running the Installer

The Sybase IQ installation media supports GUI, console, and unattended (silent) installs.

Installing in GUI Mode

Use this option to set up your software in a graphic user interface (GUI) environment. The GUI installer is wizard-driven and supports all product editions and installation types.

1. Insert the installation media into the DVD/CD drive.

   If the Sybase IQ installer does not start automatically, mount the drive, then enter:
   ```
   <DVD_mount_directory>/setup.bin
   ```
2. Follow the instructions on the screen.
3. Unmount, if necessary, and remove the installation media from the drive.

Installing from the Command Line

Choose a command-line installation if you prefer a non-windowing interface or for developing custom installation scripts. Installing components in console mode is similar to installing in GUI mode, except that you run the installer from the command line, and enter text to select the installation options.

1. Insert the installation media into the DVD/CD drive.

   If the installation program launches automatically, click Cancel to stop the installer.
2. From a console, change to the setup directory.

The setup directory is the directory that contains `setup.bin`, `installer.properties`, and other installation files.

3. Enter:

```
setup.bin -i console
```

4. Follow the instructions on the screen.

Installing in Unattended (Silent) Mode

Unattended or silent installations are typically used for updating multiple systems. Except for the absence of the GUI screens, all actions of InstallAnywhere are the same, and the result of an installation in silent mode is exactly the same as one done in GUI mode with the same responses.

Prerequisites

A response file is a text file that contains installation options. You can override or edit the default response file, `installer.properties`, for use in subsequent installations. To override the default installation options, use the **-f** argument to point to a custom response file.

Task

1. Insert the installation media into the DVD/CD drive.

If the installation program launches automatically, click Cancel to stop the installer.

2. From a console, change to the setup directory.

The setup directory contains `setup.bin`, `installer.properties`, and other installation files.

3. Enter:

```
setup.bin -f <responseFileName> -i silent
-DAGREE_TO_SYBASE_LICENSE=true
-DSYBASE_PRODUCT_LICENSE_TYPE=license
```

Consider these arguments when you use a response file:

- **-f <responseFileName>**

  Optional argument that includes the absolute path and file name of the response file. If you omit this argument, the installer defaults to `installer.properties` or other `.properties` file in the same installation directory.

- **-i silent**

  Switch that starts the silent installer.

- **-DAGREE_TO_SYBASE_LICENSE=true**

  Argument that indicates you accept Sybase end-user license agreement.

- **-DSYBASE_PRODUCT_LICENSE_TYPE=license**

Required to install a licensed version of the software.

Creating a response file
Start the installer from the command line and use the -r switch as an argument to create a response file. The installer starts a GUI installation and records your responses in a response file. The response file is a text file that you can edit to change any responses prior to using it in any subsequent installations.

1. From the command line, enter:

```
setup.bin -r <responseFileName>
```

Where `<responseFileName>` includes the absolute path and file name of the response file.

2. Run the installer in GUI mode to record your actions in the response file.

Use an editor to change any responses for subsequent installations, if necessary.

Setting Environmental Variables

Set the environmental variables required to run the server. The shell scripts that set these variables also set the variables needed by Open Client.
Do one of the following:

- For the tcsh or C (csh) shell, enter:
```
source $SYBASE/IQ-15_3/IQ-15_3.csh
```
- For the Korn (ksh), Bash, or Bourne (sh) shells, enter
```
$SYBASE/IQ-15_3/IQ-15_3.sh
```

Solutions to Common Installation Problems

Check this section for solutions to common installation issues.

Table 10. Troubleshooting installation problems

Problem	Possible Cause and Solution
Install fails	Check the installer log: `$SYBASE/log/IQSuite.log` The install log can help you trace the install progress and troubleshoot failed installations. The installer also writes individual logs for components included as part of the install to this directory.
The installer reports: `gzip: "Not Found"`	The installer cannot find the path to the `gzip` file. `gzip` must be installed and referenced in the path for the installer to succeed. The installer expects to find `gzip` in the `/usr/bin` directory. On some machines, `gzip` is located in the `/usr/contrib/bin/` directory. If this is the case, update the path and run the installer.

Problem	Possible Cause and Solution

$SYBASE variable not defined	Use these shell commands to set the $SYBASE environment variable to the path of the Sybase IQ parent installation directory.
	
	For the tcsh or C shell (csh):
	`setenv SYBASE <directory name>`
	
	For the Korn (ksh), Bash, or Bourne (sh) shells:
	`SYBASE=<directory name>;`
	`export SYBASE`

Uninstalling UNIX Servers

To uninstall Sybase IQ on UNIX-like operating systems, run the uninstaller from a terminal.

1. Open a terminal, change to:
   ```
   $SYBASE/sybuninstall/IQSuite
   ```
2. Enter:
   ```
   ./uninstall
   ```
3. Follow the instructions on the screen.

Testing Installed Products

This section tells you how to start Sybase IQ and use the demo database to test your installation.

Demo Database

Many of the examples throughout the documentation use the Sybase IQ demo database, (iqdemo.db), as a data source. The demo database is installed as part of the Sybase IQ Server Suite and resides on the server. Client access is provided by tools installed as part of the Sybase IQ Client Suite.

Creating and Starting the Demo Database

To use the demo database to test your installation, create and load the demo database.

1. To create the demo database, open a terminal.
2. Change to :
   ```
   $IQDIR15/demo
   ```
3. Enter:
   ```
   mkiqdemo.sh
   ```
4. To start the demo database, enter:
   ```
   start_iq @iqdemo.cfg iqdemo.db
   ```

Table Names

Sybase IQ 15 demo database table names are different than those in the 12.*x* version of the software.

This table includes current and former iqdemo table names. If you are upgrading from an earlier version of Sybase IQ, use this table to update the table names or permissions in your scripts or applications.

Table 11. Demo database table names

12.7 name	15.x name	15.x owner
contact	Contacts	GROUPO
customer	Customers	GROUPO
department	Departments	GROUPO
employee	Employees	GROUPO

12.7 name	15.x name	15.x owner
fin_code	FinancialCodes	GROUPO
fin_data	FinancialData	GROUPO
product	Products	GROUPO
sales_order_items	SalesOrderItems	GROUPO
sales_order	SalesOrders	GROUPO
emp1	emp1	DBA
iq_dummy	iq_dummy	DBA
sale	sale	DBA
SYSOPTIONDEFAULTS		DBO

Note: To create the older asiqdemo version of the demo database, open a command window, change to $IQDIR15/demo, and run this command: **mkiqdemo.sh -old**. For additional options, use the **mkiqdemo.sh -help** command.

Running Sybase IQ Server

Use the startup utility, **start_iq**, to start your server. **start_iq** is a command line utility that runs on all platforms and ensures that all required parameters are set correctly.

Starting a Server or Database

To use **start_iq** to start a server or database, change to a directory where you have write privileges, and use the following command format:

```
start_iq [ server-options ] [ database-file
[ database-options ], ...]
```

- *server-options* include the database server name and other options that control the behavior of the server, for all databases that are running on that server.
- *database-file* is the file name of the catalog store. You can omit this option, or enter one or more database file names on the command line. Each of these databases is loaded and available for applications. If the starting directory contains the database file, you do not need to specify the path; otherwise, you must specify the path. You need not specify the .db file extension.
- *database-options* are options that you can specify for each database file you start, that control certain aspects of its behavior.

Default Startup Parameters

start_iq uses a default configuration file ($IQDIR15/scripts/default.cfg) to set the default start-up parameters.

Table 12. Parameters set by start_iq

Parameter	Value	Description
–c	48MB	Catalog store cache size.
–gc	20	Checkpoint interval.
–gd	all	Allows all users to start the database by connecting.
–gl	all	Allows all users to load or unload tables.
–gm	10	Default number of connections.
–gp	4096	Catalog store page size.
–ti	4400	Client timeout set to 72 hours. Prevents users with long queries from being logged off over a long weekend.

For a complete list of startup parameters, see *Utility Guide > start_iq Database Server Startup Utility > start_iq Server Options*.

Note: If SQL Anywhere is installed on the same subnet as Sybase IQ, the server must have a unique name. Both SQL Anywhere and Sybase IQ servers default to the port 2638.

Use a new port number for each server. Set each new port number in the $IQDIR15/scripts/default.cfg file. Change the following line in each IQ database configuration file (for example, $IQDIR15/demo/iqdemo.cfg)to update the port number:

```
-x tcpip{port=2638}
```

Configuration Files

To override the default start-up parameters, store your options in a configuration file.

To start a server with a configuration file, use this syntax:

```
start_iq @configuration_filename.cfg dbname.db
```

This command starts the database and sets parameters named in the (optional) configuration (.cfg) file.

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

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

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

A configuration file for the demo database (`iqdemo.cfg`) is installed in the `$IQDIR15/demo` directory as an example.

Default Server Directories
The directory where the server is started becomes the default directory for all server files created by Sybase IQ.

Note: If you run the **start_iq** utility from `$IQDIR15/bin64`, the script changes the directory to "`./..`" to avoid creating database files in the `/bin` directory.

Startup and Server Logs
Sybase IQ writes start-up and server logs in the `$IQDIR15/logfiles` directory:

• Startup information is saved in the `servername.nnnn.stderr` file.
• Server status is logged in the `servername.nnnn.srvlog` file.

The *nnnn* variable in the file name indicates the number of times the server has been started. For example, `localhost.0004.srvloglocalhost_iqdemo.0006.srvlog`

Starting Sybase IQ with Sybase Central
If you have Sybase Central, you can use the Start Database Server wizard to start the database instead of **start_iq**.

Note: Always use the default colors of the Common Desktop Environment when you run Sybase Central and **dbisql**. Running these products under Open Windows or changing the default colors may cause display problems.

Stopping Servers

You may need to stop and restart a server to install a new version of Sybase IQ, change server server-wide database options, or close an operating system session.

DBA is the default permission level required to stop a server, but you can use the **-gk** startup option to change the default to ALL or NONE. Setting the permission level to ALL lets all users stop the server; setting the permission level to NONE means that no one, not even the DBA, can stop the server. In a production environment, Sybase strongly recommends that only the DBA be allowed to stop the database server.

Do not shut down a server while it is still connected to one or more clients. If you try this, you get a warning that any uncommitted transactions will be lost. Disconnect or close all the clients and try again.

Server Shutdown Commands
Several commands shutdown the database server.

Command	Description
stop_iq	Shutdowns a server completely without regard for users, connections or load process status. Syntax: `stop_iq [ options ]` See the *Utility Guide > Stopping the Database Server* for complete options and syntax.
dbstop	Stops a server or database. Additional options let you stop a server, even if there are active connections. Syntax: `dbstop [ options ] server-name` To use **dbstop** to stop a server, you must specify a **server-name**, as well as any connection parameters you specified when you started the server. See the *Utility Guide > Stopping the Database Server* for complete options and syntax.
STOP ENGINE	**STOP ENGINE** is a SQL statement that stops a database server. Syntax: `STOP ENGINE engine-name [ UNCONDITIONALLY ]` The UNCONDITIONALLY argument stops a database server, even if there are client connections to the server. See *Reference: Statements and Options > SQL Statements > STOP ENGINE Statement* for complete options and syntax.

Server Shutdown Methods
There are several methods to shutdown or stop a server.

To stop the server from..	Do this...
Command line	Run the **stop_iq** or **dbstop** commands.
Sybase Central	Right-click the server name and choose **Stop Server**.
Interactive SQL	Run the **stop engine** command to to stop a named database server.

Managing Processes
The **stop_iq –agent** command stops the IQ Agent.

Stopping Servers in Cron or At Jobs

To use **stop_iq** in a **cron** or **at** job, specify the full path name to the **stop_iq** executable and the appropriate **–stop** option:

```
stop_iq -stop one
```

Setting **–stop one** shuts down a single server, when exactly one running server was started by the user ID that starts the **cron** or **at** job. This prevents accidentally shutting down the wrong server if several are running.

```
stop_iq -stop all
```

You can also use **stop_iq** to shut down IQ agent, for example:

```
stop_iq -agent -stop all
```

Setting **–stop all** shuts down all servers that were started by the user ID that starts the **cron** or **at** job.

Note: For other administrative issues, see *System Administration Guide: Volume 1 > Sybase IQ Startup > Database Server Shutdown.*

Starting and Stopping the Sybase Control Center Agent

(Optional) Launch or shut down the Sybase Control Center agent included with your Sybase IQ server.

Sybase Control Center is a Web-based tool for managing and monitoring Sybase servers, including Sybase IQ single-node and multiplex servers. One SCC server can manage and monitor all the Sybase IQ servers in your enterprise. SCC performs management tasks using the SCC agent, which is installed with each Sybase IQ server.

For instructions on installing SCC, see the *Sybase Control Center Installation Guide*. For instructions on setting up the SCC agent to run as a service, starting and stopping the SCC agent service, and running the SCC agent from the command line, see the Sybase Control Center online help. Sybase Control Center documentation (including the installation guide and online help) is available at *http://sybooks.sybase.com/nav/summary.do?prod=10680.*

1. Use the **scc** script to start the SCC agent.

   `$SYBASE/SCC-3_2/bin/scc.sh`

2. Connect to the Sybase Control Center server. In your Web browser, enter: `https://`
   `scc-hostname:8283/scc.`

3. Log in.

   When logging in to a newly installed Sybase Control Center for which secure authentication has not been configured, use the sccadmin account (with no password, by default). For more information, see the *Sybase Control Center Installation Guide*.

4. In SCC, register and authenticate your Sybase IQ server's SCC agent.

5. (Optional) Stop the SCC agent. To shut down from the `scc-console>` prompt, enter `shutdown`.

   If you use SCC to manage your Sybase IQ server, leave the SCC agent running whenever Sybase IQ is running. The easiest way to do this is to run the SCC agent as a service that restarts automatically.

Running Interactive SQL

Interactive SQL is a utility that ships with Sybase IQ that lets you execute SQL statements, build scripts, and display database data.

Prerequisites

Assume that `iqdemo` is running on a local host named *localhost-xp*.

Task

1. Open a console, and run **dbisql** from the `$SYBASE/DBISQL/bin` directory.

   Starting Interactive SQL opens a dialog that lets you define the connection parameters to your server or database. If you need connection help, click the arrow in the top-right corner to display a Connect Assistant.

2. On the Identification tab, type DBA in the User box, and sql in the Password box.

   DBA is not case-sensitive; sql is case-sensitive, and must appear in all lowercase letters.

3. On the Database tab, choose a server from the Server name box. If there are no servers listed, click Find to search for running servers.

   Server names appear in this format *<hostname>_dbname*. If the host name is *localhost-xp* and you are running the demo `iqdemo` database, the server name is *localhost-xp_iqdemo*.

4. Choose `iqdemo` as the database. If there is only one database running, you can leave this field blank.

5. What you do next depends on the server location:

   • If the server is installed on the same machine, choose an appropriate database file.
   • For a networked server, click the Network tab, click the TCP/IP box, then specify the Host name and Port number

6. Click OK to open Interactive SQL.

7. In the SQL Statements window, enter: `SELECT * FROM employees`

8. To execute the query, click the right-facing triangle on the toolbar.

Note:

- Sybase Central and Interactive SQL no longer support jConnect. Sybase Central and Interactive SQL use the SQL Anywhere JDBC Driver. jConnect is provided with Sybase IQ 15 for customer application backward compatibility purposes.
- As a general rule, the jConnect driver cannot use ODBC data sources. However, Sybase Central and Interactive SQL are special cases. When you use the jConnect driver in either of them, you can specify an ODBC data source to establish a connection.

