SYBASE®

Installation and Configuration Guide

Sybase IQ

15.0

[LINUX]

Contents

About This Book

Subject

The Sybase® IQ 15.0 *Sybase IQ Installation and Configuration Guide for Linux* provides information you need to install Sybase IQ. This book includes:

- Installation procedures for Sybase IQ Server and Client components.

- Instructions on how to install Sybase IQ with SySAM 2.

- Migration procedures for legacy databases.

Audience

This book is intended for Sybase System Administrators and other qualified personnel who are familiar with their system's environment, networks, disk resources, and media devices.

Related documents

The Sybase IQ 15.0 documentation set includes the following documents:

- *Sybase IQ Error Messages* – Lists Sybase IQ error messages referenced by Sybase error code, SQLCode, and SQLState, and SQL preprocessor errors and warnings.

- *Sybase IQ Installation and Configuration Guide* – Provides platform-specific instructions on installing, migrating to a new version, and configuring Sybase IQ for a particular platform.

- *New Features Guide* – Documents new features and behavior changes for this version.

- *Quick Start* – Lists steps to build and query the demo database supplied with Sybase IQ.

- *Sybase IQ Release Bulletin* – Provides information about last-minute changes to the product and documentation. Read for help if you encounter a problem.

Refer back to the 12.7 documentation set for syntax and functionality that is unchanged. To access the Infocenter Web site, go to Sybooks Online Help at http://infocenter.sybase.com/help/index.jsp.

The Sybase IQ 12.7 documentation set includes the following documents:

- *Encrypted Columns in Sybase IQ* – Covers the use of user encrypted columns within the Sybase IQ data repository. You need a separate license to install this product option.

- *Introduction to Sybase IQ* – Includes hands-on exercises for those unfamiliar with Sybase IQ or with the Sybase Central™ database management tool.

- *Large Objects Management in Sybase IQ* – Explains storage and retrieval of Binary Large Objects (BLOBs) and Character Large Objects (CLOBs) within the Sybase IQ data repository. You need a separate license to install this product option.

- *Sybase IQ Performance and Tuning Guide* – Describes query optimization, design, and tuning issues for very large databases.

- *Reference Manual* – Describes stored procedures, data types, system tables, and the SQL statements and options that Sybase IQ supports.

- *System Administration Guide* – Covers administration issues such as database creation and load operations, data security and integrity, server start-up and connection, and multiplex operations.

- *Sybase IQ Troubleshooting and Recovery Guide* – Read to solve problems and perform system recovery and database repair.

- *Sybase IQ Utility Guide* – Provides Sybase IQ utility program reference material, such as available syntax, parameters, and options.

Sybase IQ and SQL Anywhere

Because Sybase IQ is an extension of SQL Anywhere Studio®, a component of SQL Anywhere Studio, Sybase IQ supports many of the same features as SQL Anywhere. The IQ documentation set refers you to SQL Anywhere Studio documentation where appropriate.

Documentation for SQL Anywhere:

- *SQL Anywhere Server – Database Administration* describes how to run, manage, and configure SQL Anywhere databases. It describes database connections, the database server, database files, backup procedures, security, high availability, and replication with Replication Server, as well as administration utilities and options.

- *SQL Anywhere Server – Programming* describes how to build and deploy database applications using the C, C++, Java, PHP, Perl, Python, and .NET programming languages such as Visual Basic and Visual C#. A variety of programming interfaces such as ADO.NET and ODBC are described.

- *SQL Anywhere Server – SQL Reference* provides reference information for system procedures, and the catalog (system tables and views). It also provides an explanation of the SQL Anywhere implementation of the SQL language (search conditions, syntax, data types, and functions).

You can also refer to the SQL Anywhere documentation in the SQL Anywhere Studio 11.0 collection at Product Manuals at http://www.sybase.com/support/manuals/ and in DocCommentXchange at http://dcx.sybase.com/dcx_home.php.

Documentation for Sybase Software Asset Management (SySAM):

- *Sybase Software Asset Management (SySAM) 2* 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.

Other sources of information

Use the Sybase Getting Started CD, the SyBooks™ CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.

- Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

To access the Sybase Product Manuals Web site, go to Product Manuals at
http://www.sybase.com/support/manuals/.

- Infocenter is an online version of SyBooks that you can view using a
  standard Web browser. To access the Infocenter Web site, go to Sybooks
  Online Help at http://infocenter.sybase.com/help/index.jsp.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest product certifications**

1 Point your Web browser to Technical Documents at
  http://certification.sybase.com/ucr/search.do.

2 Either select the product family and product under Search by Base
  Product; or select the platform and product under Search by Platform.

3 Select Search to display the availability and certification report for the
  selection.

❖ **Creating a personalized view of the Sybase Web site (including support pages)**

Set up a MySybase profile. MySybase is a free service that allows you to create
a personalized view of Sybase Web pages.

1 Point your Web browser to Technical Documents at
  http://www.sybase.com/support/techdocs/.

2 Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ **Finding the latest information on EBFs and software maintenance**

1 Point your Web browser to the Sybase Support Page at
  http://www.sybase.com/support.

2 Select EBFs/Maintenance. If prompted, enter your MySybase user name
  and password.

3 Select a product.

4 Specify a time frame and click Go. A list of EBF/Maintenance releases is
  displayed.

Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the "Technical Support Contact" role to your MySybase profile.

5   Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Syntax conventions   This documentation uses the following syntax conventions in syntax descriptions:

• **Keywords**   SQL keywords are shown in UPPERCASE. However, SQL keywords are case insensitive, so you can enter keywords in any case you wish; SELECT, Select, and select are equivalent.

• **Placeholders**   Items that must be replaced with appropriate identifiers or expressions are shown in *italics*.

• **Continuation**   Lines beginning with an ellipsis (...) are a continuation of the statements from the previous line.

• **Repeating items**   Lists of repeating items are shown with an element of the list followed by an ellipsis (...). One or more list elements are allowed. If multiple elements are specified, they must be separated by commas.

• **Optional portions**   Optional portions of a statement are enclosed by square brackets. For example:

```
RELEASE SAVEPOINT [ savepoint-name ]
```

The square brackets indicate that the *savepoint-name* is optional. The brackets should not be typed.

• **Options**   When none or only one of a list of items must be chosen, the items are separated by vertical bars and the list enclosed in square brackets. For example:

```
[ ASC | DESC ]
```

The square brackets indicate that you can choose ASC, DESC, or neither. The brackets should not be typed.

• **Alternatives**   When precisely one of the options must be chosen, the alternatives are enclosed in curly braces. For example:

```
QUOTES { ON | OFF }
```

The curly braces indicate that either ON or OFF must be provided. The braces should not be typed.

Typographic conventions

Table 1 lists the typographic conventions used in this documentation.

Table 1: Typographic conventions

Item	Description
`Code`	SQL and program code is displayed in a mono-spaced (fixed-width) font.
`User entry`	Text entered by the user is shown in bold serif type.
emphasis	Emphasized words are shown in italic.
file names	File names are shown in italic.
database objects	Names of database objects, such as tables and procedures, are shown in bold, sans serif type in print, and in italic online.

The demo database

Sybase IQ includes scripts to create a demo database.

The demo database represents a small company. It contains internal information about the company (employees, departments, and financial data), as well as product information (products), sales information (sales orders, customers, and contacts), and financial information (fin_code, fin_data).

To create the demo database, run the file *$IQDIR15/demo/mkiqdemo.sh* on UNIX or *%ALLUSERSPROFILE%\SybaseIQ\demo\mkiqdemo.bat* on Windows. The sample database is created in a file named *iqdemo.db*.

Accessibility features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Sybase IQ 15.0 and the HTML documentation have been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.For information about accessibility support in the Sybase IQ plug-in for Sybase Central, see the plug-in online help, which you can navigate using a screen reader.

Configuring your accessibility tool
You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool and see the *Introduction to Sybase IQ* for information on using screen readers.

For information about how Sybase supports accessibility, see Sybase Accessibility at http://www.sybase.com/accessibility. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

For a Section 508 compliance statement for Sybase IQ, go to Sybase Accessibility at http://www.sybase.com/products/accessibility.

If you need help   Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

C H A P T E R 1   **Preparing for Installation**

About this chapter   This chapter describes preinstallation requirements and considerations.

Contents

Overview

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 Adaptive Server Anywhere. Component Integration Services embedded within Sybase IQ provide direct access to relational and non-relational databases on mainframe, UNIX, or Windows servers.

Product editions

Sybase IQ is available in the following editions:

- Enterprise Edition
- Small Business Edition
- Single Application Server Edition

Platforms and operating systems

Sybase IQ analytics server is compatible with the following platforms and operating systems:

Server
- Red Hat Enterprise Linux 5.2 (Linux on POWER; 64-bit) – minimum patch level Update-2
- Red Hat Enterprise Linux 5.2 x86-64, Advanced Server and Workstation Editions – minimum patch level Update-2
- SuSE Linux Enterprise Server (SLES) 10.0 x86-64 – minimum patch level SP1
- SuSE Linux Enterprise Server (SLES) 10.0 (Linux on POWER; 64-bit) – minimum patch level SP1

Client
- Red Hat Enterprise Linux 5.2 x86, Advanced Server and Workstation Editions (Sybase IQ Client only) – minimum patch level Update-2
- SuSE Linux Enterprise Server (SLES) 10.0 (32-bit systems; Sybase IQ Client only) – minimum patch level SP1

Network Client supported platforms

Sybase IQ 15.0 provides a Network Client for all supported server platforms and includes the platform-specific network client with each server purchased.