Post-Installation Tasks

Sybase recommends that you perform these tasks after you install and test Sybase IQ.

Verify Your License
To verify your license, start the server, and run the **sp_iqlmconfig** stored procedure in Interactive SQL (**dbisql**). To verify the license from a custom license installation, check the $SYBASE/SYSAM-2_0/log/SYBASE.log file. You can use iqdemo.db to perform this task.

A line that begins with "Checked out license..." indicates a successful license configuration. If you see a "Sysam: FLEXnet Licensing error:" message, check with your Sybase representative to resolve the issue.

For additional information about license issues, see the *Troubleshooting SySAM* section and the *SySAM 2 Users Guide*.

Change the Default Connection Parameters
Sybase recommends that you change the default connection parameters:

Table 13. Default Connection Parameters

Parameter	Default Value
User ID	DBA (case insensitive)
Password	sql (case sensitive)
Port Number	2638

Upgrade Your Database
To upgrade a database from an earlier edition of Sybase IQ, see the "Database Upgrades" section. Depending upon your current version of the IQ database the method of upgrading will differ. There are separate upgrade procedures for simplex and multiplex databases

Client Installations

Installing client components lets you connect a client to a network server.

Client Components

Client components are grouped by feature. Items marked as default column are installed in a typical installation. Optional items can be installed in a custom installation.

Table 14. Sybase IQ Client Components.

Feature	Components	De-fault	Notes
Sybase IQ	Sybase IQ Client	•	Installs client software, Web application development tools, and ODBC drivers.
	Interactive SQL	•	
	Sybase IQ Web Drivers	•	
	Sybase IQ ODBC Driver	•	
Sybase Central	Sybase Central	•	Sybase Central is an administrative console for Sybase products. The Sybase IQ Plug-in is required to administer tasks for Sybase IQ.
	Sybase Central Plug–in	•	
jConnect	jConnect 6.0 for JDBC	•	jConnect is a Type 4 JDBC driver, which is entirely Java-based:
	jConnect 7.0 for JDBC	•	• jConnect 6.0 is JDBC 3.0 compliant • jConnect 7.0 is JDBC 4.0 compliant

What a Client Installation Does

A typical client installation installs the network client components into a parent directory on a client machine.

Directory Variables
Sybase IQ uses environment variables to point to different installation directories.

Table 15. Directory Variables.

Variable	Contents
$SYBASE	Points to the parent installation directory. This variable is set by the installer.
$IQDIR15	Points to the Sybase IQ installation directory.

Parent Installation Directory
The $SYBASE directory contains the files, scripts, and other objects required to support the client.

Table 16. Parent Installation Directory.

Directory	Contents
DBISQL	Interactive SQL.
IQ-15_3	Sybase IQ client components.
jConnect-6_0	JDBC 3.0 compliant driver for Sybase IQ.
jConnect-7_0	JDBC 4.0 compliant driver for Sybase IQ.
shared	Shared software components.
Sybase_Install_Registry	Client component registry file.
jre	Java Runtime Environment directory.
jutils-2_0	Version 2.0 Java-based utilities, including Ribo, a Tabular Data Streams utility.
jutils-3_0	Version 3.0 Java-based utilities, including Ribo, a Tabular Data Streams utility.
log	Client log file directory.
ThirdPartyLegal	Third-party license agreements.
sybuninstall	Subdirectories for scripts that uninstall IQ and SySAM.

Product Installation Directory
$IQDIR15 is a $SYBASE subdirectory that contains the files, scripts, and other objects required to run the client.

Table 17. Product Installation Directory.

Directory	Contents
bin*<platform>*	Utilities you use to start and manage your server. *<platform>* value is 64 or 32, depending on your installation.
install	Installation history file.
java	Different JAR files.
lib<platform>	Various library files. *<platform>* value is 64 or 32, depending on your installation.
res	Character encoding resource files.
samples	Sample SQL Anywhere tools and scripts.
sdk	Subdirectories for various language utilities.
shared	Subdirectories for various shared components.
tix	Tcl/Tk extension library.

Installing Client Software

The client installation CD supports interactive, console, and unattended installations. Perform all pre-installation tasks before you install the client.
- Stop any servers currently running in the installation directory.
- Mount the drive, if necessary.

Running the Installer

The Sybase IQ installation media supports GUI, console, and unattended (silent) installs.

Installing in GUI Mode

Use this option to set up your software in a graphic user interface (GUI) environment. The GUI installer is wizard-driven and supports all product editions and installation types.

1. Insert the installation media into the DVD/CD drive.
   If the Sybase IQ installer does not start automatically, mount the drive, then enter:
   ```
   <DVD_mount_directory>/setup.bin
   ```
2. Follow the instructions on the screen.
3. Unmount, if necessary, and remove the installation media from the drive.

Installing from the Command Line

Choose a command-line installation if you prefer a non-windowing interface or for developing custom installation scripts. Installing components in console mode is similar to

---

installing in GUI mode, except that you run the installer from the command line, and enter text to select the installation options.

1. Insert the installation media into the DVD/CD drive.

   If the installation program launches automatically, click Cancel to stop the installer.

2. From a console, change to the setup directory.

   The setup directory is the directory that contains `setup.bin`, `installer.properties`, and other installation files.

3. Enter:
   ```
   setup.bin -i console
   ```

4. Follow the instructions on the screen.

Installing in Unattended (Silent) Mode

Unattended or silent installations are typically used for updating multiple systems. Except for the absence of the GUI screens, all actions of InstallAnywhere are the same, and the result of an installation in silent mode is exactly the same as one done in GUI mode with the same responses.

Prerequisites

A response file is a text file that contains installation options. You can override or edit the default response file, `installer.properties`, for use in subsequent installations. To override the default installation options, use the **-f** argument to point to a custom response file.

Task

1. Insert the installation media into the DVD/CD drive.

   If the installation program launches automatically, click Cancel to stop the installer.

2. From a console, change to the setup directory.

   The setup directory contains `setup.bin`, `installer.properties`, and other installation files.

3. Enter:
   ```
   setup.bin -f <responseFileName> -i silent
   -DAGREE_TO_SYBASE_LICENSE=true
   -DSYBASE_PRODUCT_LICENSE_TYPE=license
   ```

   Consider these arguments when you use a response file:

   • **-f <responseFileName>**

Optional argument that includes the absolute path and file name of the response file. If you omit this argument, the installer defaults to `installer.properties` or other `.properties` file in the same installation directory.

- **-i silent**

  Switch that starts the silent installer.
- **-DAGREE_TO_SYBASE_LICENSE=true**

  Argument that indicates you accept Sybase end-user license agreement.
- **-DSYBASE_PRODUCT_LICENSE_TYPE=license**

  Required to install a licensed version of the software.

Creating a response file
Start the installer from the command line and use the -r switch as an argument to create a response file. The installer starts a GUI installation and records your responses in a response file. The response file is a text file that you can edit to change any responses prior to using it in any subsequent installations.

1. From the command line, enter:

   ```
   setup.bin -r <responseFileName>
   ```

   Where `<responseFileName>` includes the absolute path and file name of the response file.
2. Run the installer in GUI mode to record your actions in the response file.

   Use an editor to change any responses for subsequent installations, if necessary.

Setting Environment Variables

Open a terminal and set the environment variables required to run Sybase IQ. The shell scripts that set these variables also set the variables needed by Open Client.
Do one of the following:

- For the tcsh or C (csh) shell, enter:

  ```
  source $SYBASE/IQ-15_3/IQ-15_3.csh
  ```
- For the Korn (ksh), Bash, or Bourne (sh) shells, enter

  ```
  $SYBASE/IQ-15_3/IQ-15_3.sh
  ```

Uninstalling Unix Clients

To uninstall Sybase IQ on UNIX-like operating systems, run the uninstaller from a terminal.

1. Open a terminal, change to

   ```
   $SYBASE/sybuninstall/IQClientSuite
   ```
2. Enter:

   ```
   ./uninstall
   ```

    3. Follow the instructions on the screen.

Uninstalling Windows Clients

To uninstall Sybase IQ, run the uninstaller from the Control Panel.

    1. On the Control Panel, choose Add or Remove Programs.

    2. Choose Sybase IQ Client Suite from the program list.

    3. Click Change/Remove.

    4. Follow the instructions on the screen.

Database Upgrades

The following topics describe how you can upgrade your existing Sybase IQ database to version 15.3. Depending upon your current version of the IQ database the method of upgrading will differ.

Pre-Upgrade Tasks

After you install Sybase IQ, perform these steps before you upgrade your existing Sybase IQ database to version 15.3.

- An issue affects any partition defined on a DATE column that contains a time portion. Before you upgrade databases that contain such partitions:
    1. Use **ALTER TABLE MERGE** to combine the partition with the next partition.
    2. Use **ALTER TABLE SPLIT** to divide the resulting partition into a definition with the same criteria, but no time portion.
- Drop and recreate all Sybase IQ 15.2 GA **TEXT** indexes that have not already been dropped and recreated as part of a version 15.2 ESD upgrade. **TEXT** indexes created in Sybase IQ 15.2 GA are not compatible with later versions of Sybase IQ.

Upgrading a Sybase IQ 15.x Database

Read this section for information about upgrading your database from Sybase IQ 15.x to Sybase IQ 15.3. There are separate upgrade procedures for simplex and multiplex databases.

Simplex Upgrade

Follow these steps to upgrade a simplex database from 15.x to 15.3.

1. Disconnect all users from the server.
2. Back up the database with the version of Sybase IQ you currently use.
3. Shut down the server.

    If the server aborts or hangs during shutdown, stop and do not proceed to the next step. Instead restart the database with the IQ version you are currently using and shut down the server. Proceed to the next step only on a clean shutdown.
4. Use Sybase IQ 15.3 to restart the server with these startup flags: **-gm 1** and **-iqro 1**.
5. Start Interactive SQL and connect to the database.
6. Use an **ALTER DATABASE UPGRADE** statement to upgrade the database.

See *Reference: Statements and Options > SQL Statements > ALTER DATABASE statement* for more information.

7. Run **sp_iqcheckdb ('allocation database')** and verify that there are no errors.
8. Shut down and restart the server normally (without the **-gm 1** and **-iqro 1** startup flags).
9. Back up the database.

Multiplex Upgrade

Run this procedure to upgrade a 15.x multiplex database to 15.3.

1. Disconnect all users from the servers.
2. Back up the database with the version of Sybase IQ you currently use.
3. Shut down all nodes in the multiplex.

   If the server aborts or hangs during shutdown, stop and do not proceed to the next step. Instead restart the database with the IQ version you are currently using and shut down the server. Proceed to the next step only on a clean shutdown.
4. Use IQ 15.3 to restart the coordinator with startup flags **-iqmpx_sn 1**, **-gm 1** and **-iqro 1**.
5. Start Interactive SQL and connect to the database.
6. Use an **ALTER DATABASE UPGRADE** statement to upgrade the database.

   See *Reference: Statements and Options > SQL Statements > ALTER DATABASE statement* for more information.
7. Run **sp_iqcheckdb ('allocation database')** and verify that there are no errors.
8. Shut down and restart the coordinator normally (without the **-gm 1** and **-iqro 1** startup flags).
9. Synchronize and restart all multiplex secondary servers.
10. Back up the database.

Note: IQ 15.3 introduces the concept of logical servers, which consist of one or more multiplex servers. The upgrade process changes the database configuration and modifies the login policies with appropriate logical server assignments to ensure that users continue to have access to same set of multiplex servers that they did prior to upgrade. Review the updated configuration and if needed, further modify it to suit your requirements.

See *Sybase IQ 15 Login Policy Migration* for details.

Upgrading a Sybase IQ 12.6/12.7 Database

Upgrading from IQ 12.6/12.7 database is very different than upgrading from a Sybase IQ 15.x database. This upgrade involves an unload/reload of the catalog. Follow this task flow to migrate a 12.6/12.7 database to Sybase IQ 15.3

- Read the Sybase IQ release bulletin for the latest information. See the special installation instructions section in the release bulletin for any installation instructions not included in this document.
- Upgrade Sybase IQ. If you are migrating from a version earlier than 12.6, you must upgrade to Sybase IQ 12.6 or later. All migration paths assume that you are migrating from Sybase IQ 12.6 ESD #11 or Sybase IQ 12.7 ESD #5 as a minimum.
- The **SORTKEY** function uses the International Components for Unicode (ICU) library, instead of the Sybase Unicode Infrastructure Library (Unilib®). Sort-key values created using a version of Sybase IQ earlier than 15.0 do not contain the same values created using version 15.0 and later. Regenerate any sort-key values in your database that were generated using a version of Sybase IQ earlier than 15.
- Sybase IQ 15 no longer supports creating custom collations. If you are rebuilding a database with a custom collation, the collation is preserved if you rebuild in a single step. If you unload the database, then load the schema and data into a database that you create, then you must use one of the supplied collations.
- Back up your current installation and save copies of any changes you made to default login and post-login scripts. Create your backups on removable media, like tape, DVD, or CD. Make sure the backups are readable before you proceed.
- To effectively plan your migration, review and understand the database migration utilities. Sybase IQ 15 includes an **iqunload** utility that you can use to re-create the schema for your database or simply migrate your database from 12.x to 15. An **iqlsunload** utility is available for 12.7 multiplex users who need to move local stores.
- Resolve potential migration errors before you migrate. Sybase IQ 15 no longer supports some 12.x features and objects. Update these objects before you begin for the migration utilities to function correctly.
- Follow the work flows for the migration you want to perform. There are separate migration paths for simplex and multiplex databases.

Database File Migration

Upgrading from Sybase IQ 12.6/12.7 to Sybase IQ 15 requires database file migration instead of a simple database upgrade. Sybase IQ 15 includes tools to ensure a smooth and error-free migration.

iqunload is a fully-automated migration tool included with Sybase IQ 15 that recreates the catalog store and reloads it with the existing database schema metadata. Only the catalog is recreated; the IQ data and temp dbspaces are not changed.

The migration process is completely automated, self-contained, and does not require your existing Sybase IQ 12.6 or 12.7 software. There are a few pre-migration steps you need to follow to ensure a smooth and error-free migration.

Migration Modes

iqunload has two working modes: schema unload and migration.

Migration mode uses a special engine bundled with Sybase IQ 15 , and does not require any earlier versions of Sybase IQ. Schema unload mode requires that the database be running in an existing Sybase IQ server 12.6 ESD #11, or later, or Sybase IQ 12.7 ESD #5, or later. Schema unload is not supported with IQ 15 or later servers.

Embedded IQ 12.7 Engine

iqunlspt is a self-contained subset of the Sybase IQ 12.7 (ESD #5) database engine that provides support for legacy database unloads. **iqunlspt** is installed with Sybase IQ 15.

iqunlspt runs in the background and contains no console or GUI window. When you run the migration tool, you will see the **iqunlspt** process running.

iqunlspt accepts startup arguments if your database requires special switches or memory settings. This is useful for providing diagnostic switches, such as **-z** or -**zr**, while troubleshooting. Since the migration process re-creates only your database schema, it is unlikely that you will need to set any command line arguments for the unload support engine.

Using iqunload

iqunload has the same look and feel as other IQ command line tools. It also accepts the same format for command line arguments.

Unlike **ALTER DATABASE UPGRADE**, **iqunload** migrates database options. During migration, all existing database options for which values are no longer the default value are saved, and a new database is created with Sybase IQ 15 options. Afterwards, the saved options are applied, overwriting any Sybase IQ 15 options. Sybase IQ 12.7/12.6 options that are no longer available in Sybase IQ 15 are ignored. Remember that only the catalog is re-created; no existing data in the IQ data stores is affected or changed. As an extra precaution, dbspaces are opened in read-only mode during the migration process.

This provides a window of opportunity during which you can revert back to Sybase IQ 12.7/12.6 in the event of failure.

Note: Sybase IQ 15 preserves previous settings of database options and reapplies them after you create the new 15 database. You no longer need to run SELECT * from SYS.SYSOPTIONS before upgrading.

Schema Unload Mode

Running iqunload in schema unload mode generates a reload SQL file that contains all the SQL statements necessary to re-create the legacy database schema.

You can use schema unload mode to create an empty version of a database. When you execute **iqunload** in schema unload only mode, using the command line argument **-n**, a file that contains all the SQL necessary to create the database schema is created.

Once the schema SQL file is generated, manually create a new database and execute the generated SQL file. No data is migrated. To migrate data into the new database, manually extract the data from your older database and manually load it into the new database.

Schema unload mode does not use the special unload support engine that comes with Sybase IQ 15. To run the schema unload mode, you must connect to your existing Sybase IQ database server.

Database Migration Mode

Running **iqunload** in database migration mode starts your legacy database, then uses the Sybase IQ 15 server to re-create the catalog store.

iqunload executes out these migration tasks:

1. Starts the Sybase IQ 12.7/12.6 database with the unload database engine.
2. Performs various internal database checks.
3. Determines if the database is a multiplex write server, in which case some extra steps are taken to migrate the writer to an IQ 15 coordinator.
4. Generates the schema (similar to schema unload mode).
5. Starts the Sybase IQ 15 server.
6. Creates a new database.
7. Applies the generated schema.
8. Executes various internal database checks to validate the metadata for migrated objects.
9. Backs up the operating system files. This backup does not use the SQL backup command.

Understanding Data Storage Changes and Schema Reload

Migrating to Sybase IQ 15 creates a new catalog store and changes some Sybase IQ 12.7/12.6 options.

dbspaces

In Sybase IQ 15, all user data should reside in a user dbspace comprised of one or more files. Migration converts main dbspaces into files under one user dbspace, IQ_MAIN, for the IQ main store, and temporary dbspaces into files under a single temporary dbspace, IQ_SYSTEM_TEMP, for a single IQ temporary store. Existing Catalog Store dbspaces remain as dbspaces with a single file. All of the old main dbspaces become files in the new `iq_main` user main dbspace. Migration sets the option PUBLIC.default_dbspace to the value `iq_main`.

The logical names for files created from converted dbspaces are the dbspace name followed by an underscore and the file ID. For example, a main dbspace with file ID 16384 becomes `IQ_SYSTEM_MAIN_16384`.

Main Store

Migration creates a new system file for the IQ_SYSTEM_MAIN dbspace that contains no tables. By default the name of this file is `new_main_store.iq`, but you can use the **ms_filename** argument to specify a different file name. The **iqunload** utility computes the size of the new IQ_SYSTEM_MAIN based on the size of your existing database.

If you accept the default settings for **iqunload -au -c**, the new store marked as MAIN has DBSpaceName = IQ_SYSTEM_MAIN, DBFileName = IQ_SYSTEM_MAIN and path = new_main_store.iq. For multiplex migration, the location of the new main store must be visible to all nodes on the multiplex, and you must use the **-ms_filename** argument to specify the path instead of the default value of `new_main_store.iq`.

When you migrate a database, you may specify which file to use for the new IQ_SYSTEM_MAIN dbspace, its name, whether or not to use a raw device, and the size of the main store and its reserve.

Unloading a Schema

Schema unloading allows you to verify schema correctness and estimates how long migration may take.

• Make sure that you copy the required scripts to your IQ 12.x server.
• Run **iqunload -n** to create the empty schema. This creates a SQL file with called `reload.sql` in the current directory that contains the entire schema. The `reload.sql` file also contains a template **CREATE DATABASE** command that is commented out.

> **Note:** The generated `reload.sql` does not contain any checkpoints. If you have a very large schema, you may want to edit `reload.sql` and add a few checkpoints. If you do not, IQ generates additional metadata objects that requiring extra (**-iqmc**) main cache memory.

• Use Interactive SQL or Sybase Central to create a new Sybase IQ 15.3 database with IQ SIZE and TEMPORARY SIZE clauses to create an IQ_SYSTEM_MAIN of 10GB and IQ_SYSTEM_TEMP of 5GB.
  For example:
```
CREATE DATABASE 'test.db'
IQ PATH 'test.iq'
IQ SIZE 10240
TEMPORARY PATH 'test.iqtmp'
TEMPORARY SIZE 5120
```
• Start the new database, connect to it, and execute the `reload.sql` file generated by **iqunload -n** against the new database. The execution time roughly approximates the actual time to allow for database migration, excluding validation checks.

Migrating IQ_SYSTEM_MAIN

You can specify the IQ_SYSTEM_MAIN size in the database migration command. The **-ms_size** parameter requires a value in MB, not GB. Omit **-ms_size** and **-ms_reserve** if you are specifying a raw device. For a raw device, you must specify an unused raw partition.

This statement creates an IQ_SYSTEM_MAIN on a raw device:

```
iqunload -au -ms_filename /dev/rdsk/c1t0d1 -c
"UID=DBA;PWD=SQL;DBF=latest.db"
```

Before and After Migration Files

Migration generates a set of files from your legacy database. Migrating the IQ 12.7 demo database to Sybase IQ 15 creates this set of corresponding files.

Table 18. Files before and after migration

Before	After	Description
asiqde-mo.db	asiqdemo.db.be-fore_schema_reload	The IQ 12.7 catalog database. This file is copied at the OS level upon successful migration; it is not a result of the SQL backup command.
asiqde-mo.log	asiqdemo.log	The database log file is regenerated when the migrated database is used with the IQ 15 server.
asiqde-mo.iq	asiqdemo.iq	The old IQ 12.7 IQ_SYSTEM_MAIN dbspace. This file and all other user dbspaces are not affected by the migration process. This dbspace is added as a file to a user main dbspace.
asiqde-mo.iqtmp	asiqdemo.iqtmp	The IQ_SYSTEM_TEMP dbspace. No operations are performed on this dbpace during migration. This file becomes the IQ 15 database temporary store.
asiqde-mo.iqmsg	asiqdemo.iqmsg.be-fore_schema_reload	The IQ 12.7 message file. This file is copied at the OS level upon successful migration.
	asiqdemo.db	The new IQ 15 migrated catalog database.
	new_main_store.iq	The new IQ_SYSTEM_MAIN dbspace for the migra-ted database.

Understanding iqunload

iqunload is a fully automated migration tool that recreates the catalog store and reloads it with the existing database schema metadata. Only the catalog is recreated; the IQ data and temp dbspaces are not changed.

iqunload has two working modes:

• **schema unload** - unloads a 12.x database schema.

• **migration** - migrates a 12.x database to Sybase IQ 15.

Syntax

iqunload [*options*] *directory* [@*data*]

data:[*environment variable* | *file*]

Parameters
The **iqunload** utility takes one or more parameters.

Table 19. iqunload parameters

Option	Description
-ap *<size>*	(Optional) Set the page size for the new catalog store.
-au	(Required, except when specifying -n) Migrate the database.
	You can specify **-au** or **-n** as an argument, but not both.
-c "*keyword=value*,..."	(Required) Supply database connection parameters. You must specify a DBF parameter to specify the name of the database file for migration. The file path is either absolute or relative to the server startup directory.
-dc	(Optional) Recalculate computed columns in the database.
-ms_filename	**-ms_filename** is optional for a simplex migration, but required for a multiplex migration.
	Specify filename for new empty IQ_SYSTEM_MAIN store created during database migration. If not specified, the default new main store is a file system file called `new_main_store.iq`
-ms_reserve	(Optional) Size of new IQ_SYSTEM_MAIN reserve, in MB. If un-specified, defaults to zero.
-ms_size	(Optional) Size of new IQ_SYSTEM_MAIN store, in MB, based on the database size. The minimum, assuming a default page size, is 200MB. If the user specifies an **-ms_size** value smaller than the computed value, IQ uses the computed value; otherwise the user–specified value is used.
-n	(Optional, except for schema unload.) Unload schema definition only. You can specify **-au** or **-n** as an argument, but not both.
	Note: You *must* use Sybase IQ 12.7 ESD 5 or later to use this option.

_____

Option	Description
-new_startline	(Optional) Specify startup switches for the new Sybase IQ 15.3 server that is the migration target. For a complete list of server startup switches, see the *Utility Guide*.
-o `<file>`	(Optional) Log output messages to file.
-q	(Optional) Suppress messages and windows.
-r `<file>`	(Optional) Specify name of generated the `reload.sql` command file.
-t `<list>`	(Optional) Output listed tables only. Can specify OwnerName.Table-Name, or TableName alone.
-v	(Optional) Return verbose messages.
-y	(Optional) Replace existing reload schema SQL script with new output without confirmation.

Example 1

Migrate a simplex database to 15.x server:

```
iqunload -au -c
"UID=DBA;PWD=SQL;DBF=/mydevice/test_dir/test2.db"
```

Example 2

Unload schema from database and rename the `reload.sql` file to
`test2_reload.sql`:

```
iqunload -n -c
"UID=DBA;PWD=SQL;DBF=/test_dir/test2/test2.db;ENG=
 myserver_test2" -r "/test_dir/unload_dir/test2_reload.sql"
```

Example 3

Migrate database `test3.db`. The **START** connection parameter specifies switches for starting the database being reloaded. The **–new_startline** parameter specifies switches for starting the `utility_db` to create the new catalog store.

```
iqunload -au -c
"UID=DBA;PWD=SQL;DBF=test3.db;START=-ch 128M -iqmc 50" -
new_startline "-ch 256M -iqtc 400"
```

Do not include -**iqnotemp** *XYZ* in the new start line or else migration will fail. As part of the migration process, temp files will be added to IQ_SYSTEM_TEMP. If the server is started with iqnotemp then the iqunload will NOT be able to add these temp files.

Example 4

Migrate the database `asiqdemo.db` using a raw device for a new IQ_SYSTEM_MAIN store:

```
iqunload –au –c
"UID=DBA;PWD=SQL;DBF=asiqdemo.db" –ms_filename "/dev/rdsk/c4t0d0s3"
```

```
iqunload –au –c
"UID=DBA;PWD=SQL;DBF=asiqdemo.db" -ms_filename \\\\.\\PhysicalDrive2
```

Usage

For schema unloads, the **iqunload** requires the following minimum server versions:

- Sybase IQ version 12.6 ESD #11
- Sybase IQ version 12.7 ESD #5

For schema unload, start your existing 12.6 or 12.7 server and specify the **-n** parameter on the **iqunload** command. Schema unload creates schema load scripts and output files in the current directory.

Permissions

DBA authority is required.

Determining Migration Switches

In some cases, you may need to set some switches to migrate your database.

The **-au** parameter starts these two servers:

- **iqunlspt** – a special unload engine based on Sybase IQ 12.7 ESD 5.
- **iqsrv15** – the new Sybase IQ 15.3 server.

To minimize startup times, first verify that your server starts cleanly in IQ 12.6 or 12.7.

Default iqunlspt Arguments

By default, **iqunlspt** starts on your Sybase IQ 12.6/12.7 database with these command line arguments:

```
iqunlspt -iqnotemp 100 -iqro 1 -c 48MB -gc 20 -gd
all -gk all -gl all -gm 1 -gu all -ti 4400 -x shmem
```

iqunlspt also includes the **-n** parameter followed by a special randomly generated server name.

These cache settings suffice for most migration cases, although you may need to increase these values. At migration, data queries execute against the system catalogs, not IQ data, so the **iqunlspt** engine needs lower cache levels than complex IQ queries or multiple concurrent users. The time required to start your existing Sybase IQ 12.6 or 12.7 database is the same amount of time required to start the **iqunlspt** process. This time is included in the startup time for **iqunload**.

_____

Default iqsrv15 Arguments

iqsrv15 starts with these command line arguments:

```
iqsrv15 -gp 4096 -c 40p -gc 20 -gd all -gk all -gl all
-gm 1 -gu all -qi -qs -ti 4400
```

iqsrv15 also includes the **-n** parameter followed by a special randomly generated server name. The **-c 40p** setting provides a larger cache for the catalog store, allowing the Sybase IQ 15 engine to execute many schema DDL statements. Both server start commands use the default values for **-iqmc** and **-iqtc**. If your Sybase IQ 12.6/12.7 server requires larger values for startup performance, consider adding these to the START= parameter (see below).

iqunload Results

The **iqunload** utility generates script and output files in the $IQTMP15 directory and removes them on completion.

By default, Sybase IQ allocates the lesser of 256MB or 40% of physical memory for the Catalog Store cache size upper limit. Insufficient cache causes **iqunload** errors like Tried to allocate 4167598080 bytes, failed. You can control the upper and lower limits for the Catalog Store cache size by setting the **–ch** and **-cl** startup switches through the StartLine [START] connection parameter. See the *Utility Guide*.

Users with wide tables (large numbers of column/null values) should not decrease the Catalog Store page size for database migration.

If the database being migrated contains invalid views, Sybase IQ completes migration but issues user warnings. This may occur, for example, if tables involved in a view are dropped.

If you set the *IQTMP15* environment variable, you must set it to a valid directory name.

If the database being migrated is encrypted, then you must specify the DBKEY connection parameter to provide the encryption key for the database. The migrated database uses the same encryption key.

The **iqunload** utility automatically starts the old version of the server (12.6 or 12.7) to upgrade the catalog .db file, and shuts down the old server after the unload completes. The utility uses default values for the various cache sizes (catalog cache, main cache, temp cache). If the database being migrated is normally run with much higher cache values, specify these values as part of the START connection parameter to ensure that the server has adequate catalog cache, main cache, temp cache for the database being unloaded.

The current defaults are 48MB for the catalog cache size, 32MB for the main cache size and 24MB for the temp cache size.

Logging and Correcting Errors

Check the database engine output logs to isolate migration problems. There are separate output logs for the Sybase IQ 15 and Sybase IQ 12.7 database engines.

Sybase IQ 15 Engine Logs

Use the **-new_startline** parameter on the **iqunload** command to specify diagnostic parameters **-z** and **-zr all** to start the Sybase IQ 15 server with extra logging, .

For example:

```
iqunload -au -c "UID=DBA;PWD=SQL;DBF=/iq-15/
unload/127/db/iq127db.db" -new_startline "-z -zr all"
-o iq15db.out
```

This statement causes the new `iqsrv15` database engine to produce more verbose logging output to the file `iq15db.out`.

Sybase IQ 12.7 Engine Logs

To start the bundled Sybase IQ 12.7 engine with extra logging, use the START= connection property to specify the **-z** and **-zr all** parameters. For example:

```
iqunload -v -au -c "UID=DBA;PWD=SQL;DBF=/iq-15/
unload/127/db/iq127db.db;START=-z -zr all
-o iq127db.out"
```

This statement causes the bundled Sybase IQ 12.7 engine to produce more verbose logging output. Examine the output file `iq15db.out` or `iq127db.out` for server error messages.