The Sybase IQ Network Client for Linux on 32-bit systems can be installed on Red Hat Enterprise Linux 5.2 Advanced Server or Workstation Edition. For all other platforms, the Sybase IQ Network Client is supported on the same platforms as the Sybase IQ server.

Sybase IQ Network Client for Windows includes the 32-bit ODBC Windows drivers. Windows 64-bit ODBC drivers allow you to connect to Sybase IQ from third-party packaged 64-bit Windows applications and 64-bit Windows applications written in C++.

The Sybase IQ installation for UNIX and Linux platforms includes ODBC drivers needed for the client and dbisql.

Updated software components

Sybase IQ Server 15.0 installs the following software components, updated to the latest versions available at release time:

• Sybase jConnect™ JDBC Driver version 6.0. This component is optional but strongly recommended. If you install the jConnect driver, Java classes installed into a database can make JDBC calls to access and modify data.

• Java Runtime Environment (JRE) version 6.0 to use the latest Sybase Central plug-in for Sybase IQ.

Licensing requirements

Sybase IQ is available in Evaluation and Licensed Editions.

Installing an unlicensed Evaluation Edition of Sybase IQ 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 Sybase Software Asset Management (SySAM) license.

To install and configure any product edition and any optional feature, you should know which features you are licensed to use. See "Licensing your software" on page 23 for information about obtaining and installing your license.

Installation media

Sybase IQ includes installation media for server and client installations.

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

Planning your installation

How you install Sybase IQ 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.

Installing Sybase IQ for the first time

If you are installing Sybase IQ for the first time, not upgrading from a previous edition, follow these steps:

- 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.

- 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. See "Pre-installation tasks" on page 8 for specific information.

- Determine the license model you want to use. SySAM supports served and unserved license models. See Chapter 2, "Choosing a License Model" in the *Sybase Software Asset Management Users Guide* to determine which model is best suited for your environment.

- Use the product and licensing information from the e-mail or web key you received from Sybase to logon 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. See "Installing Sybase IQ Server" on page 13 and "Testing installed products" on page 35 for additional information.

- Install Sybase IQ and configure your clients See "Installing Sybase IQ Client Components" on page 43 and "Configuring Sybase IQ" on page 97 for more information.

Upgrading from a previous version

- Upgrade Sybase IQ, if necessary. If you are migrating from a version prior to 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 12.7 ESD 5.

- Review and understand the migration utilities before you begin your migration. Sybase provides utilities that you can use to migrate your 12.x data to Sybase IQ 15.0. Understanding these utilities before you begin can help you plan your migration more effectively. See "Migration utilities" on page 53 for more information.

- Follow the work flows for the type of databases you want to migrate. There are separate migration paths for simplex and multiplex databases. See "Migrating simplex databases" on page 72 and "Migrating multiplex databases" on page 76 for additional information.

Changing hardware platforms

If you are upgrading from a previous version and want to move your Sybase IQ databases to a different supported hardware platform, perform the conversion before you upgrade your data to Sybase IQ 15.0.

- Use the procedure "Migrating databases to a 64–bit machine" in the *Sybase IQ Installation and Configuration Guide* for Sybase IQ 12.7 to convert the database to the 64–bit hardware platform.

- For information about migrating a database from one platform to another with the same endian structure, see "Converting to a new hardware platform" on page 51 for specific steps.

- After you change upgrade your hardware platform, follow the appropriate workflow in "Migrating Data" on page 49 to migrate your databases.

Notes

- If you created your Sybase IQ database on a Linux 32–bit version prior to Sybase IQ 12.6 ESD #2, you must 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 higher releases support migration between Windows and Linux.

Disk space and memory requirements

Use the following tables to plan your disk space and memory requirements. Server and client components are listed separately. Typical installations share some common components and generally require less disk space than the totals indicate.

Table 1-1: Sybase IQ AMD64 Server Requirements

Component	Feature	Disk Space	RAM
Server	Sybase IQ	719 MB	-- --
	Sybase Central	183 MB	-- --
	jConnect	100 MB	-- --
	SySAM License Server	154 MB	-- --
	Sybase Software Asset Management Utilities	149 MB	-- --
	Total	1305 MB	1 GB

Table 1-2: Sybase IQ LinuxIBM64 Server Requirements

Component	Feature	Disk Space	RAM
Server	Sybase IQ	893 MB	-- --
	Sybase Central	364 MB	-- --
	jConnect	193 MB	-- --
	SySAM License Server	248 MB	-- --
	Sybase Software Asset Management Utilities	242 MB	-- --
	Total	1940 MB	1 GB

Table 1-3: Sybase IQ IBM64 Client Requirements

Component	Feature	Disk Space	RAM
Client	Sybase IQ	415 MB	-- --
	Sybase Central Java Edition	364 MB	-- --
	jConnect	193 MB	-- --
	SySAM License Server	248 MB	-- --
	Sybase Software Asset Management Utilities	242 MB	-- --
	Total	1462 MB	1 GB

Table 1-4: Sybase IQ AMD64 Client Requirements

Component	Feature	Disk Space	RAM
Client	Sybase IQ	312 MB	-- --
	Sybase Central Java Edition	226 MB	-- --
	jConnect	123 MB	-- --
	SySAM License Server	176 MB	-- --
	Sybase Software Asset Management Utilities	172 MB	-- --
	Total	1009 MB	1 GB

Note Servers require additional disk space for data storage. See *Chapter 3, "Working with Database Objects"* in the *Sybase IQ System Administration Guide* for information about estimating the disk space you need for your IQ data.

Pre-installation tasks

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

Check for operating system patches

Apply any operating system patches before you install Sybase IQ. Check with your operating system vendor for information on the latest operating system patches.

Note Do not use a patch that is earlier than the version suggested for your operating system. Use the patch recommended by the operating system vendor, even if it supersedes the patch listed under "Platforms and operating systems" on page 2.

Check the available disk space and memory

Check the available disk space and RAM against the "Disk space and memory requirements" on page 6 to ensure that you have sufficient resources to install and run Sybase IQ.

Increase the swap space

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. See the chapter "Managing System Resources" in the *Sybase IQ Performance and Tuning Guide* for more information about buffers.

Depending upon the load on the system where the Sybase IQ executable is running, 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.

Resolve potential name conflicts

Red Hat Linux includes a unixodbc package that can cause name conflicts with components provided with Sybase IQ. To avoid name conflicts:

• Rename /usr/lib/libodbc.so to /usr/lib/libodbc.orig.so.

• Rename /usr/lib/libodbc.so.1 to /usr/lib/libodbc.orig.so.1.

Set the kernel parameters

Sybase recommends that you set the number of available semaphore identifiers to at least 4096. Use this command to display the current values of the semaphore kernel parameters:

```
cat /proc/sys/kernel/sem
250 32000 32 4096
```

The four values displayed are:

• SEMMSL — Maximum number of semaphores per set

• SEMMNS — Maximum number of semaphores system–wide

• SEMOPM — Maximum number of operations allowed for one semop call

• SEMMNI — Maximum number of semaphore identifiers (sets)

To adjust these values, edit the */etc/sysctl.conf* file. Setting values in this file saves them for future reboots. For example, to increase the maximum number of semaphore identifiers to 4096, add the following line:

```
kernel.sem = 250 32000 32 4096
```

Make sure that you leave spaces around the equals sign.

To make a change that only lasts until the next reboot, use this command:

```
echo 250 32000 32 4096 > /proc/sys/kernel/sem
```

Install the required package groups

Linux x86-64 requires 32-bit libraries to run client applications, such as Sybase Central. Before you install Sybase IQ on Linux x64-86, you must also install the following:

• *libncurses.so.5*

To find the ncurses version on your installation, type rpm -q ncurses at a shell prompt.

- *compat-libstdc++-296*

    To find the compatibility version on your installation, type rpm -q -a | grep compat at a shell prompt.

Install the required runtime libraries for Linux on POWER

Sybase IQ is compiled and linked with the Visual Age compiler, xlc version 7.0. Before installing Adaptive Server, the runtime libraries for the Visual Age compiler must be installed on your system.

❖ **Installing Visual Age xlc compiler runtime libraries**

1   Check to see if the Visual Age runtime packages have been installed previously, by running:

        rpm –q vacpp.rte-7.0.0

    If `vacpp.rte-7.0.0-0` is returned, the Visual Age runtime packages have been installed and you can skip the following steps.

    If `package vacpp.rte-7.0.0-0 is not installed` is returned, the Visual Age packages have not been installed, and you must install them.

2   To install XLC V7.0 libraries, go to the IBM Web site at http://www-1.ibm.com/support/docview.wss?uid=swg24007906

    Follow the instructions to download and install the runtime libraries on your operating system. Select the package appropriate for your Linux distribution.