For details about the **-z** diagnostic parameter, see the *Utility Guide*. For information about request logging (**-zr**), which has been updated for Sybase IQ 15, see "Request logging" in *SQL Anywhere Server – SQL Usage > Monitoring and Improving Database Performance > Improving database performance > Other diagnostic tools and techniques*.

Server Not Found

A message similar to this indicates that **iqunload** started the database but could not connect:

```
SQL error:Database server not found
```

Check to see if an `iqunlspt` process is running, and stop the process before retrying **iqunload**.

For example, here is **top** output:

```
load averages:  1.45,  1.19,  0.80; up 3+16:22:31
10:2
172 processes: 168 sleeping, 2 zombie, 2 on cpu
CPU states: 79.1% idle, 18.9% user,  1.9% kernel,  0.0%
iowait,  0.0% swap
Memory: 16G phys mem, 13G free mem, 16G swap, 16G free
swap

PID USERNAME LWP PRI NICE  SIZE   RES STATE    TIME   CPU COMMAND
```

```
21223 ybrown 1  59   0 2908K 1844K cpu        0:00  0.12% top
21172 ybrown 476 59    0  319M  264M sleep     0:01  0.05% iqunlspt
24890 ybrown 14  29  10   79M   43M sleep      0:49  0.03% java
20103 ybrown 1  59    0 7916K 2080K sleep      0:00  0.00% sshd
```

To stop the process, enter the command **kill –9** and supply the process ID, for example:

```
kill –9 21172
```

In Task Manager, click the Processes tab, and locate the image name `iqunlspt`. Right-click `iqunlspt` and choose End Process.

If you try to run **iqunload** without killing the orphaned `iqunlspt` process, you may see this:

```
SQL error: Unable to start specified database: autostarting database
failed.
```

Obsolete Stored Procedures

Migration replaces 12.7 login procedures to support new Sybase IQ 15 login management functionality.

The procedure `sp_login_environment` replaces the 12.7 default login procedure `DBA.sp_iq_process_login` and `dbo.sa_post_login_procedure` replaces the 12.7 default post login procedure `DBA.sp_iq_process_post_login`. The **iqunload** utility generally replaces obsolete options with their new defaults, but if the 12.7 option is set on a specific user instead of PUBLIC (the default), the log file may report errors similar to:

```
E. 10/31 16:53:40. Login procedure
'DBA.sp_iq_process_login' caused SQLSTATE '52W09'
E. 10/31 16:53:40. Procedure 'sp_iq_process_login' not
found
```

Note: See *SQL Anywhere Server - Database Administration > Configuring Your Database > Managing user IDs, authorities, and permissions* for an overview of login management functionality.

Working With Large Schemas

With large schemas, the default cache settings may be too small and could exhaust dynamic memory in the Sybase IQ 15 server. Use the **-c** server switch to increase the server's cache memory, and the **-new_startline** to pass this switch to the server.

A diagnostic example includes these switches:

```
-ca 1
-c 1000m
-o /iq15outputdir/iq15console.out
```

Note: The value shown for **-c** is in bytes and arbitrary. Set switches appropriately for your system. To specify megabytes, use the m suffix, as shown.

The switches in the example perform as follows:

- **-ca 1** – enables dynamic catalog cache sizing, and logs memory cache statistics to the console.
- **-c 1000m** – sets initial catalog cache at 1GB.
- `-o /outputdir/iq15console.out` – specifies the log file for console output.

Now you can load `-o /outputdir/iq15console.out` into your editor and examine the log entries. You can watch how the catalog store adjusts the cache and determines if the setting is appropriate.

Working with Extremely Large Schemas

Generating a reload schema (**iqunload -n**) places the entire schema in a single `reload.sql` file. In some cases, you may need to break a very large `reload.sql` file into pieces that can be executed sequentially. This also helps the server manage the cache.

If the migration fails (**iqunload -au**) with dynamic memory exhausted and you have set cache settings as high as your hardware and operating system limitations allow, contact Sybase for assistance.

Sybase IQ 12.7 Local Store Utility

In Sybase IQ 15 multiplex configurations, multiple nodes can write to the main store, which eliminates the need for local stores. **iqlsunload** is a migration tool that you can use to unload an IQ 12.7 local store.

iqlsunload is included with all versions of Sybase IQ starting with 12.7 ESD5, and is located in the `$IQDIR15/lsunload` directory.

iqlsunload

iqlsunload unloads an IQ 12.7 local store.

Syntax
iqlsunload [*options*] *directory* [*@data*]

data:[*environment variable* | *file*]

Parameters
directory is a required parameter that identifies a directory on the file system where you want **iqlsunload** to unload the data files. This directory must be relative to the database on the database server and exist before you run **iqlsunload**.

iqlsunload takes one or more optional parameters.

Table 20. iqlsunload command line switches

Parameter	Description
-al	Unload IQ local store schema and data.

Parameter	Description
-c "*keyword=value*;..."	Supply database connection parameters.
-h	Print out the syntax (help) for the utility.
-o *<file>*	Log output messages, including errors, to file.
-q	Suppress messages and windows.
-r *<directory>*	Specify the directory where SQL scripts are generated. The default reload command file is `reload.sql` in the current directory. The directory is relative to the current directory of the client application, not the server.
-t *<list>*	Output listed tables only. Can specify OwnerName.TableName or TableName alone. (Cannot be specified with **-al**).
-v	Output verbose messages.
-y	Replace existing reload schema SQL script without confirmation.

Usage

- **iqlsunload** requires a minimum server version of Sybase IQ 12.7 ESD #5.
- The specified directory is a path relative to the database on the database server, not relative to the path where **iqlsunload** is executed.
- **iqlsunload** connects to the database using connection parameters **-c** in a way which is consistent with other tools like **dbping** and **dbisql** as described in the *Utility Guide*.
- On Windows, you must specify double instead of single slashes in all path and file names.
- Run **iqlsunload** from the `$IQDIR15/lsunload` directory to pick up updated libraries before resolving any IQ 12.7 libraries.

Objects Unloaded
Unloading the local store does not unload database objects that are not part of the local store.

The **-al** parameter unloads these persistent objects:

- Base tables
- Global temporary tables
- Indexes
- Domains (user-defined data types)
- Constraints (column check constraint, table constraint, primary key, foreign key, unique, default, IQ unique, not null)
- Views
- Stored procedures and functions
- Messages

- Remote servers and external logins
- Events

Empty User Names

Sybase IQ 15.3 no longer allows users with empty user names. Users with empty user names cannot be dropped with the 12.6 or 12.7 server and cannot be migrated. The schema reload operation warns that an empty user name has been encountered and that the user will not be re-created. The reload operation ignores such users and any associated objects.

Unloading Tables

iqlsunload ignores any system tables or nonexistent tables specified. When unloading tables:

- Join indexes, which do not exist in a local store, are unloaded only when the user specifies unloading schema and data for a list of tables (**-t**). All tables involved in the index must be specified with the **-t** parameter.
- When extracting table schema and data only, be sure that the collation of the database you run the reload script against matches the collation of the database specified on the initial **iqlsunload** command.
- If you do not qualify table names with owner names, **iqlsunload** extracts table data from all tables with that table name.

Example

Unload local stores from a database called `mpxtest2`, extracting any table data to the directory `/mydevice/test_dir/unload_dir`:

```
iqlsunload -o iqunload_624.out -al
-c "UID=DBA;PWD=SQL;ENG=myserver_mpxtest02"
/mydevice/test_dir/unload_dir
```

This command generates these scripts in the current directory:

- `reload_schema.sql` – recreates schema for unloaded objects (either objects from local store or tables selected by the user.) This script is executed against a node that writes to the multiplex. This node can be either the existing writer node or a writer or coordinator for the multiplex after migration, depending on where you plan to recreate the schema.
- `extract_data.sql` – extracts table data for the unloaded tables from the local store. Execute this script in Interactive SQL while connected to the query node from which it was generated. When this script executes, it generates the data files into the directory data. This script's contents follow the guidelines set out in "Moving Data In and Out of Databases" in the *Sybase IQ System Administration Guide*.
- `reload_data.sql` – loads extracted table data. This script is executed on the node where you ran `reload_schema.sql` and reloads the data extracted from the `extract_data.sql` file.

Permissions

DBA authority is required.

_____

Avoiding Potential Migration Problems

Sybase IQ 15 no longer supports some Sybase IQ 12.7/12.6 features and objects. To avoid potential problems, correct these items before you migrate your database.

Note: During migration, the Sybase IQ 15 server creates a message file (`*.iqmsg.R`) as it reloads the generated schema. This file is normally deleted as part of a cleanup operation for successful migrations. If the migration fails during the reload stage, cleanup does not occur, and `*.iqmsg.R` remains in the `unload` directory. `*.iqmsg.R` may contain information that can help solve your migration problems.

Addressing Unsupported Objects

Check the schema for objects that Sybase IQ 15 does not support.

Run **iqunload** in schema unload mode (**iqunload -n)** against the legacy server to generate a script(`reload.sql`) that contains the entire database schema. Compare the contents of this script against this table to find identify objects not supported by Sybase IQ 15.

Table 21. Troubleshooting unsupported metadata

Object	Details	Action
Invalid database, table, or user names	Table names cannot contain double quote characters.User names and database names cannot contain double quote characters, single quote, and semicolon characters. User names and database names cannot start or end with a space.	Change these names.
Join indexes	**iqunload** does not run if the database to be migrated contains join indexes.	Use Sybase Central to view the join index definitions. Copy them to a file. Drop join indexes before migrating data. After migrating data, recreate the join indexes.
Database with BLANK PADDING OFF	**iqunload** searches BLANK PADDING OFF databases for any indexes that would become invalid after migration. **iqunload** fails and lists indexes and constraints that must be dropped and in which order.	Drop these indexes and constraints before the schema reloads and recreate the indexes and constraints after schema reload has been completed.
Unenforced constraints	**iqunload** fails and lists unenforced constraints that must be dropped.	Drop unenforced constraints before proceeding with migration.

Object	Details	Action
Old 1–byte FP or old 2–byte FP indexes	Databases created with IQ 12.4.2 or earlier may have these indexes. Because these indexes were automatically created by Sybase IQ, you cannot drop and recreate them; you must rebuild them.	Allow **iqunload** to check for these and list them. Rebuild these indexes using **sp_iqrebuilddindex** before migration. The rebuilt indexes are upgraded.

Correcting Invalid SQL Syntax

Some SQL syntax in Sybase IQ 15 is different that previous versions and can cause **iqunload** failures.

Table 22. Troubleshooting schema issues

Problem	Solution
A **DECLARE LOCAL TEMPORARY TABLE** statement in a procedure or trigger causes a syntax error if the table name is prefixed with an owner name.	Remove the owner name.
If a **CREATE TRIGGER** statement does not include an owner name for the table on which the trigger is defined, and the table must be qualified with an owner when referenced by the user executing the reload.sql file, the statement fails with a Table 'table-name' not found error.	Prefix the table name with the owner name.
If an object name (such as a table, column, variable or parameter name) corresponds to a reserved word introduced in a later version of Sybase IQ or SQL Anywhere (such as NCHAR), the reload fails. (For reserved words, see *Sybase IQ Reference: Building Blocks, Tables, and Procedures* and *SQL Anywhere Server SQL Reference*.) For example: ```CREATE PROCEDURE p( )\nBEGIN\nDECLARE NCHAR INT;\nSET NCHAR = 1;\nEND```	Change all references to the reserved word to use a different name. For variable names, prefixing the name with @ is a common convention that prevents naming conflicts.
Views that use Transact–SQL® outer joins (by specifying *= or =*) may not be created properly when they are reloaded.	Add the following line to the reload script: ```SET TEMPORARY OPTION\ntsql_outer_joins='on'``` Also set this option for your database. Rewrite any views or stored procedures that use Transact–SQL outer joins.
Stored procedures that use Transact–SQL outer joins may not work correctly.	Rewrite views and stored procedures.

Problem	Solution
Functions that have OUT or INOUT parameters cannot be reloaded.	OUT and INOUT parameters are no longer supported. Drop these functions before reloading.

Running iqunload -n

Sybase recommends that you run this procedure to avoid the SQL syntax problems that cause failures.

1. Copy the following scripts from `$IQDIR15/unload` to `$ASDIR/scripts`:

   - `unloadold.sql`
   - `unload.sql`
   - `optdeflt.sql`
   - `opttemp.sql`

2. Use your 12.x server to start the database you want to migrate.

3. Use the **iqunload** utility to run a schema unload.

   For example:

   ```
   iqunload –n –c "UID=DBA;PWD=SQL;ENG=my_eng;DBN=my_dbname".
   ```

   The schema unload produces a `reload.sql` script that contains the schema for the database being migrated in the directory from which you ran the unload.

   The generated `reload.sql` file contains a sample **CREATE DATABASE** statement, for example:

   ```
   -- CREATE DATABASE command: CREATE DATABASE
   '/iq-15/unload/127/db/iq127db.db' LOG ON
   '/iq-15/unload/127/db/iq127db.log' CASE IGNORE
   ACCENT IGNORE PAGE SIZE 4096 COLLATION 'ISO_BINENG'
   BLANK PADDING ON JCONNECT OFF CHECKSUM OFF IQ PATH
   'iq127db.iq' IQ SIZE 50 TEMPORARY PATH
   'iq127db.iqtmp' TEMPORARY SIZE 25 MESSAGE PATH
   'iq127db.iqmsg' IQ PAGE SIZE 131072 BLOCK SIZE 8192
   ```

   Running the `reload.sql` script does not execute the sample statement, but you can use this generated statement as a template when you create your database with Sybase IQ 15.

4. Start the utility database using the Sybase IQ 15 Server, and create a new, empty, Sybase IQ 15 database.

5. Use Interactive SQL to run the `reload.sql` script that contains the unloaded schema against the new Sybase IQ 15 database. This loads the schema of the old database into an Sybase IQ 15 database. If there are any SQL syntax errors, you receive warnings when running the SQL script. Ignore the errors to get a complete list of problems. Modify the database to be migrated to eliminate the errors found.

6. Perform this process iteratively, if necessary, until you can cleanly reload the schema unloaded from the database to be migrated, against the empty 15 database.

Migrating Simplex Databases

Perform all of the steps in this task flow to migrate a Sybase IQ 12.7/12.6 database simplex database to Sybase IQ 15.

Verify the Sybase IQ 12.7/12.6 Database

Start your Sybase IQ 12.7/12.6 database, then use **sp_iqcheckdb** to verify that your database is clean and error-free.

1. Start and connect to your Sybase IQ 12.7/12.6 database.

2. Issue a **CHECKPOINT** command.

3. Run the **sp_iqcheckdb** stored procedure in verify mode: sp_iqcheckdb 'verify database' If you run the procedure from Interactive SQL, redirect output to a file by entering: dbisql -c "..." "sp_iqcheckdb ('verify database')" >& filename where "..." represents startup parameters for your database.

4. Issue a **COMMIT** statement.

5. Check **sp_iqcheckdb** results for errors. Repair any index and allocation problems.

6. Use the SQL backup command to back up the Sybase IQ 12.7/12.6 database.

7. Shut down your Sybase IQ 12.7/12.6 server.

Note: For information on interpreting the **sp_iqcheckdb** results and corrective action, see *System Administration Guide: Volume 1 > System Recovery and Database Repair*. If you need to contact Sybase Technical Support, you must provide the output from **sp_iqcheckdb**.