3   You can proceed with installing Sybase IQ on Linux on POWER.

Update the C++ Runtime environment on IBM P series machines

If you run RedHat 5.x or SuSE 10 an IBM P Series machine, you must install the XL C/C++ Advanced Edition for Linux, V9.0 environment or the Sybase IQ Sybase Central Plugin will fail to register. This update is available from IBM update site for C and C++ compilers at http://www-01.ibm.com/support/docview.wss?rs=2030&context=SSJT9L&q1=Support+C+and+C%2b%2b+compilers&uid=swg21110831&loc=en_US&cs=utf-8&lang=en.

Verify network functionality

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.

Installing your software

The Sybase IQ installation media includes all of the components you need to install your software. There are interactive and silent install options available.

Interactive installs   The interactive installer is a standard installation wizard that lets you set up your software in a graphically-based environment. The interactive installer supports all product editions and installation types.

Silent installs   Silent install scripts perform standard IQ installations without the need to interact with the product. These scripts are configured to perform typical server installations by default; custom installs require additional configuration. See "Silent Installations" on page 109 for more information.

Installation types   Typical and custom installation types are available.Typical installations install features suitable for most production environments. Custom installs let you choose the components you want to install. You can select group packages, individual items, or a combination of the two.

CHAPTER 2 **Installing Sybase IQ Server**

About this chapter   This chapter provides instructions for installing Sybase IQ for operation
as a network server.

Contents

Overview

Server installation is part of a planned installation sequence. Follow the installation path you chose in "Planning your installation" on page 4. Read this section for information you should know before you begin your server installation:

Before you install Sybase IQ server

- Read the Sybase IQ Release Bulletin for the latest information. See the special installation instructions section for any installation instructions not included in this document.

- Complete all pre-installation tasks. Apply any required operating system patches, check to see that there is sufficient disk space and memory, and make any necessary configuration changes. See "Pre-installation tasks" on page 8 for specific information.

- Generate your license key and download your license to a temporary location on the local file system. If you chose a served license model, install and start the new license server, or know which license server you plan to use. See "Licensing your software" on page 23 for details.

- Know the product edition you intend to install and license you intend to use. If you plan to use a license server, you need to know the license server's Host ID, host name, and port number.

- Determine the target installation directory. The default installation directory is */opt/sybase* . You can choose an alternate location during installation.

Note Instructions in this chapter are specific to the graphically-based interactive installer, see "Silent Installations" on page 109 for silent installation setup instructions.

Server components

Server components are grouped by feature. In this table, items marked as default are installed in a typical installation.

Feature	Components	Default	Notes
Sybase IQ	Sybase IQ Server	•	Installs server software, client tools, and ODBC drivers.
	Sybase IQ Client	•	
	Sybase IQ ODBC	•	
Sybase Central	Sybase Central Viewer	•	Viewer and plug–in are required for Sybase Central.
	Sybase Central Plug–in	•	
jConnect		•	Required for JDBC connections.
Sybase Software Asset Management	Network License Server		Network license server utilities, and FLEXnet Licensing utilities
	License Utilities	•	

Plug–in compatibility

Sybase Central is a graphically-based administration console for Sybase products. The Sybase IQ Server 15.0 plugin for Sybase Central is NOT compatible with previous versions. You can only use the 15.0 plugin with 15.0 servers; you cannot use it to connect to 12.x servers.

Table 2-1: Sybase IQ plug–in compatibility

IQ Plug–in	Sybase IQ	Sybase Central Toolkit	Agent	JRE
15.0	Compatible only with 15.0 servers. You cannot use the 15.0 plug-in to connect to 12.x servers.	6.0 or later required	Agent – 15.0 agent required	JRE – 6.0 required
12.7	12.5 and higher 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 server installation installs server components to separate directories on the host machine.

Parent installation directory

The default installation directory is */opt/sybase* . You can, however, specify another target directory during installation if you use the interactive GUI installer, or by changing the silent installation script if you perform a silent install.

The parent installation directory contains subdirectories, scripts, and property files required to support Sybase IQ. This document uses the variable *$SYBASE* to refer to the parent directory. A list of the major subdirectories you are likely to use during installation appears in the table below.

Directory	Contents
charsets	Provides character sets available for use with Adaptive Server.
collate	Provides the collation sequences available for use with Adaptive Server.
config	Contains various configuration files.
IQ-15_0	Contains subdirectories, scripts, and property files required to run Sybase IQ server.
jConnect-6_0	Provides the Java Database Connectivity (JDBC) driver for Sybase IQ.
locales	Provides language modules for system messages and date/time formats.
shared	Contains components and libraries that are shared by several other components, including Sybase Central 60 and Java Runtime Environment (JRE).
SYSAM-2_0	Location of the SySAM licenses and utilities. This directory contains two subdirectories: bin and licenses.
uninstall	Contains subdirectories for scripts that uninstall IQ and SySAM.

Sybase IQ installation
directory

The Sybase IQ installation directory is a child of *$SYBASE* directory called *IQ-15_0*.

This directory contains subdirectories, scripts, and property files required to run Sybase IQ server. This document uses the variable *$IQDIR15* to refer to the Sybase IQ installation directory. A list of the major subdirectories you are likely to use in this directory during installation appears in the table below.

Directory	Contents
bin*<platform>*	Utilities you use to start and manage your server. *<platform>* value is 64 or 32 depending on your installation.
charsets	Character encoding files.
demo	Sample data and scripts for the sample 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	Various logfiles. Sybase IQ writes log files to this directory.
lsunload	Local Store Unload utility.
samples	Sample SQL Anywhere tools and scripts.
scripts	Sample scripts and stored procedures.
sdk	Subdirectories for various language utilities.
shared	Subdirectories for various shared components.
Sysam	IQ 15.0 local sysam configuration information.
unload	Various unload scripts, files and utilities.

Installing Server components

The following list summarizes the tasks that are required to install the Server components.

- Mount the DVD and set up the sybase account. If your DVD drive is set up for automount, skip this step. If you need to mount the DVD drive, see "Mounting the installation media" on page 18.

- Create a sybase user account to provide consistent ownership and privileges for all installation, configuration, and upgrade tasks.

If a sybase account already exists, skip this step, but verify that you can use this account to log in the system. For more information, see "Setting up the sybase account" on page 19.

• Set the *$SYBASE* environmental variable to directory where you plan to install Sybase IQ. See "Setting the $SYBASE environmental variable" on page 19 for more information.

• Run the installer from the Sybase IQ Server Installation DVD. Be sure to complete all pre-installation tasks before you run the installer. See "Running an interactive install" on page 19 for more information.

• Set the environment variables. Installing IQ also installs some environment files that set variables that are required to run Sybase IQ Server. See "Setting environment variables" on page 21 for more information.

Mounting the installation media

If your DVD drive has been set up for auto–mount, get the absolute path name for this device from the System Administrator, change to this directory, and skip to the procedure titled "Setting up the sybase account" on page 19. Otherwise follow these steps.

❖ **Mounting the product DVD**

1 Log on as a user with root privileges.

2 Place the DVD into the DVD drive.

3 Create a subdirectory of the root directory where you will mount the DVD, if one does not already exist. For example:

```
% cd /
% mkdir /dvd
```

4 If your system does not auto–mount, mount the DVD with a command like the following:

```
% mount  -o ro  /dev/dvd  /dvd
```

5 Log out as root and type exit.

Setting up the *sybase* account

Create a sybase account to provide consistent ownership and privileges for all IQ installation, configuration, and upgrade tasks. This account is sometimes called the sybase user account.

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.

Setting the *$SYBASE* environmental variable

During installation, the installer prompts you to identify the directory where you want to install Sybase IQ. The default installation directory is */opt/sybase*.

Use the following formats to set the *$SYBASE* environment variable to the path of the directory where you plan to install Sybase IQ. In these examples, the SYBASE installation directory is */work/sybase*.

C shell

For the C shell (csh), add this line to the *.cshrc* file:

```
% setenv SYBASE /work/sybase
```

Korn (ksh), Bash, or Bourne (sh) shells

For the Korn (ksh), Bash or Bourne (sh) shells, add this line to the *.profile* file:

```
% SYBASE=/work/sybase;
% export SYBASE
```

Running an interactive install

The Sybase IQ Server Installation media includes all of the components you need to install an IQ Server. Before you run the installer, see the sections on "Mounting the installation media" on page 18 and "Setting up the sybase account" on page 19.

❖ **Installing Sybase IQ Server components**

1   Stop any servers currently running in the install directory before you run the installer.

2   Insert the Sybase IQ Product DVD into the drive.

If the Sybase IQ installer does not start automatically, mount the drive, then use this command to start the appropriate installer:

```
% <DVD_mount_directory>/setup<platform>
```

where *<Platform>* is the name of your operating system. For example, use the command <DVD_mount_directory>setupLinuxAMD64 on AMD 64-bit platforms.

3   On the Welcome screen, click Next.

4   Choose an installation directory, click Next.

5   Choose the product edition you want to configure, click Next.

6   On the next screen, choose a location, then read and accept the license agreement.

You must accept an appropriate license agreement to install Sybase IQ. You can review all available license agreements on the Sybase Web site at http://www.sybase.com/softwarelicenses.