Migrate the Sybase IQ 12.7/12.6 Database

Make sure that the database file is not in use, and run the **iqunload** utility with the **-au** (migrate database) and **-c** (connection parameters) to migrate the database.

This command migrates the simplex database mytest and saves output in unload.out in the current directory:

```
iqunload –au –c "uid=DBA;pwd=SQL;dbf=mytest" –o unload.out
```

The database and the **iqunload** utility must be on the same machine to migrate the database, or **iqunload** returns an error. **-o** is an optional switch that sends a copy of the console output to the specified log file, here named unload.out.

Because the example specified DBF=mytest.db, the **iqunload** utility attempts to connect to this database in the current directory. You can also specify the full path to the database, as shown in the following example:

```
iqunload -au -c
"dbf=/ybrown/iq-15/unload/iq127db.db;uid=DBA;pwd=SQL"
Output:
```

```
    Sybase IQ Unload Utility Version 15.0.0.5533
    Connecting and initializing
    Unloading user and group definitions
    Unloading table definitions
    Unloading index definitions
    Unloading functions
    Unloading view definitions
    Unloading procedures
    Unloading triggers
    Unloading SQL Remote definitions
    Creating new database
    Creating indexes for (1/14)
        "DBA"."sales_order"
```

```
    Creating indexes for (2/14)
        "DBA"."sales_order_items"
    Creating indexes for (3/14) "DBA"."contact"
    Creating indexes for (4/14) "DBA"."customer"
    Creating indexes for (5/14) "DBA"."fin_code"
    Creating indexes for (6/14) "DBA"."fin_data"
    Creating indexes for (7/14) "DBA"."product"
    Creating indexes for (8/14) "DBA"."department"
    Creating indexes for (9/14) "DBA"."employee"
    Creating indexes for (10/14)"DBA"."alt_sales_order"
    Creating indexes for (11/14) "DBA"."alt_sales_order_items"
    Creating indexes for (12/14) "DBA"."iq_dummy"
    Creating indexes for (13/14) "DBA"."emp1"
    Creating indexes for (14/14) "DBA"."sale"
```

```
Successfully backed up file "/ybrown/iq-15/
unload/127/db/iq127db.db" by renaming it to "/ybrown/iq-15/unload/
127/db/iq127db.db.before_schema_reload".
Successfully backed up file "/ybrown/iq-15/unload/127/db/
iq127db.iqmsg"
by renaming it to "/ybrown/iq-15/unload/127/
db/iq127db.iqmsg.before_schema_reload"
Successfully reloaded schema of database "/ybrown/iq-15/unload/127/
db/iq127db.db".
```

Perform post–migration tasks. Make sure that the migration completed correctly. Back up your new databases.

Verify the Migrated Simplex Database
Start the migrated database in read-only mode and perform post-migration tasks to verify the migration.

1. Start the Sybase IQ 15 database in read-only mode: start_iq -iqro 1

   When starting the coordinator in Sybase IQ 15, use the same port as the Sybase IQ 12.7/12.6 writer server.
2. Issue a **CHECKPOINT** command.
3. Run the Sybase IQ 15 version of **sp_iqcheckdb** in verify mode:

   ```
   sp_iqcheckdb ('verify database')
   ```

If you run the procedure from Interactive SQL, redirect output to a file by entering:

```
dbisql -c "..." "sp_iqcheckdb ('verify database')" >& filename
```

where "..." represents startup parameters for your database.

4. Issue a **COMMIT** statement.
5. Check **sp_iqcheckdb** results for errors.

   If there is an error, you can revert to the previous database as long as you do not restart the database in write mode. To revert back to the IQ 12.7 catalog , copy all the `.before_schema_reload` files to the same file without the `.before_schema_load file` extension.

6. Once you have successfully performed the read-only checks, you can stop the database server and restart it in write mode.

Note: For information on interpreting the **sp_iqcheckdb** results and corrective action, see "System Recovery and Database Repair" in System Administration Guide: Volume 1. If you need to contact Sybase Technical Support, you must provide the output from **sp_iqcheckdb**.

Migrating Multiplex Databases

Follow this workflow to migrate multiplex databases. Review the task summary and perform all steps in sequence.

Synchronize and Shutdown Multiplex Query Nodes

Check the SQL Remote and multiplex server log files for synchronization problems.

1. Start the multiplex server.
2. If it is not already, start SQL Remote on all nodes in the multiplex. These are normally started by Sybase Central when starting servers.
3. Give the multiplex time to propagate any changes throughout the multiplex. To do this, look at the write server console log file and check that the events starting with ev_iqmpx have successfully executed. By default, the server console log file is created in $ASDIR/logfiles.

   For example:

```
Now accepting requestsOS Available: 933096K, Working Set: 83988K,
Cache Target: 11483K
OS Available: 860680K, Working Set: 83996K, Cache Target: 11483K
Next time for 'ev_iqmpxq2w' is 2008/11/23 22:03:00.000
Next time for 'ev_iqmpxstatus' is 2008/11/23 22:03:00.000
OS Available: 859232K, Working Set: 84112K, Cache Target: 11489K
OS Available: 861052K, Working Set: 84424K, Cache Target: 11489K
OS Available: 860972K, Working Set: 84428K, Cache Target: 11489K
OS Available: 850248K, Working Set: 85540K, Cache Target: 11579K
OS Available: 850104K, Working Set: 85568K, Cache Target: 11579K
Next time for 'ev_iqmpxq2w' is 2008/11/23 22:04:00.000
Next time for 'ev_iqmpxstatus' is 2008/11/23 22:04:00.000
```

```
OS Available: 850120K, Working Set: 85600K, Cache Target: 11579K
Next time for 'ev_iqmpxq2w' is 2008/11/23 22:05:00.000
Next time for 'ev_iqmpxstatus' is 2008/11/23 22:05:00.000
OS Available: 852668K, Working Set: 85604K, Cache Target: 11579K
```

4. Wait for SQL Remote to scan its log files a few times and view the logs. To view these log files from Sybase Central, right-click the multiplex server and select SQL Remote | View SQL Remote Log.

   The console indicates that SQL Remote is hovering at the end of the active log. Wait for SQL Remote to process any messages. For example:

```
I. 11/23 22:06:10. Scanning logs starting at offset 0001787252
I. 11/23 22:06:10. Hovering at end of active log
```

5. Shut down SQL Remote and multiplex servers.

   If you simply shut down the multiplex servers, the SQL Remote servers will hover, then detect that the multiplex servers are no longer running and shut themselves down. By default, the SQL Remote servers should shut themselves down within 60 seconds.

6. Shut down query servers in the multiplex. They are no longer required.

7. If the logs report no errors, verify the database.

Migrate Your Local Stores

Move any local stores for your 12.x multiplex query nodes before you migrate your data to Sybase IQ 15. **iqlsunload** connects to a Sybase IQ 12.7 database with an IQ local store and creates scripts that unload its schema and data.

To migrate your local store, consolidate node-specific information into either the existing 12.7 writer or the new Sybase IQ 15 main store. Customize the process to meet your data requirements.

For query nodes with node- or department-specific information, use tablespaces and partitioning to achieve the same results.

If information is duplicated across your query nodes, you may need to migrate only a single query server's local store. The duplicated information on the other query servers becomes redundant and can be ignored for multiplex migration.

Partitioning Query Server Data

If the same table exists on multiple query nodes, and each node has its own subset of the data, you must manually edit the local store migration scripts.

For a department-specific `employee` table on each query server, follow these basic steps:

1. Unload the schema and data from the query nodes.

   The `reload_schema.sql` script produced for each query node contains the same schema definition for `employee`.

2. Execute the `reload_schema.sql` from one of the query nodes against either the existing 12.7 writer or the new Sybase IQ 15 main store.

---

3. Execute the `reload_data.sql` script from each of the query nodes against the same server used in the previous step.

   This procedure creates the `employee` table once but loads each query node data set.

Addressing Overlapping Query Server Data

If the same table exists on each query node with overlapping data sets, you must resolve the issue. You can edit the extract data files to ensure that the data sets are unique, or you can rename the tables and then reload all the unique tables.

1. Run **iqlsunload** against all your query servers with local stores that have the data you want to consolidate.
2. Modify the `reload_schema.sql` and `reload_data.sql` files to use the new table names. Do not modify `extract_data.sql`; it references the table found in the query server's local store.
3. Run `extract_data.sql` from each of the nodes.

The following example shows modifications to the **iqlsunload** output to carry out step 2. Suppose that the `reload_schema.sql` script contains:

```
CREATE TABLE "DBA"."sales_order"

    "id"              unsigned int NOT NULL  IQ UNIQUE (648),
    "cust_id"            unsigned int NOT NULL  IQ UNIQUE (111),
    "order_date"          "datetime" NOT NULL  IQ UNIQUE (376),
    "fin_code_id"        char(2) NULL  IQ UNIQUE (1),
    "region"             char(7) NULL  IQ UNIQUE (5),
    "sales_rep"          unsigned int NOT NULL  IQ UNIQUE (75),
PRIMARY KEY ("id"),
```

Modify `reload_schema.sql` to:

```
CREATE TABLE "DBA"."q1_sales_order"

    "id"              unsigned int NOT NULL  IQ UNIQUE (648),
    "cust_id"            unsigned int NOT NULL  IQ UNIQUE (111),
    "order_date"          "datetime" NOT NULL  IQ UNIQUE (376),
    "fin_code_id"        char(2) NULL  IQ UNIQUE (1),
    "region"             char(7) NULL  IQ UNIQUE (5),
    "sales_rep"          unsigned int NOT NULL  IQ UNIQUE (75),
    PRIMARY KEY ("id"),
```

`extract_data.sql` contains:

```
---- Extract Table Data for table sales_order
-- NOTE: Approximately 57672 bytes of storage space.
-- will be required to extract the data for this table.
--
-- The following will unload the data for table
sales_order, row group 1, column group 1
```

```
SET TEMPORARY OPTION temp_extract_name1 =
'DBA_sales_order_1_1_DATA_1.inp';
SET TEMPORARY OPTION temp_extract_name2 =
```

```
'DBA_sales_order_1_1_DATA_2.inp';
SET TEMPORARY OPTION temp_extract_name3 =
'DBA_sales_order_1_1_DATA_3.inp';
SET TEMPORARY OPTION temp_extract_name4 =
'DBA_sales_order_1_1_DATA_4.inp';
SET TEMPORARY OPTION temp_extract_name5 =
'DBA_sales_order_1_1_DATA_5.inp';
SET TEMPORARY OPTION temp_extract_name6 =
'DBA_sales_order_1_1_DATA_6.inp';
SET TEMPORARY OPTION temp_extract_name7 =
'DBA_sales_order_1_1_DATA_7.inp';
SET TEMPORARY OPTION temp_extract_name8 =
'DBA_sales_order_1_1_DATA_8.inp';

SELECT id, cust_id, order_date,
IFNULL(fin_code_id, @null_string, fin_code_id),
IFNULL(region, @null_string, region), sales_rep
FROM "DBA"."sales_order"
WHERE rowid( "sales_order" ) >= 1
AND  rowid( "sales_order" ) <= 648;
```

```
SET TEMPORARY OPTION temp_extract_name1 = '';
SET TEMPORARY OPTION temp_extract_name2 = '';
SET TEMPORARY OPTION temp_extract_name3 = '';
SET TEMPORARY OPTION temp_extract_name4 = '';
SET TEMPORARY OPTION temp_extract_name5 = '';
```

Leave extract_data.sql code unchanged to extract the sales_order table from the query server.

Suppose that reload_data.sql contains:

```
-- Reload Table Data for table "sales_order"
-------------------------------------------------
ALTER TABLE "DBA"."sales_order" MODIFY cust_id NULL;
ALTER TABLE "DBA"."sales_order" MODIFY order_date NULL;
ALTER TABLE "DBA"."sales_order" MODIFY sales_rep NULL;

SET @max_row_id =
( SELECT MAX( rowid( "sales_order" ) )+1
FROM "DBA"."sales_order" );

SET @load_statement =
'LOAD TABLE "DBA"."sales_order"
(id, cust_id, order_date, fin_code_id NULL(
'''||@null_string||''' ) , region NULL(
'''||@null_string||''' ) , sales_rep)
FROM
'''||@extract_directory||'DBA_sales_order_1_1_DATA_1.
inp'',
'''||@extract_directory||'DBA_sales_order_1_1_DATA_2.
inp'', '''||@extract_directory||'DBA_sales_order_1_1_DATA_3.
inp'',
'''||@extract_directory||'DBA_sales_order_1_1_DATA_4.
```

```
inp'',
'''||@extract_directory||'DBA_sales_order_1_1_DATA_5.
inp'', '''||@extract_directory||'DBA_sales_order_1_1_DATA_6.
inp'', '''||@extract_directory||'DBA_sales_order_1_1_DATA_7.
inp'', '''||@extract_directory||'DBA_sales_order_1_1_DATA_8.
inp'' ROW DELIMITED BY ''\n'' QUOTES ON
ESCAPES OFF DEFAULTS OFF FORMAT ASCII
IGNORE CONSTRAINT ALL 0 START ROW ID
'||@max_row_id;
```

```
CALL IqExecuteCommand( @load_statement );
ALTER TABLE "DBA"."sales_order" MODIFY cust_id NOT
NULL;
ALTER TABLE "DBA"."sales_order" MODIFY order_date NOT
NULL;
ALTER TABLE "DBA"."sales_order" MODIFY sales_rep NOT
NULL;
```

Change reload_data.sql to:

```
-- Reload Table Data for table
"q1_sales_order"
------------------------------------------------
ALTER TABLE "DBA"."q1_sales_order" MODIFY cust_id NULL;
ALTER TABLE "DBA"."q1_sales_order" MODIFY order_date
NULL;
ALTER TABLE "DBA"."q1_sales_order" MODIFY sales_rep
NULL;

SET @max_row_id = ( SELECT MAX( rowid( "q1_sales_order"
) )+1 FROM "DBA"."q1_sales_order" );
```

```
SET @load_statement =
'LOAD TABLE "DBA"."q1_sales_order"
(id, cust_id, order_date, fin_code_id NULL(
'''||@null_string||''' ) , region NULL(
'''||@null_string||''' ) , sales_rep) FROM
'''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
1.inp'',
'''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
2.inp'',
'''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
3.inp'',
'''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
4.inp'', '''||
@extract_directory||'DBA_q1_sales_order_1_1_DATA_5.inp'', '''||
@extract_directory||'DBA_q1_sales_order_1_1_DATA_
6.inp'', '''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
7.inp'', '''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
8.inp'' ROW DELIMITED BY ''\n'' QUOTES ON ESCAPES OFF
DEFAULTS OFF FORMAT ASCII IGNORE CONSTRAINT ALL 0
START ROW ID '||@max_row_id;
```

```
CALL IqExecuteCommand( @load_statement );ALTER TABLE
"DBA"."q1_sales_order" MODIFY cust_id NOT
NULL;
ALTER TABLE "DBA"."q1_sales_order" MODIFY order_date
```

```
NOT NULL;
ALTER TABLE "DBA"."q1_sales_order" MODIFY sales_rep NOT
NULL;
```

This example shows query server schema and data that require intervention during migration. Your situation may vary, but you have complete control of the content of the final reload_schema.sql and reload_data.sql files.

Moving Local Stores With iqlsunload
Unload the schema and data for local objects in the 12.7 IQ Local Store, and move them to another node for migration.

Prerequisites
Upgrade to Sybase IQ 12.7 ESD #5 at a minimum before local store migration.

Task

1. Source the ASIQ-12_7.sh or .csh file before running the **iqlsunload** utility.
2. Run the 12.7 **iqlsunload** utility against each query server with a local store.
3. Edit the reload_schema.sql
   • Delete unwanted objects.
   • Change any commented objects in the reload_schema.sql that you want to reload.
   • Add commands to define any objects that you defined in **sp_mpxcfg**_<servername> procedures.
4. Edit the extract_data.sql script to remove objects that you do not want to migrate. These objects are generally the same ones that you removed from reload_schema.sql.
5. Use Interactive SQL to run extract_data.sql from your 12.7 local store.

   You now have unloaded the schema and data for local objects in the 12.7 local store.
6. Run the reload_schema.sql and reload_data.sql scripts against the 12.7 write server.

   Note: If you prefer, wait until the write server has been migrated to 15 and run the reload_schema.sql and reload_data.sql scripts against the new 15 coordinator.

Verify the Legacy Databases Before Migration
Verify the consistency of your Sybase IQ 12.7/12.6 database before you migrate.

1. Start version 12.6 or 12.7 and connect to your database.
2. Issue a **CHECKPOINT** command.

3. Run **sp_iqcheckdb** in verify mode.

```
sp_iqcheckdb 'verify database'
```

If you run the procedure from Interactive SQL, redirect output to a file:

```
sp_iqcheckdb 'verify database' >& filename
```

4. Issue a **COMMIT** statement.

Note: For information on interpreting the **sp_iqcheckdb** results and corrective action, see *System Administration Guide: Volume 1 > System Recovery and Database Repair*. If you need to contact Sybase Technical Support, you must provide the output from **sp_iqcheckdb**.

Start the Multiplex Write Server in Single Node Mode

To ensure a smooth multiplex migration, start the write server in single node mode to clean internal state information. To do this, start the writer node in single node mode, shutdown, start again normally, and shut down.

Start the writer node with the server arguments -**gm 1** and **-iqmpx_sn 1**:

```
-gm 1 -iqmpx_sn 1
```

If you use administrative startup scripts, create a copy of the start_server script to start the write server you want to migrate. For example, copy the existing file start_server to a new file called start_server_single_node.

Suppose that start_server contains this startup command:

```
start_asiq -STARTDIR /work/iq-127/mpx/main @/work/iq-
127/mpx/main/params.cfg -n mpx_main $readonly $nomain -
x tcpip{port=62631} /work/iq-127/mpx/main/main.db
$dbkey
```

Add the two single node startup arguments to change the preceding command as follows in start_server_single_node:

```
start_asiq -STARTDIR /work/iq-127/mpx/main @/work/iq-
127/mpx/main/params.cfg -n mpx_main -gm 1 -iqmpx_sn 1
$readonly $nomain -x tcpip{port=62631} /work/iq-127/
mpx/main/main.db $dbkey
```

There are now two script files, start_server and start_server_single_node to make the server ready for migration:

1. Start the writer node with start_server_single_node.
2. Shutdown the writer node.
3. Start the writer node with start_server.
4. Shut down the writer node.
5. Shut down the Sybase IQ 12.7/12.6 server.

Use iqunload to Migrate Your Database

Run **iqunload** with the appropriate parameters to migrate your databases.

Minimum required parameters for a multiplex writer are **-au** (migrate database), **-c** (connection parameters), **ENG=** connection parameter and **-ms_filename**. The **ENG=** value must match your existing server name used in IQ 12.7 and the **-ms_filename** specifies the new main store for the migrated writer. This path must be the same for all nodes in the muliplex.

For multiplex, there are two differences in the way you will execute the **iqunload** utility:

*   Specify the engine name in the **-c** connection parameters. This is the same name that your `<mpx_dir>/<writer_node>/start_server` script file uses to start the writer node. The **iqunload** utility attempts to start the database server as simplex first. This start requires the name of the server match the naming conventions for the multiplex nodes. Once **iqunload** detects that the server is a multiplex node, it shuts the node down and restarts it with the **-iqmpx_sn 1** option.
*   The name of the new main store must be visible and accessible by all nodes of the multiplex. This is important because the main store file name defaults to `new_system_main.iq`. This path is relative to the catalog database file (.db). Later, when the Sybase IQ 15 multiplex is synchronized, the catalog is replicated to the secondary nodes, formerly known as the query nodes. If you leave the default value for the main store name unchanged, the path remains `new_system_main.iq` and secondary nodes cannot find the shared main store.

For multiplex writers, required arguments are **ENG=**, **DBF=**, and **-ms_filename**:

*   The **ENG=** argument specifies the multiplex main engine name. This is because the **iqunload** utility attempts to start the database to determine if the database is a simplex or multiplex database. If multiplex, the server name is enforced. If you are unsure of the server name, check the administrative script `start_server` in the database directory.
*   The **DBF=** argument must specify the actual path used to create the multiplex. If you are unsure of this, look at the SYSIQFILE table in your 12.7 server to verify the database path.
*   The **-ms_filename** argument specifies the location of the new main store. This path must be visible and accessible by all servers in the multiplex.

For example:

```
iqunload -au -v -c
"uid=DBA;pwd=SQL;dbf=/sunx5prod/users/marshall/mpx127/
w1/w1.db;eng=w1_1234" -ms_filename
../shared/new_main_store.iq
```

```
 Sybase IQ Unload Utility Version 15.2.0.5533
Connecting and initializing
    2008-11-23 22:32:07 Unloading user and group
        definitions
    2008-11-23 22:32:08 Unloading table definitions
    2008-11-23 22:32:09 Unloading index definitions
```

```
    2008-11-23 22:32:09 Unloading functions
    2008-11-23 22:32:09 Unloading view definitions
    2008-11-23 22:32:09 Unloading procedures
    2008-11-23 22:32:09 Unloading triggers
    2008-11-23 22:32:09 Unloading SQL Remote          definitions
    2008-11-23 22:32:09 Unloading MobiLink definitions
    2008-11-23 22:32:10 Creating new database
    2008-11-23 22:32:48 Reloading user and group
       definitions    2008-11-23 22:32:48 Reloading table definitions
    2008-11-23 22:32:53 Reloading index definitions
    2008-11-23 22:32:53 Reloading functions
    2008-11-23 22:32:53 Reloading view definitions
    2008-11-23 22:32:53 Reloading procedures
    2008-11-23 22:32:53 Reloading triggers
    2008-11-23 22:32:53 Reloading SQL Remote
        definitions
    2008-11-23 22:32:53 Reloading MobiLink definitions
```

```
       Successfully backed up file "/sunx5prod/users/marshall/
mpx127/w1/w1.db" by
renaming it to
"/sunx5prod/users/marshall/mpx127/w1/w1.db.before_schema_reload".
Successfully backed up file
"/sunx5prod/users/marshall/mpx127/main.db" by renaming it to
"/sunx5prod/users/marshall/mpx127/main.db.before_schema_reload".
Successfully backed up file
/sunx5prod/users/marshall/mpx127/main.iqmsg" by renaming it to
"/sunx5prod/users/marshall/mpx127/main.iqmsg.before_schema_reload".
Successfully reloaded schema of database
"/sunx5prod/users/marshall/mpx127/main.db".
```

Troubleshooting Multiplex Migration

If you are unable to migrate your multiplex database following the procedures up to this point, try this alternate method.

- Change your Sybase IQ 12.6/12.7 multiplex to a simplex (non-multiplex) database by dropping all query nodes.
- Migrate the database to Sybase IQ 15 by following the steps for simplex databases only.
- Recreate the multiplex by converting the simplex Sybase IQ 15 database to a multiplex, following the steps in "Converting Databases to Multiplex" in *Using Sybase IQ Multiplex*.

Verify the Migrated Multiplex Database

Verify the migrated database in read-only mode and correct any errors.

1. Start the database using the read-only switch, **-iqro 1**. Start the coordinator (the 12.6 or 12.7 write server) using both **-iqro 1** and single node mode, **-iqmpx_sn 1**.

When starting the coordinator in 15.3, use the same port used by the 12.6 or 12.7 writer server.

2. Issue a **CHECKPOINT** command.

_____

3. Run **sp_iqcheckdb** in verify mode:

```
sp_iqcheckdb 'verify database'
```

If you run the procedure from Interactive SQL, redirect the output to a file:

```
dbisql -c "..." "sp_iqcheckdb ('verify database')" >& filename
```

where "..." represents startup parameters for your database.

4. Issue a **COMMIT** statement.

The server is currently in read-only mode, and cannot complete some postmigration tasks. Additionally, the verification reports some problems with Block Count Mismatch, Blocks Leaked, and Unallocated Blocks in Use. All other segments of the verify database should not report any errors.

For example:

```
'** Block Count Mismatch','79','*****'
'** Blocks Leaked','25','*****'
'** Unallocated Blocks in Use','104','*****'
```

Examine the **sp_iqcheckdb** report for errors. If you need to contact Sybase Technical Support, you must provide the output from **sp_iqcheckdb**.

Start and Verify the Coordinator

In the previous section, the multiplex coordinator was started in single node mode (**-iqmpx_sn**) and read-only (**-iqro**) to perform some initial database checks. For coordinators, the server needs to reset an identity cookie before the multiplex can be used.

Once you successfully restart the coordinator with **iqro 1** and **iqmpx_sn 1**, shut it down and restart it without any special switches.

For example:

```
start_iq @params.cfg -n mpx_main -iqmpx_ov 1 -x 'tcpip{port=62631}' /
workserver/work/iq-127/mpx/main.db
```

Manually Synchronize the Secondary Nodes

To start the secondary nodes, install Sybase IQ 15 and then synchronize from the coordinator node. When you migrate a query node, it becomes a reader node.

1. Back up the query node files. Sybase recommends that you at least back up existing catalog .db, catalog .log and iqmsg files.

For example:

```
rename /sunx5prod/users/work/iq-127/mpx/q1/q1.db
/sunx5prod/users/work/iq-127/mpx/q1/q1.db.before_schema_reload
rename /sunx5prod/users/work/iq-127/mpx/q1/q1.log /sunx5prod/
users/work/iq-127/mpx/q1/q1.log.before_schema_reload
rename /sunx5prod/users/work/iq-127/mpx/q1/q1.iqmsg
/sunx5prod/users/work/iq-127/mpx/q1/q1.iqmsg.before_schema_reload
```

Database Upgrades

2. Issue a **dbbackup** command to synchronize servers. You might have a different name for the query node's catalog file, depending on your configuration. In the following example, `q1.db` is the catalog file name on the query node:

```
dbbackup -y -x -c
"uid=dba;pwd=sql;eng=mpx_main;dbf=/sunx5prod/users/
work/iq-127/mpx/main/main.db"
/sunx5prod/users/work/iq-127/mpx/q1
```

```
SQL Anywhere Backup Utility Version 11.0.1.5533 Debug
(702 of 699 pages, 100% complete)
Transaction log truncated
Database backup completed
```

3. If your query nodes do not use a different catalog database name, skip to step 4.

Step 2 synchronized the catalog database file from the coordinator. If you prefer to use the same catalog database file name as the coordinator, adjust any server start and stop administration scripts on the secondary nodes to use the new name.

To retain the same catalog database file names:
- Rename the synchronized coordinator catalog database file name. For example, assuming the coordinator file was called `main.db` and the secondary server was called `q1.db` enter:
```
mv main.db q1.db
```
```
rename main.db q1.db
```
- Rename the log file for the query node. This is necessary as the file renamed above still contains an internal pointer to main.log:
```
dblog -t q1.log q1.db
```

4. Start the secondary server in normal mode:
```
start_iq @params.cfg -n mpx_q1 -x
'tcpip{port=62632}' -o /worksrver/work/
iq-127/mpx/q1/o.out -Z -zr all -zo /worksrver/
iq-127/mpx/q1/zo.out /workserver/work/
iq-127/mpx/q1/main.db
```

The above command line is derived from your existing query server `start_server` administration script.

5. Repeat these steps on the remaining secondary nodes that you want to migrate.

Start the Secondary Nodes
To start the secondary nodes all nodes of the multiplex must be up and running .
Start the secondary servers with the command line startup utility.

For example:
```
start_iq @params.cfg -n <server_name> database_file.db
```

<*server_name*> specifies the secondary server. You can obtain the name from the existing start server administration script. The specified `database_file.db` is the name resulting after you performed the secondary node synchronization steps.

Set the Failover Node
After you migrate the multiplex data, connect to the coordinator, and set the failover node. Use a command like this to set the failover node: `ALTER MULTIPLEX SERVER servername ASSIGN AS FAILOVER SERVER`.

In this command *servername* is one of the secondary nodes.

Postmigration Tasks
Check this list after you migrate you database.

Re-create Indexes for EUC_TAIWAN Data
In Sybase IQ 15, the character encoding specification for the EUC–TAIWAN collation now uses the EUC_TW character set. You must re-create indexes on data in version 12.7 or earlier databases that use the EUC_TAIWAN collation to make them work with Sybase IQ 15.

Update Configuration Files
Compare your existing `params.cfg` files with the new `default.cfg` file created by the installation. The installation does not update or overwrite existing `params.cfg` files. In each `params.cfg` file, update parameter defaults that differ from those in the `default.cfg` file, while maintaining any customized parameter settings appropriate for your system. Add any new start–up parameters in `default.cfg` to your `params.cfg` file. The **-gl** parameter, for example, is required for server start–up in version 12.5 and later.

Preserve Database Options
Sybase IQ preserves the settings of all 12.7 database options that are still valid in Sybase IQ 15 in migrated databases. Check for deprecated features.

Back Up Your Databases After Migration
Back up your databases again with the **BACKUP** statement. If you use the IQ **BACKUP** statement instead of a system–level backup, you can run backups and queries concurrently. For complete syntax, see *System Administration Guide: Volume 1 > Data Backup, Recovery, and Archiving*.

For a multiplex migration, back up only the coordinator only in this manner. For secondary servers, run the **dbbackup** utility from the secondary server directory.

Converting to a New Hardware Platform

You can move a database from one platform to another as long as both platforms share the same endian structure. To move your database to a different platform, move your database first, then migrate your data.

1. Back up the database.
2. Shut down the Sybase IQ server.
3. Install the Sybase IQ server on the new platform. Your migration can take place on the same or a different machine.
4. Start the Sybase IQ server on the new hardware platform.
5. Connect to the utility database, `utility_db`.
6. Restore the database from the backup you created in Step 1.
7. Shut down the server and restart it against the restored database.
8. Start Interactive SQL and issue the database upgrade statement.

   For example:

   ```
   ALTER DATABASE UPGRADE
   ```

Note: If the current version of Sybase IQ is higher than the version on which this database was running you will need to upgrade the database.

Configuration

This section outlines basic configuration concepts and procedures.

Using Configuration Files

Use a configuration file to store server startup options.

The `@data` option allows you to specify environment variables and configuration files on the command line. To specify a configuration file, replace *data* with the path and name of the configuration file:

```
start_iq @configuration_filename.cfg dbname.db
```

If a startup parameter is passed from the command line and the same parameter exists in the `.cfg` file, the value in the command line overrides the value in `.cfg` file.

Configuration files can contain line breaks, and can include any set of options, including the `@data` option. Use a number sign (#) to designate lines as comments. An ampersand (&) character that appears by itself at the end of a line indicates that the previous token is continued on the next line. See the Utility Guide for a complete list of switches.