7   When the installer prompts you for the Sybase IQ 15.0 license key(s), do one of the following:

- Use an editor to open the license you downloaded from the SPDC site, and paste the license in the license key text box. Perform this action only if you are installing an unserved license.

- Click Browse, navigate to the location where you stored you license, and, choose your license. Perform this action only if you are installing an unserved license.

- Click Use previously deployed license server, then specify the server's *Host Name* and *Port Number*. Use this option if you want to use a previously deployed license server.

- Click Continue installation without a license key. Use this option to install Sybase IQ in evaluation mode.

8   Click Next, then choose the appropriate Product Edition and License Type.

9   Follow the instructions on the screen to complete your installation. Unmount and remove the installation media from the drive, if necessary.

Notes

- If you decide to install Sybase IQ in grace mode, you can install the product with out a license, and activate a 30 grace period. You must obtain an appropriate license from Sybase before the grace period ends.

- If you plan to obtain your license from the License Server, you must identify the server's host name and port number. If you do not specify a port number, SySAM software communicates over the first available port in the 27000 to 27009 range.

- You can configure SySAM to send e-mail alerts when an event occurs that may need administrative attention. To configure the e-mail alerts, you must identify the SMTP server, SMTP port number, and e-mail recipients.

Setting environment variables

You must set the following environmental variables to run Sybase IQ Server. The environment files that set Sybase IQ variables also set the variables needed by Open Client.

C shell

For the tcsh or C (csh) shell, use this command:

```
% source $SYBASE/IQ-15_0/IQ-15_0.csh
```

Korn (ksh), Bash, or Bourne (sh) shells

For the Korn (ksh) , Bash, or Bourne (sh) shells, use this command:

```
% . $SYBASE/IQ-15_0/IQ-15_0.sh
```

Note These procedures set the *$IQDIR15* environment variable used to start and test Sybase IQ. To test Sybase IQ, see "Running Sybase IQ Server" on page 37.

Post-installation tasks

- Check the Sybase Support Page at http://www.sybase.com/support for software updates. This site provides access to product updates and a variety of downloads to help you get the most out of your Sybase products and solutions.

- If you are migrating from a previous version, see "Migrating Data" on page 49 for information about migrating 12.x databases to Sybase IQ 15.0.

- Make sure that all raw devices have read and write permissions and */dev/rawctl* has read permission for Sybase IQ before you create a database or add a dbspace.

   Raw device names on Linux use the following format: /dev/raw/r***devname***. For example: `/dev/raw/raw10`.

- See "Testing installed products" on page 35 for information about starting your server, running queries with Interactive SQL, and working with the demo database.

- See "Installing Sybase IQ Client Components" on page 43 for information about installing Sybase IQ Client tools.

- See *Introduction to Sybase IQ* for hands-on exercises if you are unfamiliar with Sybase IQ and the Sybase Central management console.

Uninstalling Sybase IQ server

Use these instructions to uninstall Sybase IQ server.

❖ **Uninstalling server components**

Use these instructions to start the Sybase IQ uninstaller.

- From a command prompt, change to *$SYBASE/uninistall/IQ*, and type IQuninstall. Follow the instructions on the screen.

Licensing your software

About this chapter   This chapter provides information about generating a SySAM license for
                      Sybase IQ 15.0 server.

Contents

Overview

Generate your SySAM license before you install your server or client components. SySAM is a licensing mechanism that provides system administrators with a means to enable and monitor their site's use of Sybase products.

Sybase IQ includes separate SySAM licenses for each product edition and separate licenses for any optional features available for that edition. For example, all Sybase IQ product editions require a CPU license, but an MPXO license is available only for the Enterprise and Single Application editions where multiplex is offered as a separate purchase option.

Sybase Product Download Center

When you purchase a Sybase product you receive an e-mail or web key that tells you how to logon to the Sybase Product Download Center (SPDC). SPDC is an online Web portal where you can download and license your software. You must use SPDC to dynamically generate license keys for Sybase IQ.

The e-mail or web key you get 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 for that product edition.

Enterprise Edition (EE)

Orderable License	Actual License	Required	Optional	Description
CPU	IQ_CORE	•		License restricts the number of cores (not CPUs) which an IQ server can use.
STORAGE	IQ_TBSIZE	•		License restricts the maximum size of the main store (in TB) on which an IQ server can run. The total TB size of the main store size cannot exceed that specified in the license.
LOMO	IQ_LOB		•	License allows you to create and use CLOB or LOB columns. The number of LOB licenses must match number of cores. This license is required for 12.6 or 12.7 customers who already have a LOB license (IQ_LOB).
VLDB MO	IQ_VLDBMGMT		•	Allows you to create multiple table spaces and more than one user dbspace. Number of VLDB licenses must match Storage.
ASO	IQ_SECURITY		•	Allow you to use column encryption, ECC Kerberos and FIPS. The number of ASO licenses must match cores This license is required for 12.6 or 12.7 customers who already have encryption (IQ_ENC).
MPXO	IQ_MPXNODE		•	Allow 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 customers who are already running multiplex. This is a new license requirement.

Small Business Edition (SE)

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

Other limits set by the SE license

- Server will always check out 4 cores (MAXCP)

- Total concurrent IP connections <= 10 (MC)

- Total main store <= 250GB (MS)

Single Application Edition (SA)

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

Other limits set by the SA license

- Total cores per/server <= 32 (MAXCP)

- Total number of user tables <= 25 (MT)

- Total number of columns <= 100 (MV)

- Total number of mpx nodes <= 1 (MSN)

Demo/Evaluation Edition

No license is required for the Evaluation Edition. The Evaluation Edition is an unlicensed server that 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.

These documents are available on the *Getting Started* CD and on the Web at Product Manuals at http://www.sybase.com/support/manuals/

Decide on the license model you want to use

Decide on the license model you want to use before you generate your license file. A license model refers to the way you set up your license files for your licensed program(s). 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.

For information about served and unserved licenses, and which model is best suited for your environment, see Chapter 2, "Choosing a License Model" in 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 at http://www.globes.com/support/fnp_utilities_download.htm#unixdownload

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 at http://www.ece.uci.edu/eceware/flexlm/chap9.htm

---

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 use 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 loop back adapters.

Determine the host name

Although the host name is optional for unserved license models, you must specify the host name of the license server for served licenses. 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, you must specify a port outside of this range.

Generate your SySAM license

The procedure you use to generate your license depends on the license model you plan to use. This section includes separate procedures for "Unserved licenses" on page 29 and "Served licenses" on page 30.

Unserved licenses

Before you generate your license, you need the host ID and host name of the server where you plan to install Sybase IQ. See "Get your host ID" on page 28 and "Determine the host name" on page 28 for more information.

❖ **Generating an unserverd license**

1 Start your browser and log in to the Sybase Product Download Center (SPDC) at https://sybase.subscribenet.com.

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, do the following:

• Choose Unserved license, click Next.

• Specify the number of licenses you want to generate, 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 or SySAM will not recognize the license.

Note If you chose an unserved license model, you can now install Sybase IQ server components. During installation, the installer will prompt you for information about your license.

Served licenses

Before you generate your license, you need the host ID of license server. You may also want to specify the host name to simplify future license administration. See "Get your host ID" on page 28 and "Determine the host name" on page 28 for more information.

❖ **Generating served licenses**

1 Start your browser and log in to the Sybase Product Download Center (SPDC) at https://sybase.subscribenet.com.

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, do the following:

   • Choose Served License, click Next.

   • Specify the number of licenses you want to generate, click Next.

   • Identify the host ID, host name, and port number of the license server. Host name and port number are 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 or SySAM will not recognize the license.

Notes
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. See "Installing a new license server" on page 31 for additional information.

Installing a new license server

If you chose a served license model and do not plan to use an existing license server, you need to install a new one.

The Sybase IQ Server Installation DVD includes SySAM Network License Server utilities you can use to install a new license server. You must run a custom server installation to install a new license server. You do not need to run the license server on the same machine where you install the Sybase IQ server components.

❖ **Installing a new license server**

1   Stop any servers currently running in the install directory before you run the installer.

2   Insert the Sybase IQ Server Product DVD into the drive.

     If the Sybase IQ installer does not start automatically, mount the drive, then use this command to start the appropriate installer:

     ```
     % <DVD_mount_directory>/iq_cd1/setupLinuxAMD64
     ```

     – *Or* –
     ```
     % <DVD_mount_directory>/iq_cd1/setupLinuxAMD32
     ```

3   On the Welcome screen, click Next.

4   Choose an installation directory, click Next.

5   Choose **Evaluation** as the product edition, click Next.

6   On the next screen, choose a location, then read and accept the license agreement. Click Next.

     You must accept an appropriate license agreement to install Sybase IQ. You can review all available license agreements on the Sybase Web site at http://www.sybase.com/softwarelicenses.

7   Choose Custom as the installation type you want to perform, click Next.

8   Deselect all installation features except Sybase Software Asset Management.

9   Select both Sybase Software Asset Management features: Network License Server and Network License Utilities, click Next.