Default Configuration File

When you start a server and do not include any command line parameters or specify a configuration file, IQ retrieves the start-up parameters from a default configuration file, `default.cfg`, in the `$IQDIR15/scripts` directory. The `default.cfg` file is also the source of parameters for the Service Manager, the `params.cfg` file used by Sybase Central, and for multiplex configurations. You can maintain consistency by editing parameters in `default.cfg`.

To increase the amount of virtual memory, change the parameters in `default.cfg`. To reduce the virtual memory that other programs use, add swap space to machines or reduce IQ memory demands (cache sizes, thread count, stack size, etc.).

Security and Configuration Files

You can use the **dbfhide** (File Hiding) utility to encrypt your configuration files. If you specify log file parameters (**-o** *logfile*) in encrypted files, the log is not available to the IQ Agent or **start_iq**, which cannot display server log information back to the database administrator. You may want to add parameters that do not require encryption on the command line or in a separate configuration file. For example:

```
start_iq @encrypt_params @other_params
```

or

```
% start_iq @encrypt_params -n myserv -c 400 -o $IQDIR15/logfile/
myserv.log
```

See these topics in the SQL Anywhere documentation: *SQL Anywhere 11.0.1 > SQL Anywhere Server - Database Administration*

Configuring Backup Devices

Set the appropriate parameters to set backup devices.

The value for block_size is 0. This tape drive is in variable-length block mode. A non-zero value indicates fixed length block mode, which IQ backup does not support.

See *System Administration Guide Volume: 1 > Data Backup, Recovery, and Archiving* for more information.

Configuring Client Connectivity

Sybase IQ supports ODBC and JDBC connectivity.

Connecting Using ODBC

Open Database Connectivity (ODBC) is a standard API that allows a single application to access a variety of data sources through ODBC–compliant drivers.

ODBC Conformance
ODBC drivers manufactured by different vendors may vary widely in the functions they provide. Sybase IQ supports ODBC 3.5.2.

Table 23. ODBC Conformance Levels.

Conformance level	Sybase IQ support
Core conformance	All core features.
Level 1 conformance	All level 1 features, except for asynchronous execution of ODBC functions. Sybase IQ does support multiple threads sharing a single connection. The requests from the different threads are serialized by Sybase IQ.

Conformance level	Sybase IQ support
Level 2 conformance	All level 2 features, except: • Three-part names of tables and views. This is not applicable for Sybase IQ. • Asynchronous execution of ODBC functions for specified individual statements. • Ability to time out login requests and queries.

Note:

• Although you can use new ODBC 3.5.x features such as descriptors in your ODBC applications, ODBC 2.x applications will continue to work with Sybase IQ.
• For more information about ODBC, see the *ODBC Programmer's Reference*, available from Microsoft Corporation as part of the ODBC software development kit or from the *Microsoft Web site*.

Installing ODBC Drivers

The Sybase IQ 32-bit ODBC driver is installed as part of the Network Client.

Installing Sybase IQ on a UNIX or Linux server also installs an ODBC driver that applications directly access.

Application Linking

If you use an ODBC application that uses `libodbc.so` (`libodbc.so.1`) or `libodbcinst.so`(`libodbcinst.so.1`), you can create symbolic links to that point to `$IQDIR15/lib64/libdbodbc11.so.1` for single-threaded or `$IQDIR15/lib64/libdbodbc11_r.so.1` for multi-threaded applications.

If you are creating an ODBC application, you can link directly to `libdbodbc11.so` for non–threaded applications and `libdbodbc11_r.so` for threaded applications. References to ODBC functions are resolved at run-time. The `so` suffix shown is specific to the Sun Solaris system.

32-bit and 64-bit Drivers

UNIX 64-bit applications, including many third-party tools, can use 64-bit ODBC drivers to connect to 64-bit IQ servers. 32-bit applications can use 32-bit ODBC drivers to connect to 64-bit IQ servers. (32-bit applications cannot use 64-bit ODBC drivers to connect to 64-bit IQ servers.)

The Sybase IQ Network Client CD supplied with all platforms contains an ODBC driver for connecting from Windows-based ODBC applications. To connect via ODBC from 32-bit UNIX client applications, download and install the 32-bit ODBC driver kit. Check the EBF/Update information for the appropriate driver.

Supported ODBC Drivers
For a list of ODBC drivers for supported UNIX-like platforms, see "Linking ODBC applications on Unix" in *SQL Anywhere 11.0.1 > SQL Anywhere Server - Programming > SQL Anywhere Data Access APIs > SQL Anywhere ODBC API > Building ODBC applications.*

Note: Sybase IQ does not support the Mac OS X platform.

Using UNIX or Linux–Based Query Tools Through ODBC
There are several ways to use ODBC to connect to UNIX or Linux-based Query tools.

Tools that do not require extensive driver manager services can use symbolic links to connect directly to the driver. Tools that require a driver manager may ship with a driver manager.

For information and recommendations about compatible driver managers available for specific applications, see the documentation for the application.

On a UNIX-like operating systems, ODBC data sources are stored in .odbc.ini. Use a text editor to update this file manually, or use the cross–platform iqdsn utility to create an ODBC data source.

You can use files as data sources instead of databases. File data sources are stored as files with the extension .dsn.

Managing ODBC Data Sources
On UNIX-like operating systems, ODBC data sources are stored in a system information file. Although this file is named `.odbc.ini` by default, it can have any name.

A system information file can include any connection parameter. This sample DSN connects to the demo database:

```
[My Data Source]
Userid=DBA
Password=sql
EngineName=test_server
CommLinks=tcpip(port=1870)
AutoStop=no
DatabaseName=iqdemo
DatabaseFile=iqdemo.db
```

You can create and manage ODBC data sources with the iqdsn utility. See "Data Source Utility (iqdsn) in the *Utility Guide*.

System Information File Location
Sybase IQ searches the following locations, in order, for the system information file:

1. The $ODBCINI environment variable
2. The $HOME environment variable
3. The user's Home directory

_____

4. The $PATH environment variable

Note: Sybase IQ clients ignore these environment variables when searching for the system information file:

- $ODBC_HOME
- $ODBC_INI

Connecting with a Data Source Entry

To use a data source entry to connect to your database, open a terminal, and use **dbisql** with the data source entry name in a connection string. For example:

```
% dbisql -c "dsn=sample_dsn"
```

Note: Any database started from a connection string is, by default, stopped when there are no more connections to it, and any database that is loaded from a connection string is unloaded as soon as there are no more connections to it when the Autostop connection parameter is set to yes. This does not apply to multiplex IQ databases started with Sybase Central.

Connecting without a Data Source Entry

To connect without using `.odbc.ini`, open a terminal, and use **dbisql** all connection parameters. For example:

```
% dbisql -c "UID=DBA;PWD=sql;AutoStop=no;
    ENG=<engine name>;DBF=$IQDIR15/demo/iqdemo.db"
```

This example is shown on multiple lines, however, you must enter the entire command at the command prompt on one line.

Command Line Connections to 32-bit Applications

Use ISQL to connect to 32-bit applications from the command line.

To connect to a 32-bit application without using a data source, use dbisql to connect from the command line. To connect to the `demo.db` you would use a command similar to this:

```
dbisql -c "UID=DBA;PWD=sql;AUTOSTOP=no;
ENG=<engine name>;DBF=%ALLUSERSPROFILE%\SybaseIQ\
demo\iqdemo.db"
```

Connecting Using JDBC

JDBC and jConnect provide access to Sybase IQ for Java-based applications.

JDBC (Java Database Connectivity) is a Java application programming interface (API) that provides programmatic access to Sybase IQ. jConnect is a Sybase utility that provides high-performance native access to all Sybase products as a standard installation option.

Sybase Central and Interactive SQL (**dbisql**) can use either JDBC or ODBC connectivity.

See the SQL Anywhere documentation: *SQL Anywhere 11.0.1 > SQL Anywhere Server - Programming > SQL Anywhere Data Access APIs > SQL Anywhere JDBC driver*

Note:

- The SQL Anywhere JDBC driver is the default driver for Sybase Central and Interactive SQL.
- IQ Agent can use only jConnect with JDBC to connect to a multiplex server. The jConnect JDBC drivers have been certified with simplex and multiplex servers.

Connecting Using OLE DB

Sybase IQ includes an OLE DB provider as an alternative to ODBC.

OLE DB is a data access model from Microsoft that uses the Component Object Model (COM) interfaces. Unlike ODBC, OLE DB does not assume that the data source uses a SQL query processor. Although OLE DB requires a Windows client, you can use using OLE DB to access Windows and UNIX servers.

For more information, see the *SQL Anywhere Server Programming Guide* and *Sybase IQ System Administration Guide Volume 1 > Database Connections Using OLE DB*.

Note: Sybase IQ OLE DB support differs from SQL Anywhere support:

- Sybase IQ does *not* support Windows CE.
- Sybase IQ does *not* support remote updates through a cursor.
- Sybase IQ supports Dynamic (dynamic scroll), Static (insensitive) and Forward only (no–scroll) cursors, but does *not* support Keyset (scroll) cursors.
- In Sybase IQ the isolation level is always 3, no matter what you specify.

Connecting Using Open Client

Clients require a server object in the interfaces file to access server properties and methods. Use the IQ Directory Services Editor (iqdsedit)to create the interfaces file entries.

Prerequisites

- You must be the owner of the Sybase home directory ($SYBASE) to run `iqdsedit`
- Make a copy of the interfaces file before you make any changes.

Task

1. From a terminal or command line, change to $IQDIR15/bin<platform>

   The *<platform>* value is 64 or 32, depending on your installation.
2. Start **iqdsedit**.
3. On the Directory Service dialog, click OK.
4. From the Server Object menu, choose Add.

---

5. On the Server Name dialog, enter the server name, the click OK.

6. From the Server Entry Editor, type the server name, then click Add new network transport. The server name in your `iqdsedit` entry must be the same as the database name.

7. On the Network Transport Editor, select TCP transport, then enter the host name and port number.

   The defaults are usually sufficient.

8. Click OK to return to the Server Entry Editor.

   The server is now listed under Available Network Transports.

Running the Client and Server On the Same System

Sybase IQ uses a shared memory segment and several semaphores for communication between the client and server on the same machine.

Shared memory is the default communications mechanism when the client and server are on the same system. Shared memory is configured automatically, and starts automatically, on both UNIX and Windows platforms.

Local connections through shared memory are not supported for Solaris x64 systems. Use standard network connectivity by adding the parameters **-host** <*hostname*> and **-port** <*portnumber*> to the client connection string.

See the *Sybase IQ System Administration Guide* for more information on using the **-host** and **-port** parameters.

Network Issues for IQ Servers

Properly configured Sybase IQ UNIX servers run under the TCP/IP protocol, which enables non–UNIX clients to communicate with a UNIX database server.

Verified TCP/IP Protocol Stacks

For Sybase IQ to run properly, the protocol stack on the client and server computers must be compatible at each layer. Many vendors supply TCP/IP protocol stacks and associated software. Sybase IQ communications have been explicitly verified with these TCP/IP implementations:

- TCP/IP For NetWare
- Microsoft Winsock version 2.0

User Datagram Protocol

There are several entries into the TCP/IP protocol stack. Sybase IQ employs the User Datagram Protocol (UDP). While it is called a transport protocol here and elsewhere, UDP provides little more than a user interface to the network layer IP. In particular, UDP is not a guaranteed transmission protocol.

Tuning Sybase IQ Performance Under TCP/IP

Although the default packet size for TCP/IP is 1460 bytes, a larger packet size may improve query response time, especially for queries that transfer a large amount of data between a client and a server process. You can set the maximum packet size using the database server command lines or CommBufferSize (CBSIZE) in the client connection string. This option may be used with the **start_iq** command.

Connecting Across a Firewall

Set the CommLinks connection parameters in your application's connection string to connect across a firewall.

There are restrictions on connections when the client application is on one side of a firewall and the server is on the other. Firewall software filters network packets according to network port. Also, it is common to disallow UDP packets from crossing the firewall.

- Set the **ClientPort** parameter to a range of allowed values for the client application to use. You can then configure your firewall to allow these packets across. You can use the short form **CPort**.
- Set the **HOST** parameter to the host name on which the database server is running. You can use the short form **IP**.
- Specify the port your database server is using in the **ServerPort** parameter if it is not using the default port of 2638. You can use the short form **Port**.
- Set the **DoBroadcast=NONE** parameter to prevent UDP from being used when connecting to the server.

In this example, the connection string fragment:

- Restricts the client application to ports 5050 through 5060
- Connects to a server named `myeng` running on the machine at address `myhost` using the server port 2020

No UDP broadcast is carried out because of the DoBroadcast option.

```
CEng=myeng;Links=tcpip(ClientPort=5050-
5060;Host=myhost;Port=2020;DoBroadcast=NONE)
```

For more information on these parameters, see *System Administration Guide Volume 1 > Connection and Communication Parameters*.

Sybase IQ 15 Login Policy Migration

A Sybase IQ 15.3 multiplex database upgrade changes the way users access multiplex servers. Starting with Sybase IQ 15.3, logical servers provide the only means to access the multiplex server nodes.

Upgrading a multiplex database creates an appropriate logical server for each server-specific login policy. Login policies are updated to use a logical server configuration which provides access to the same set of multiplex servers that they did prior to upgrade.

If a login policy does not allow access to any node (such as when based setting of locked is ON and there are no multiplex server level overrides), the login policy is set to a system defined logical server, NONE, instead of creating a new logical server. NONE indicates that the login policy does not allow access to any multiplex server.

If a login policy does not have any explicit setting for 'locked' option, either at base level or multiplex server level override, then no logical server is created for this policy. Such login policy will inherit the logical server assignment of the root login policy, which provides access to the same set of servers that are accessible from the root login policy.

- Membership configuration of a logical server is done is such a way that it provides access to the same multiplex nodes as the corresponding 15.x login policy. A logical membership of the coordinator is also added to the logical server if login policy allowed access to the current coordinator server.
- Logical server names are derived from the login policy names. If the login policy name is <= 125 characters, the logical server follows this naming convention: **LS_<login policy name>**. For example, for a login policy named **mpx_grp1**, a logical server **LS_mpx_grp1** is created and assigned to that login policy.

  If the login policy name exceeds 125 characters, a logical server is created with the same name as of the login policy i.e. without adding a prefix of **LS_**.
- During the upgrade, some login policy option settings or multiplex server-level overrides are reset or removed. In the root login policy, **locked** and **max_connections** overrides are reset to default values (**locked** 'ON' becomes **locked** 'OFF'; **max_connection** values are reset to '10').

  Settings for **locked** and **max_connections** are removed from user-defined login policies. Mutliplex server level overrides are removed from all login policies.

Note: For more information, see *Using Sybase IQ Multiplex > Logical Servers*.

Login Policy Migration Example

Consider these sample login policies for a 15.x database with multiplex nodes {n1, n2, n3, n4} (assume that {n1} is the coordinator).