10   Review the installation summary, then click Install.

❖   **Starting your license server**

1   From a command prompt, 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:

Use this command...	To do this...
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   Use the sysam status command to verify that the license server is running. If the server is running, you can expect to see output like this:

```
Starting the SYSAM service
The SYSAM service is starting.
The SYSAM service was started successfully.
lmutil - Copyright (c) 1989-2006 Macrovision Europe
Ltd. and/or Macrovision Corporation. All Rights
Reserved. Flexible License Manager status on Wed
5/24/2006 15:04
License server status: 27000@mysysamserver
License file(s) on keyserver:
/opt/sybase/SYSAM_0/licenses/mysysamserver_60302031
5.lic
```

Notes

• To verify your license, you can run the sp_iqlmconfig stored procedure in Interactive SQL (iqisql) or check the Sybase IQ error log and *<dbname>.iqmsg* file.

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

• 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. You can use the SySAM script to configure the appropriate version of the license server: sysam configure [IPv6|IPv4].

  For example: sysam configure IPv6 will configure use of the IPv4/IPv6 dual stack binaries. For more information, see *SySAM Users Guide*.

• If you have problems activating your license, see "Troubleshooting SySAM" on page 123.

Testing installed products

About this chapter

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

Contents

Topic	Page
Demo database	36
Running Sybase IQ Server	37

Note Instructions in this chapter assume that you installed Sybase IQ server and sourced all environmental variables.

Demo database

Sybase IQ 15.0 now includes a *Do It Yourself (DIY)* demo database. In the *$IQDIR15/demo* directory, you will see the script *mkiqdemo.sh*. This script creates the demo database by modifying, then running the script *mkiqdemo.sql*, which will load in data from the sub-directory */adata*.

The sample data supports an imaginary sporting goods company. The sample database has been updated for the 15.0 release. If you are upgrading from a previous edition, see "Table names" on page 36 for demo database table names. The table includes the former names and new names of sample tables so that you can update scripts or applications.

❖ **Creating the sample database**

The file *$IQDIR15/demo/mkiqdemo.sql* contains SQL statements that create the database and *$IQDIR15/demo/adata* contains its data.

• In the directory *$IQDIR15/demo*, run mkiqdemo.sh.

Note For customers who prefer the older asiqdemo database, you can still create that database by passing in the flag -old to the *mkiqdemo.sh* script.

Table names
If you are upgrading from a previous edition, see the following table for a list of current table names in the sample database. Use this table to update scripts or applications.

Table 4-1: Sample database table names

12.7 name	15.0 name	15.0 owner
contact	Contacts	GROUPO
customer	Customers	GROUPO
department	Departments	GROUPO
emp1	emp1	DBA
employee	Employees	GROUPO
fin_code	FinancialCodes	GROUPO
fin_data	FinancialData	GROUPO
iq_dummy	iq_dummy	DBA
product	Products	GROUPO
sale	sale	DBA
sales_order_items	SalesOrderItems	GROUPO
sales_order	SalesOrders	GROUPO
SYSOPTIONDEFAULTS		DBA

> **Note** Megaphone Telco is not included with Sybase IQ 15.0.

Running Sybase IQ Server

This section tells how to start and stop Sybase IQ Server with the sample database.

Starting the sample database

To start Sybase IQ Server, change to a directory where you have write privileges and run the start_iq utility, using the following command format:

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

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

> **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. Make sure to use a new port number for each server. Set each new port number in the *$IQDIR15/scripts/default.cfg* file. Update each IQ database configuration file (for example, *$IQDIR15/demo/iqdemo.cfg*) by changing the port number in the following line:
>
> ```
> -x tcpip{port=2638}
> ```
>
> To an unused number, for example, 4444:
>
> ```
> -x tcpip{port=4444}
> ```

You can use a configuration file to specify options that you want to set whenever you start your server. A configuration file for the sample database is installed in the *$IQDIR15/demo* directory as an example. For details about configuration files, see "Using configuration files" on page 98.

> **Note** The directory where the server is started becomes the default directory for all server files created by Sybase IQ.

For example, to start the sample database installed with the product, you could use these commands:

```
% cd $IQDIR15/demo
```

```
% start_iq @iqdemo.cfg iqdemo.db
```

Startup parameters
Startup information is saved in the *stderr* log. Output from start_iq ends with this line:

```
Server started successfully
```

For more information, see the *Sybase IQ System Administration Guide*.

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

Using defaults in *default.cfg*, the start_iq utility sets any required environment variables that have not been set and sets parameters that govern Sybase IQ to the recommended defaults. (You may use a configuration file to override these defaults.)

Table 4-2: 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 time–out set to 72 hours. Prevents users with long queries from being logged off over a long weekend.
-iqmc	32	Sets the IQ main cache size
-iqtc	24	Sets the IQ temp cache size

Note On the start_iq command line, the last option specified takes precedence, so if you want 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.

For a complete list and description of startup parameters, see "The database server" in Chapter 2 of *Sybase IQ Reference Manual*.

Note Do not discard this document after installing Sybase IQ. You may need these required parameters later.

If you have Sybase Central, you may use the Start Database Server wizard, as documented in *Introduction to Sybase IQ*, instead of start_iq.

Note On UNIX systems, always run Sybase Central Java and dbisql using the default colors of the Common Desktop Environment. Running these products under Open Windows or changing the default colors may cause display problems.

The server process runs in the background. It sends output to a server log file, *$IQDIR15/logfiles/servername.NNNN.srvlog* where *NNNN* is the number of times the server has been started. For example, *$IQDIR15/logfiles/fiona.0004.srvlog*.

Creating databases

To create your own databases, start and connect to the utility database, as described in Chapter 3. See *Sybase IQ System Administration Guide* for complete instructions.

Running queries

Follow the "Quick start" directions in the *readme.txt* file to run Interactive SQL queries in the sample database.

Stopping the server

To stop a server, run the stop_iq command, for example:

```
                    stop_iq

Checking system...
The following 2 server(s) are owned by 'admin'

##      Owner    PID    Started   CPU_Time
-- --------- ----- -------- --------
1:     admin 4378  10:34:42      0:04
start_iq -gn 25 @iqdemo.cfg iqdemo.db -o /c
--
2:      admin 4726  10:41:09      0:04
start_iq -gn 25 @jd_banking.cfg jd_banking.db -o /c
--
Please note that 'stop_iq' will shutdown a server completely
without regard for users, connections or load process status.
For a finer level of detail the utility 'dbstop' has the options
to control whether a server is stopped based on active
connections.
Enter the server to shutdown ('1'...'2') or 'Q' to Quit:

                    2
Shutting down server ...
Checkpointing server ...............
Server shutdown.
```

Managing processes

The stop_iq –agent command lets you stop the IQ Agent on your UNIX or Linux system. This is compatible with Sybase IQ 12.5 if you have ESD8 or above installed.

Stopping servers in cron or at jobs

To use stop_iq in a cron or at job, specify the utility with 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 You must specify the full path name to the stop_iq executable in the cron statement.

For complete stop_iq syntax, see the *Sybase IQ Utility Guide*.

Other ways to stop servers

There are other ways to stop an IQ database server:

- Run the Interactive SQL (dbisql) STOP ENGINE command

- Select the server name and choose Stop from the dropdown in Sybase Central. For details, see the *Introduction to Sybase IQ*.

- Run the Stop utility, dbstop, documented in the *Sybase IQ Utility Guide*.

CHAPTER 5

Installing Sybase IQ Client Components

About this chapter

This chapter tells you how to install Sybase IQ client components. These components are required to connect to a network server. For information about installing Sybase IQ Server Components, see Chapter 2, "Installing Sybase IQ Server".

Contents

Topic	Page

Overview

The Sybase IQ Client Tools Installation CD includes all of the client tools you need to connect a client machine to a networked IQ Server.

Before you install Sybase IQ client tools

- Read the Sybase IQ Release Bulletin for the latest information. See the special installation instructions section for any installation instructions not included in this document.

- Complete all pre-installation tasks. Apply any required operating system patches, check to see that there is sufficient disk space and memory, and make any necessary configuration changes. See "Pre-installation tasks" on page 8 for specific information.

- Determine the target installation directory. The default installation directory is */opt/sybase* . You can choose an alternate location during installation.

Note Instructions in this chapter are specific to the graphically-based interactive installer, see "Silent Installations" on page 109 for silent installation setup instructions.

Client components

Client components are grouped by feature. Items marked as default are installed in Typical installations.

Feature	Components	Default	Notes
Sybase IQ	Sybase IQ Client	•	Installs client tools, and ODBC drivers.
	Sybase IQ ODBC	•	
Sybase Central	Sybase Central Viewer	•	Viewer and plug–in are required for Sybase Central.
	Sybase Central Plug–in	•	
jConnect		•	Required for JDBC connections.
Sybase Software Asset Management	Network License Server		Network license server utilities, and FLEXnet Licensing utilities.
	License Utilities	•	

What a client installation does

A typical client installation installs the network client components to separate directories on a client machine.

Default installation directory

The default installation directory is */opt/sybase* on the client machine. You can, however, specify another target directory during installation if you use the interactive GUI installer, or by changing the silent installation script if you perform a silent install.