Pre-Upgrade Login Policies

Table 24. Root Login Policy

Login Policy Name	General Overrides		Multiplex Server	Multiplex Overrides	
root	locked	OFF	n4	max_connections	15
	max_connections	20			

Table 25. Login Policy 1

Login Policy Name	General Overrides		Multiplex Server	Multiplex Overrides	
mpx_grp1	locked	ON	n1	locked	OFF
			n2	locked	OFF
				max_connections	5

Table 26. Login Policy 2

Login Policy Name	General Overrides		Multiplex Server	Multiplex Overrides	
mpx_grp2	locked	OFF	n1	locked	ON
	max_connections	10			

Table 27. Login Policy 3

Login Policy Name	General Overrides		Multiplex Server	Multiplex Overrides	
mpx_grp3	locked	ON	—	—	

Table 28. Login Policy 4

Login Policy Name	General Overrides		Multiplex Server	Multiplex Overrides	
mpx_grp4	—		n2	locked	ON
			n3	locked	ON

Table 29. Login Policy 5

Login Policy Name	General Overrides		Multiplex Server	Multiplex Overrides
mpx_grp5	max_connections	20	—	—

Post-Upgrade Server Assignments

Table 30. Post-Upgrade Server Assignments

Login Policy Name	Logical Server	Member Servers	Overrides	
root	LS_root	n1, n2, n3, n4, FOR LOGICAL COORDINATOR	locked	OFF
			max_connections	10
mpx_grp1	LS_mpx_grp1	n1, n2, FOR LOGICAL CO-ORDINATOR	—	
mpx_grp2	LS_mpx_grp2	n2, n3, n4	—	
mpx_grp3	NONE	—	—	
mpx_grp4	LS_mpx_grp4	n1, n4, FOR LOGICAL CO-ORDINATOR	—	
mpx_grp5	—	—	—	

Note:

- The Overrides column shows **locked** and **max_connections** options set to default values in the root login policy. Settings and overrides in other login policies were removed during the upgrade.
- Because the **mpx_grp3** login policy did not allow access to any node prior to the upgrade, the conversion assigns a system defined logical server NONE.
- Because the **mpx_grp5** login policy did not have any explicit setting for locked option prior to the conversion, it simply inherits the logical server assignment from the root login

policy. This ensures that **mpx_grp5** has access to the same set of nodes as the root login policy.

Troubleshooting SySAM

If you do not address SySAM 2 related errors in a timely fashion, Sybase IQ may cease to function.

Where to Look for SySAM-related Errors

All SySAM related errors are printed in the Sybase IQ error log and `<dbname>.iqmsg` file. If e-mail alerts are configured, recipients receive an e-mail whenever a SySAM event occurs.

You can always check the current state of SySAM licenses in Sybase IQ using the **sp_iqlmconfig** procedure. The output shows the current SySAM configuration and the status of each license.

If you are using a served license model, the license server's debug log located in the log directory (`$SYBASE/SYSAM-2_0/log`) provides information about any license check-out issues).

SySAM Grace Mode

If Sybase IQ cannot find a required license, it continues to operate in grace mode, if a grace period is available.

The Sybase IQ error log and `<dbname>.iqmsg` file include information about licenses awarded under grace. If e-mail notifications are set up for warning severity, e-mail messages are sent for this event. See "Enabling and changing e-mail notifications" for more information.

The warnings in the Sybase IQ error log, `<dbname>.iqmsg` file, and e-mail message are repeated while the license is under grace.

SySAM Grace Period

Typically, you get a 30-day grace period to resolve any SySAM-related issues. When a license is awarded under grace, entries in Sybase IQ error log, `<dbname>.iqmsg` file, and optional e-mail message include the date and time when the grace period expires.

Sybase IQ disables the functionality corresponding to the license if the issue is not fixed within the grace period. Sybase IQ itself shuts down if the graced license for the *IQ_CORE* license is more than 30 days.

Enabling and Changing E-mail Notifications

You can configure the e-mail notification setup using **sp_iqlmconfig.** Run the **sp_iqlmconfig** procedures without any parameters to view the current status and configuration.

To enable e-mail notifications, set the `smtp host`, `smtp port`, `email sender`, `email recipients`, and `email severity` properties.

To change the severity that triggers e-mails, set the e-mail severity property. Values are: `ERROR`, `WARNING`, `INFORMATIONAL` and `NONE`.

To change the list of e-mail recipients, set the e-mail recipients property to the comma separated list of e-mail addresses. E-mail addresses can include distribution lists, e-mail address for a pager device, and anything that is capable of receiving SMTP messages.

Calling Sybase Technical Support for SySAM-related Issues

Review this list before you call Sybase technical support for SySAM-related issues.

- The Sybase IQ error log located in `$IQDIR15/logfiles/<servername>.stderr`.
- The `<dbname>.iqmsg` file located in the same directory as your IQ database
- If you started Sybase IQ in grace mode – output from the **sp_iqlmconfig** command.
- If Sybase IQ does not start – output from the server's log file(s) in the `$IQDIR15/logfiles/<servername>.nnn.srvlog`.
- The license files in the `$SYBASE/SYSAM-2_0/licenses` directory on the machine running Sybase IQ.
- The value of *SYBASE_LICENSE_FILE* and *LM_LICENSE_FILE* environment variables, if they have been defined.
- Output from **lmutil lmpath -status** command. The `lmutil` binary is located in the `$SYBASE/SYSAM-2_0/bin` directory.
- If you are using a served license model, the licenses in the `$SYBASE/SYSAM-2_0/licenses` directory, and the license server log file in the `$SYBASE/SYSAM-2_0/log` directory.

License Checkout and Verification

SySAM checks feature licensing at various times. Some features are only checked on startup; others after the feature has been added to the database (e.g., passed the threshold for IQ store size, created multiplex writer, etc.); others are triggered by a particular event.

License Checkout
License checkout table describes the events that determine when SySAM checks out different licenses.

License	Initial checkout occurs...	Coordina-tor	Writer	Reader	Node check-ing out	Separate license per node
IQ_SECURITY	On feature access	Yes	Yes	Yes	All nodes	Yes
IQ_MPXNODE	On database startup	Never	Yes	Yes	Secondary nodes only	Yes
IQ_VLDBMGMT	When user creates 3rd dbspace	Yes	Never	Never	Coordina-tor only	No
IQ_LOB	When user creates/accesses LOB/CLOB table	Yes	Yes	Yes	All nodes	Yes
IQ_CORE	On database startup	Yes	Yes	Yes	All nodes	Yes
IQ_UDF	When user accesses UDF	Yes	Yes	Yes	All nodes	Yes

License	Initial checkout occurs...	Coordina-tor	Writer	Reader	Node check-ing out	Separate license per node
IQ_UDA	Only the first time after an IQ server is started; when a **TEXT** index is created/al-tered; or when you ex-ecute a query with **CON-TAINS** predi-cate.					

Note: IQ_SYSTEM_MAIN and one more user dbspace is provided with the server; a third dbspace requires a IQ_VLDGMGMT license.

License Verification

License verification table describes when SySAM verifies different licenses.

License	Consistent checks on startup	Checks on start-up after feature enabled	Checks on fea-ture access
IQ_SECURITY			•
IQ_MPXNODE		•	•
IQ_VLDBMGMT		•	•
IQ_LOB			•
IQ_CORE	•		
IQ_UDA			•
IQ_UDF			•

Note: Changes to licensed features may not appear until the server is restarted. Checked out licences remain in use until you restart the server.

Diagnosing SySAM Issues

Locate specific SySAM problems here.

Cannot Find License Type During the Sybase IQ Installation

Sybase IQ installation program displays only the most commonly used license types. If you are using an optional license type, your license may not be listed by the installation program.

Installer Cannot Find a Suitable License

If you select a served license model, the install program checks for a license with the selected edition and license type. A warning message is displayed if no suitable license is available.

If you are performing a new Sybase IQ installation, Sybase provides a 30 day grace period to resolve licensing issues. When you complete the installation, debug the license check-out failure.

You can determine the date based version of the license by looking at the license checkout message in the Sybase IQ error log and `<dbname>.iqmsg file`; or by examining the license itself.

You can determine the Sybase IQ release date by the date displayed in the version string, cover letter, or SPDC download link. This data is also shown in the error message posted.

You can download an updated license from SPDC, if your support was paid for the date the update was released. If you support was not paid, you are not authorized to install an update.

License Server Executables and Scripts Are Not Installed

The SySAM wrapper script and the executables for the license server (`lmgrd` and `SYBASE`) are not installed or found in the `$SYBASE/SYSAM-2_0/bin` directory.

The license server component is not selected for installation by default. To install the license server, run the installer, choose the Custom install option, and select license server component.

Alternatively, you can download a standalone SySAM license server install program from SPDC.

License Server does Not Start

The most common cause for the license server failing to start is that there are no served licenses in the server. To start the license server, there needs to be at least one served license in the licenses directory.

When you install a new license server, there are no served licenses in the licenses directory and the license server does not start until you activate a license from SPDC and copy it into the licenses directory. Other causes include:

- The license activated from SPDC was an unserved license. Examine the license files. Served licenses always start with a *SERVER* header. If you do not see a line starting with *SERVER* at the top of the license, you are using an unserved license which does not work with the license server.
- If you are using a specific port number for the license, that port number may already be in use. Use **netstat -a** to verify that the port number is free. If not, free the port or use a different port number for the license server.
- The host name recorded in the *SERVER* header does not match the actual host name. The host name is recorded next to the *SERVER* keyword. If it does not match, correct the host name in the license file or set it to *this_host*, which is a keyword that works with any host name.
- The licenses copied may be activated for a different machine. Examine the Host ID next to the host name in the *SERVER* header. This value must match the Host ID for the machine where the license server is running.

License Server Does Not Recognize the License Files

If the license server refuses to serve the licenses, make sure you installed the license on the correct machine.

- The license was activated for a different machine or activated using incorrect Host ID. Examine the Host ID recorded in the license with the Host ID for the machine. If they are mismatched, check-in and regenerate the license with the correct Host ID.
- The license has been altered. Nothing in the license except some designated information can be altered. The license is rejected if you change other information. If the license is being created by typing from a paper copy, verify if there are any errors entering the license. You can also download a new copy of the activated license from SPDC.
- If the Host ID for your platform is based on the Network adapter identification, verify that you have used an ID associated with a valid NIC. IDs associated with loopback adapters are not valid. If the ID associated with a removable network adapter was used, ensure that the adapter is attached to the computer.

Sybase IQ Does Not Start and Generates a License Check-Out Failure Error

If Sybase IQ cannot check out a required license or start in grace mode.

If Sybase IQ cannot check out the required license, it determines whether the license can be issued in grace mode. If a grace period cannot be given, the license is not granted. If the base license for Sybase IQ (*IQ_CORE*) was in grace and the issue is not fixed by the end of the grace period, Sybase IQ fails to start.

For more information, see "Sybase IQ Starts with Graced License." Troubleshooting tips for resolving licenses issued under grace apply to solve this issue as well.

Sybase IQ Starts With Graced License

If Sybase IQ cannot check out a license and starts in grace mode, execute the **sysam diag feature_name** command to check for available licenses.

Execute the **sysam diag feature_name** command from the $SYBASE/SYSAM-2_0/bin directory, where *feature_name* is the SySAM feature name for Sybase IQ or the optional feature that is in grace mode. The feature name is printed in the Sybase IQ error log, <dbname>.iqmsg file, and in the optional e-mail notifications.

If the **sysam diag** command shows no licenses available for check-out, this may be due to one of the following reasons:

Served Models

- Verify that the license server is running and reachable from the machine you are using. Use **sysam status** to verify that you can access the license server. If you cannot access the license server, see if the license server is running.
- If the license server is running, use **sysam status -f feature_name** to determine whether the license server is serving a license for the given feature. If not, obtain the correct licenses from SPDC.
- All licenses on the license server may be in use. If **sysam status -f feature_name** indicates that no licenses are available, obtain additional licenses or shutdown Sybase IQ.

Unserved Models

- The license for the given feature may not be copied in the local licenses directory. Obtain the correct licenses from SPDC and copy them to the local licenses directory.
- The license may have been activated for a different machine or activated with incorrect Host ID. Check-in and reactivate the license with the correct Host ID.

Wrong License

- The available licenses are for a different operating system or architecture. Obtain license for the correct platform from SPDC.
- If you are running in a terminal server environment, unserved licenses cannot be used. Set up served licenses for a terminal server environment.
- The available license may not authorize use of this version of Sybase IQ.
- For information about date based versioning, see "Installing Product Updates, EBFs, and Support Renewal" in the *Sybase Software Asset Management Users Guide*.
- For information about how to update your license, see the Updating your Sybase licenses section in the *Sybase Software Asset Management Users Guide*.

Product Edition or License Type

If the **diag** command shows the license for the given feature is available for check-out, the reason that Sybase IQ cannot check-out the license could be that the product edition or license type does not match the requirement.

- If the edition and license type configuration parameters are set, Sybase IQ uses only the license that matches these settings.
- Run **sp_iqlmconfig** to see the configured values.
- Check the *VENDOR_STRING* attribute of the licenses to make sure a matching license exists. If a matching license does not exist, obtain a suitable license from SPDC or adjust the Sybase IQ configuration.

If Sybase IQ cannot start because the grace period expired, examine the iq.default.lmp file located in the $IQDIR15/Sysam directory under your parent installation directory for values. These values are stored in two lines that start with *PE=* and *LT=*. These lines are present only if the configuration is set.

Sybase IQ Shows Graced Licenses Even After the Issue Has Been Fixed

Sybase IQ performs license checks periodically, and the license status is not immediately updated. It is updated only after the next heartbeat cycle is completed. This may take a few hours.

Sybase IQ Cannot Find License For An Optional Feature, Even Though the License Exists

Execute **sysam diag feature_name** to ensure that the license for the optional feature exists and can be checked out from the machine where Sybase IQ is running.

If the feature exists, but cannot be checked out from Sybase IQ, this could be because:

- The optional feature is for a different edition.
- The Active or Standby flags do not match for base Sybase IQ and the optional feature.

Execute **sp_iqlmconfig** to determine the edition and active and standby configuration value. The edition and active and standby values are displayed in the Property Name / Property Value table. The edition value is displayed in the *PE* row and active and standby value is displayed in the *AS* row.

Compare the values of *PE* and *LT* for Sybase IQ with the same values recorded in the *VENDOR_STRING* and *ISSUER* fields of the optional feature license. If these values do not match, call technical support.

Sybase IQ Does Not Start With the Expected Edition or License Type

Set the edition and license type configuration parameters to start IQ with the appropriate license.

If the edition and license type configuration parameters are not set, Sybase IQ uses the first available *IQ_CORE* license. If you have multiple *IQ_CORE* licenses with different editions and license types, which license is the first available depends on many factors such as license file directory sort order, available licenses on the license server, and so on.

Because the edition and license type cannot be guaranteed, Sybase recommends that you set the edition and license type configuration parameters. When these parameters are set, you are assured that Sybase IQ starts with that configuration. If a matching license were to be unavailable, Sybase IQ starts under grace (if available) to allow you to resolve the licensing issue.

Sybase IQ Does Not Pick the Correct License

Sybase IQ looks for a license in several locations, and uses the first one that matches the feature, version, edtion and license type.

- Any saved values in the registry on Windows platforms, or the `.flexlmrc` file on UNIX platforms. Use **lmutil lmpath -status** to view the saved values.
- The values set for *SYBASE_LICENSE_FILE* and *LM_LICENSE_FILE* environment variables.
- All license files (any file with a `.lic` extension) in the licenses directory under the `SYSAM-2_0` directory.
- All of the above locations are searched in the order specified until a suitable license is found. When a directory is specified, license files in that directory are loaded in the directory sort order.
- The first license that matches the feature name, version, edition, and license type filter is used. This license may not be what you were expecting to use. Adjust the above locations to control this.

Index