The target installation directory contains subdirectories, scripts, and property files required to support Sybase IQ client. This document uses the variable *$SYBASE* to refer to the parent directory on the client.

Sybase IQ installation directory

The Sybase IQ installation directory is a child of *$SYBASE* directory called *IQ-15_0*.

This directory contains subdirectories, scripts, and property files required to run Sybase IQ client. This document uses the variable *$IQDIR15* to refer to the Sybase IQ installation directory on the client.

Installing Linux client components

The Sybase IQ Client Installation CD includes all IQ client components. The client installer includes a single setup wizard for all product editions. If this is the first Sybase product you plan to install, see "Installing Sybase IQ Server" on page 13 for pre-installation steps.

Note Sybase IQ Network Client does not run on IBM Linux on POWER.

❖ **Installing Sybase IQ Client Tools**

1 Insert the Sybase IQ Client installation CD into the drive.

If the Sybase IQ installer does not start automatically, mount the drive, then use this command to start the appropriate installer:

```
% <DVD_mount_directory>/setup<platform>
```

where *<Platform>* is the name of your operating system. For example, use the command <DVD_mount_directory>/iq_cd1/setupLinuxAMD64 on AMD 64-bit platforms.

2 Click Next on the Welcome screen.

3 Choose the product edition you want to configure, then click Next.

4 Choose a location, then read and accept the license agreement. Click Next.

You must accept an appropriate licence agreement to install Sybase IQ. You can review all available license agreements on the Sybase Web site at http://www.sybase.com/softwarelicenses.

5 Follow the instructions on the screen.

❖ **Setting environment variables**

You must set certain environment variables to run Sybase IQ. Sybase IQ installs environment files that you can run to set variables.

1 bash or Korn (ksh) shell users should type:

```
% . $SYBASE/IQ-15_0/IQ-15_0.sh
```

2 tcsh or C (csh) shell users should type:

```
% source $SYBASE/IQ-15_0/IQ-15_0.csh
```

Installing client components on Windows

The Sybase IQ Windows Client Installation CD includes all of the components you need to install a Sybase IQ client on Windows. The client installer includes a single setup wizard for all product editions.

Use these instructions to install Sybase IQ Client components on Windows. Before you install the client tools, review the "Disk space and memory requirements" on page 46.

Disk space and memory requirements

Use the following table to plan your Windows client installations. Typical installations share some common components and generally require less disk space.

Table 5-1: Sybase IQ 32-bit Client Requirements

Component	Feature	Disk Space	RAM
Client	Sybase IQ	311 MB	-- --
	Sybase Central Java Edition	279 MB	-- --

Component	Feature	Disk Space	RAM
	jConnect	108 MB	-- --
	SySAM Server	150 MB	-- --
	Sybase Software Asset Management Utilities	142 MB	-- --
	Total	990 MB	1 GB

❖ **Installing Sybase IQ Client Tools on Windows**

1   Use an account with Administrator privileges to log in to Windows. Stop and shut down all programs before you run the installer.

2   Insert the Sybase IQ Client installation DVD into the drive.

If the Sybase IQ installer does not start automatically, start Windows Explorer, change to your DVD drive, choose the appropriate setup file for your platform (*setup32.bat* or *setup64.bat*).

3   Click Next on the Welcome screen.

4   Choose the product edition you want to configure, then click Next.

5   Choose a location, then read and accept the license agreement. Click Next.

You must accept an appropriate licence agreement to install Sybase IQ. You can review all available license agreements on the Sybase Web site at http://www.sybase.com/softwarelicenses.

6   Follow the instructions on the screen.

Uninstalling Sybase IQ client

Use these instruction to uninstall Sybase IQ client components.

❖ **Uninstalling Linux or UNIX client components**

•   From a command prompt, change to *$SYBASE/uninstall/IQ*, and type IQuninstall. Follow the instructions on the screen.

❖ **Uninstalling Windows client components**

1   Click Start, point to Settings, choose Control Panel | Add or Remove Programs.

2   Choose Sybase IQ 15.0 from the program list.

3   Click Change/Remove.

4   Follow the instructions on the screen.

CHAPTER 6 **Migrating Data**

About this chapter
This chapter explains how to migrate data from previous Sybase IQ versions to Sybase IQ 15.0.

Contents

Overview

All Sybase IQ databases follow the same basic migration process. Platform migration and multiplex migration includes some extra additional steps. Sybase strongly recommends that you read this entire chapter before performing any migration.

Install Sybase IQ 15.0 before you migrate your 12.x databases. Sybase IQ 15.0 includes tools that you can use to migrate your databases. See "Installing Sybase IQ for the first time" on page 4 for a summary of all installation tasks.

Upgrading from a previous version

- 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 prior to 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 12.7 ESD 5.

- Sybase IQ 12.6 and higher versions do not support existing LONG BINARY columns created using any release prior to Sybase IQ 12.5 ESD8. You must explicitly drop LONG BINARY columns before migrate 15.0, and recreate them after installing. For details, see "Upgrading existing LONG BINARY columns" in *Large Objects Management in Sybase IQ*.

- The SORTKEY function now 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 prior to 15.0 do not contain the same values created using version 15.0 and higher. You should regenerate any sort key values in your database that were generated using a version of Sybase IQ prior to 15.0.

- Sybase IQ 15.0 no longer supports the creation of 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 choose to unload the database and then load the schema and data into a database that you create, then you must use one of the supplied collations.

- Backup 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.

- Review and understand the database migration utilities. Sybase IQ 15.0 includes an iqunload utility that you can use to recreate the schema for your database or simply migrate your database from 12.x to 15.0. An iqlsunload utility is available for 12.7 multiplex users who need to move local stores. Understanding these utilities before you begin can help you plan your migration more effectively.

- Resolve potential migration errors before you migrate. Sybase IQ 15.0 no longer supports some 12.x features and objects. You must 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. See "Migrating simplex databases" on page 72 and "Migrating multiplex databases" on page 76 additional information.

Converting to a new hardware platform

Sybase IQ supports migrating your database from one platform to another, as long as both have the same endian structure. If you are upgrading from a previous version and want to move your Sybase IQ databases to a different supported hardware platform, perform the conversion before you upgrade your data to Sybase IQ 15.0.

Notes

- If you created your Sybase IQ database on a Linux 32–bit version prior to Sybase IQ 12.6 ESD #2, you must 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 higher releases support migration between Windows and Linux.

❖ **Migrating a database from one platform to another (same endian structure)**

1 Back up the database.

2 Shut down the Sybase IQ server.

3 Install the Sybase IQ 12.x 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. If the current version of Sybase IQ is higher than the version on which you were previously running, you need to upgrade databases, and therefore restart the server in a way that restricts user connections. Sybase recommends using two server startup options:

   •   Use -gd DBA so that only users with DBA authority can start and stop databases.

   •   Use -gm 1 to allow a single connection plus one DBA connection above the limit so that a DBA can connect and drop others in an emergency.

   An alternate way to restrict connections is to specify

   ```
   sa_server_option 'disable_connections', 'ON'
   ```

   on the connection where you intend to perform the upgrade and

   ```
   sa_server_option 'disable_connections', 'OFF'
   ```

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

8   Start Interactive SQL and issue the database upgrade statement. For example:

   ALTER DATABASE UPGRADE

   If the database was created with the Java options off, append the keywords JAVA OFF JCONNECT OFF to the preceding command.

Note Now that your database has been converted to the new platform, you can proceed with migrating the 12.x database to Sybase IQ15.0.

Migration utilities

Sybase provides utilities that you can use to migrate your 12.x data to Sybase IQ 15.0.

Sybase IQ 15.0 Unload utility (iqunload)

iqunload is a migration utility included with Sybase IQ 15.0 that performs two main functions:

•   Unloads the schema from a 12.x database.

•   Migrates a 12.x database to Sybase IQ 15.0.

Command line arguments determine the operating mode. You can run iqunload in schema unload mode or database migration mode.

Schema unload mode

Running iqunload in schema unload mode (iqunload -n) lets you connect to an existing IQ 12.x server, and generate a *reload.sql* script that contains the entire schema for your 12.x database. You can run this script against a new Sybase IQ 15.0 database to recreate the schema of the legacy database. Running iqunload in schema unload mode provides two major benefits. You can:

•   Verify the correctness of the schema by applying it to a new IQ 15.0 database.

•   Estimate how long the actual migration process should take.

Database migration mode

Running iqunload in database migration mode (iqunload -au) performs the actual database migration from IQ 12.x to IQ 15.0. This mode does **not** use your existing IQ 12.x servers. Instead, database migration mode uses a special bundled IQ 12.7 ESD 5+ server called iqunlspt.

Running iqunload -au performs the following tasks:

1   Starts your old database with the iqunlspt database engine

2   Performs internal database checks

3   Determines if the database is a MPX write server

4   Generates the schema (similar to schema unload mode)

5   Starts the IQ 15.0 server

6   Creates a new database

7   Applies the generated schema

8   Backs up the operating system files. (This does not use the SQL BACKUP statement.)

"Understanding iqunload" on page 57 for information about prerequisites, syntax, and examples for using the iqunload utility in both modes.

Understanding data storage changes and schema reload

Migration to Sybase IQ 15.0 creates a new Catalog Store. Migration creates and modifies system tables, system procedures, and options in the new Catalog Store to enable 15.0 features and removes preexisting options eliminated for version 15.0. For lists of these options and features, see the *New Features Guide*.

Note Unlike previous versions, Sybase IQ 15.0 preserves previous settings of database options and reapplies them after you create the new Sybase IQ 15.0 database. You no longer need to run SELECT * from SYS.SYSOPTIONS before upgrading.

In Sybase IQ 15.0, 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.

For details on creating new dbspaces after migration and how to move objects from one dbspace to another, see "SQL syntax for dbspace management and partitioning", in Chapter 1, "New Features in Sybase IQ 15.0," in the *New Features Guide*.

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 another filename. 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, then the new store marked as 'MAIN' has DBSpaceName = 'IQ_SYSTEM_MAIN', DBFileName = 'IQ_SYSTEM_MAIN' and path = 'new_main_store.iq.' For MPX migration, the location of the new main store must be visible to all nodes on the multiplex, and the -ms_filename argument is required 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. See "Understanding iqunload" on page 57 for iqunload parameters that determine these specifications and their defaults. For main store size guidelines, see the *New Features Guide*.

Examples

Example 1   Schema unloading allows you to verify schema correctness and approximates how long migration may take.

For example:

*   Make sure that you have copied the required scripts to your IQ 12.x server. See "Correcting invalid SQL syntax" on page 69 for more information.

*   Run iqunload -n to create the empty schema. This creates a SQL file with the entire schema called *reload.sql* in the current directory. The *reload.sql* file also contains a template CREATE DATABASE command that is commented out.

    Note The generated *reload.sql* will not contain any checkpoints. If you have a very large schema, you may wish to edit the reload.sql and add a few checkpoints now and then. If you do not, then IQ will end up with a lot of object metadata objects requiring lots of (-iqmc) main cache memory.

*   Using Interactive SQL or Sybase Central, create a new IQ 15 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.

Example 2   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 specifying a raw device. For a raw device, you must specify an unused raw partition.

For example, 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"
```

Example 3   In CREATE DATABASE syntax, size units are in MB, not GB. The following statement creates a database with 100GB IQ_SYSTEM_MAIN with 100GB reserve:

```
CREATE DATABASE 'test.db'
IQ PATH 'test.iq'
IQ SIZE 102400
IQ RESERVE 102400
TEMPORARY PATH 'test.iqtmp'
TEMPORARY SIZE 5120
```

Files before and after migration

Assume that prior to migration, the directory */iq-i5/unload/127/db/** contains these files:

- *iq127db.db*
- *iq127db.iq*
- *iq127db.iqmsg*
- *iq127db.iqtmp*
- *iq127db.log*

After migration, the directory */iq-15/unload/127/db/** contains the following:

- *iq127db.db* – the new IQ catalog store

- *iq127db.db.before_schema_reload* – the original 12.6 or 12.7 database. If you need to go back to Sybase IQ 12.6 or 12.7, *and you have not started* Sybase IQ 15.0 *in writeable mode*, you may copy this file to the original catalog store name (*iq127db.db* in this example).

- *iq127db.iq* – the original IQ main store. The file and its data are unchanged during migration. Starting Sybase IQ 15.0 in writeable mode flags this file as upgraded, after which it cannot be used in previous versions.

- *iq127db.iqmsg.before_schema_reload* – the original 12.6 or 12.7 message file. This is left intact in its pre-migration state in case you need messages issued during migration while using the 12.7 unload utility embedded in Sybase IQ 15.0.

- *iq127db.iqtmp* – the Sybase IQ 15.0 temporary store.

- *iq127db.lmp* – the Sybase IQ 15.0 license manager property file.

- *iq127db.db.out* – output log file.

- *new_main_store.iq*– the new Sybase IQ 15.0 IQ_SYSTEM_MAIN.

In addition to the files listed, any additional SA catalog stores would also be renamed to have the *.before_schema_reload* extension and new versions created.

Understanding iqunload

iqunload is a command line utility. Refer to this section for information about syntax, parameters, usage, examples, and permissions.

Syntax

iqunload [*options*]

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

Parameters

The iqunload utility takes one or more parameters.

Table 6-1: iqunload parameters

Option	Description
-ap *<size>*	(Optional) Set 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.

Option	Description
-dc	(Optional) Recalculate computed columns in the database.
-ms_filename	Specify filename for new empty IQ_SYSTEM_MAIN store created during database migration. If not specified, the default new main store will be a file system file called *new_main_store.iq*.
	-ms_filename is optional for a simplex migration, but is required for a multiplex migration
-ms_reserve	(Optional) Size of new IQ_SYSTEM_MAIN reserve in MB. If unspecified, defaults to zero.
-ms_size	(Optional) Size of new IQ_SYSTEM_MAIN store in MB, computed by Sybase IQ based on the database size. The minimum, assuming a default page size, is 200MB. If the user specifies –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.
-new_startline	(Optional) Specify startup switches for the new Sybase IQ 15.0 server that is the migration target. For a complete list of server startup switches, see the *Sybase IQ Utility Guide*.
-o *<file>*	(Optional) Log output messages to file.
-q	(Optional) Run quietly. 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.TableName or TableName alone.
-v	(Optional) Return verbose messages.
-y	(Optional) Replace existing reload schema SQL script with new output without confirmation.

Examples

Example 1   Migrate a simplex database to 15.0 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"
```

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"
```

Example 5   Migrate a database and specify startup parameters using a configuration file.

```
iqunload -au -c
"UID=DBA;PWD=SQL;DBF=asiqdemo.db;start=@params.cfg"
```

Note that the user ID and password must be those for the old database. (The default password for 12.6 and 12.7 was upper case "SQL.")

Usage

The specified user ID must have DBA authority.

For schema unloads, the iqunload utility 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.

When migrating a database, specify -au parameter.

Determining switches for migration   The iqunload utility starts and communicates with a special IQ 12.7 server and the IQ 15.0 server. This section documents the default switches sent to these servers to help you understand the cache settings. You can use this information to determine if your existing IQ 12.7 / 12.6 server requires switches to be set during migration.

For most cases, you can skip to "Results of iqunload" on page 61.

The -au parameter causes the utility to start two servers:

- iqunlspt – A special unload engine based on Sybase IQ 12.7 ESD 5.

- iqsrv15 – The new Sybase IQ 15.0 server.

Note   Make sure that your server starts cleanly in IQ 12.6 or 12.7 to minimize startup times.

iqunlspt starts on your Sybase IQ 12.6 or 12.7 database with the following command line arguments:

```
iqunlspt.exe -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. Note that the time required to start your existing Sybase IQ 12.6 or 12.7 database will also be required to start the *iqunlspt.exe* process. This time is included in the startup time for the *iqunload.exe*.

iqsrv15 starts with the following 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 15.0 engine to execute many schema DDL statements.

Both of the server start commands above 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 docs below).

Results of iqunload   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`. If desired, 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. For details about -ch and –cl, see the *Sybase IQ 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. For example, this might occur if tables involved in a view were 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) in order 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, 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 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 cache size.

Logging and correcting errors

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

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*.

For details about the -z diagnostic parameter, see the *Sybase IQ Utility Guide*. For information about request logging (-zr), which has been updated for 15.0, see Request logging at http://dcx.sybase.com/1100en/dbusage_en11/performance-s-5753386.html.

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 to the file *d:\iq127db.out*.

Examine the output file *iq15db.out* or *iq127db.out* for server error messages.

Cannot start database   If the iqunload utility cannot start a database, it returns an error and echoes the command line so that you can correct errors.

Server not found   If you see a message like the following, 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
```

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

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

Obsolete stored procedures logged   Migration replaces 12.7 login procedures to support new 15.0 login management functionality. The procedure sp_login_environment replaces the 12.7 default login procedure DBA.sp_iq_process_login and the post-login procedure dbo.sa_post_login_procedure replaces 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 like the following:

```
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 Managing login policies overview at
http://dcx.sybase.com/index.php#http%3A%2F%2Fdcx.sybase.com%2F1100en
%2Fdbadmin_en11%2Fda-permissi-s-4686947.html. for an overview of login
management functionality.

Working with large schemas

With large schemas, the default cache settings may be too small and the IQ 15.0 server could exhaust dynamic memory. To increase the 15.0 server's cache memory, use the -c server switch. Use the -new_startline to pass this switch to the 15.0 server.

A diagnostic example would include the following switches:

```
-ca 1
-c 1000m
```

```
-o /iq15outputdir/iq15console.out
```

Note The value shown for -c is arbitrary. You should set switches appropriately for your system. The value for -c is in bytes. To specify megabytes, use the m suffix, as shown.

The switches in the example perform as follows:

- ca 1 – Enables dynamic catalog cache sizing. This will also cause memory cache statistics to be logged 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 determine if the settings of 1000m and 2000m are appropriate.

Working with extremely large schemas

Generating a reload schema (iqunload -n) places the entire schema in a single *reload.sql* file. In extreme cases, you may need to break a very large *reload.sql* file into pieces that can be executed sequentially. This will also help 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, please contact Sybase for assistance.

Sybase 12.7 Local Store Unload utility (iqlsunload)

Sybase IQ 15.0 does not support local stores. Multiple nodes can now write to the main store. This new functionality replaces the local stores functionality. iqlsunload is a utility that you can use to unload the contents of an IQ 12.7 local store, and unload schema and data from user tables in an IQ 12.7 local store.

iqlsunload is included with all versions of Sybase IQ starting with 12.7 ESD5. For Sybase IQ 15.0, you can find this utility in the *$IQDIR15/iqlsunload* directory.

Running iqlsunload

iqlsunload is a command line utility. Refer to this section for information about syntax, parameters, usage, examples, and permissions.

Syntax

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

data:[*environment variable* | *file*

Parameters

directory   is a required parameter that contains a path relative to the database on the database server, not relative to the path where iqlsunload is executed. This specification is required to generate *extract_data.sql*. The directory must exist. The unload on the server side will not create it. The iqlsunload utility unloads the actual data files to this target directory when you use Interactive SQL to run *extract_data.sql*. For example, *extract_data.stql* references the directory as follows:

```
CREATE VARIABLE @extract_directory varchar(2048);
SET @extract_directory =
'/testIQ15/database/IQ127_beta2_mpx_test/demo_r1/';
SET TEMPORARY OPTION temp_extract_directory =
'/testIQ15/database/IQ127_beta2_mpx_test/demo_r1/';
```

The iqlsunload application takes one or more optional parameters. Use the appropriate parameters for desired behavior; you cannot perform both functions in the same command.

Table 6-2: iqlsunload command line switches

Parameter	Description
-al	Unload IQ local store schema and data. (Cannot be specified with -t)
-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	Run quietly. 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   The iqlsunload utility requires a minimum server version of Sybase IQ 12.7 ESD #5.

The database must be running before you run iqlsunload.

The specified directory is a path relative to the database on the database server, not relative to the path where iqlsunload is executed.

The iqlsunload tool 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 *Sybase IQ Utility Guide*.

You must run the iqlsunload tool from within the *$IQDIR15/lsunload* directory so that it picks 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 the following 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

Sybase IQ 15.0 no longer allows users with empty user names. Users of this type could formerly be created using statements like:

```
GRANT CONNECT TO "" IDENTIFIED BY 'sql'
```

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 was encountered and the user will not be recreated. The reload operation ignores such users and any associated objects.

When unloading tables, note that:

- Join indexes, which do not exist in a local store, are only unloaded 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.

The iqlsunload utility ignores any system tables or nonexistent tables specified.

Examples

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
```

The preceding command creates three 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 will write to the multiplex. This node could either be 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 Chapter 7, "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 will reload the data extracted from the *extract_data.sql* file.

Permissions

You must have DBA authority to run this utility.

Avoiding potential migration problems

Sybase IQ 15.0 no longer supports some 12.x features and objects. You must correct these items before you migrate your databases. "Addressing unsupported objects" on page 68 tells you how to identify and update unsupported objects and metadata. "Correcting invalid SQL syntax" on page 69 identifies syntax changes you must make before migration.

Addressing unsupported objects

Running iqunload in schema unload mode (iqunload -n) against an existing 12.x server generates a *reload.sql* script that contains the entire schema for a 12.x database. You can compare the contents of this script against the following table to identify unsupported objects and metadata.

Table 6-3: Troubleshooting unsupported metadata

Object	Details	Action
Invalid database, table, or user names	Table names can no longer contain double quote characters. User names and database names can no longer contain double quote characters, single quote, and semicolon characters.User names and database names may no longer start with a space or end with a space.	Change these names
Join indexes	The iqunload utility will 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	The iqunload utility searches BLANK PADDING OFF databases for any indexes that would become invalid after migration. The iqunload utility fails and lists indexes and constraints that must be dropped and in which order.	Drop these indexes and constraints before schema reload and recreate the indexes and constraints after schema reload has been completed.
Unenforced constraints	The iqunload utility tool 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 below may have these indexes. Because these indexes were automatically created by Sybase IQ, you cannot drop and recreate them; you need to rebuild them.	Allow iqunload to check for these and list them. Rebuild these indexes using sp_iqrebuildindex before migration. For details about rebuilding indexes, see "Configuring FP(3) indexes" in the *New Features Guide*. The rebuilt indexes will be upgraded.

Correcting invalid SQL syntax

See the following table for steps to address elements that cause iqunload failures.

Table 6-4: 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), then the reload fails. (For reserved words, see the *Sybase IQ Reference Manual* and *SQL Anywhere Server SQL Reference*.) For example:
 	
```CREATE PROCEDURE p( )```
```BEGIN```
```DECLARE NCHAR INT;```
```SET NCHAR = 1;```
```END```
	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.

Problem	Solution
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: ```\nSET TEMPORARY OPTION\ntsql_outer_joins='on'\n``` and also set this option for your database. You should later 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.
Functions that have OUT or INOUT parameters cannot be reloaded.	OUT and INOUT parameters are no longer supported. The functions must be dropped before doing a reload.

Note If the examples in the above two tables do not apply to you, skip the next section, and proceed directly to either "Migrating simplex databases" on page 72 or "Migrating multiplex databases" on page 76.

❖ **Correcting invalid SQL**

Sybase recommends 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.

As an aid to users, 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 15.0. If you want to create a completely new Sybase IQ database, see Chapter 1, "New Features in Sybase IQ 15.0," in the *New Features Guide* for sizing guidelines.

4   Start the utility database using the Sybase IQ 15.0 Server, and create a new, empty, IQ 15.0 database.

5   Use Interactive SQL to run the *reload.sql* script that contains the unloaded schema against the new IQ 15.0 database. This loads the schema of the old database into an IQ 15.0 database. If there are any SQL syntax errors, you will receive warnings when running the SQL script. Ignore the errors found in order to get a complete list of the problems. Modify the database to be migrated to eliminate the errors found. See "Correcting invalid SQL syntax" on page 69 to determine how to fix the problems.

6   Perform this process iteratively if necessary until you can cleanly reload the schema unloaded from the database to be migrated, against the empty 15.0 database.

Migrating simplex databases

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

Task summary

- Review "Upgrading from a previous version" on page 50 for steps common to all database migrations.

- Run sp_iqcheckdb to check the validity of your current databases. Repair any index and allocation problems. See "Verify your databases" on page 73 for more information.

- Shutdown your 12.x servers. See "Shutdown your 12.x server" on page 73 for more information.

- Run iqunload with the appropriate parameters to migrate your databases. See "Migrate your simplex database" on page 74 for more information.

- Start the migrated database in readonly mode. Update the appropriate configuration settings and database options. See "Start the 15.0 migrated database" on page 75 for more information.

- Use sp_iqcheckdb to verify the migrated database. See "Verify the migrated databases" on page 75 for more information.

- Complete any necessary post-migration tasks. See "Post-migration tasks" on page 93 for more information.

Verify your databases

Run the 12.x version of sp_iqcheckdb to check the validity of your current databases. Make sure that the database is in a valid state and can start without requiring recovery.

The 12.x versions of sp_iqcheckdb use input parameters, rather than database options, to specify the type of database consistency checking.

❖ **Running sp_iqcheckdb**

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 by typing the following:

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

4   Issue a COMMIT statement.

Examine the sp_iqcheckdb report for errors. Repair any index and allocation problems.

Note For information on interpreting the sp_iqcheckdb results and corrective action, refer to Chapter 13, "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.

Shutdown your 12.x server

At the command prompt, issue a stop_asiq command to shutdown your 12.x server. You can also stop a server by using the STOP ENGINE command from Interactive SQL or any front–end client.

Migrate your simplex 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. See "Sybase IQ 15.0 Unload utility (iqunload)" on page 53 for additional options.

This example 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. The -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 could 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 15.0.0.5120
    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".
```

Start the 15.0 migrated database

Use this command to start the migrated database in readonly mode with Sybase
IQ 15.0:

```
start_iq -iqro 1
```

Update the appropriate configuration settings and database options. the
database. See the New Features Guide and Release Bulletin for changes to
legacy features and options.

Verify the migrated databases

Run sp_iqcheckdb to verify the consistency of the databases.

Note In this step, you run the IQ 15.0 version of sp_iqcheckdb, which uses
input parameters, rather than database options, to specify the type of database
consistency checking.

❖ **Running sp_iqcheckdb**

1   Start the database using the read-only switch, -iqro 1.

    When starting the coordinator in 15.0, 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 output to a file by typing the following:

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

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

4   Issue a COMMIT statement.

Note For information on interpreting the sp_iqcheckdb results and corrective action, refer to Chapter 13, "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.

Task summary

- Review "Upgrading from a previous version" on page 50 for steps common to all database migrations.

- Synchronize and shutdown multiplex all query nodes. See "Synchronize and shutdown multiplex query nodes" on page 77 for more information.

- Migrate local stores, if necessary. This step is only required if you created local stores for your 12.x multiplex query nodes. See "Migrate your local stores" on page 79 for more information.

- Run sp_iqcheckdb to check the validity of your 12.x databases. Repair any index and allocation problems. See "Verify the 12.x databases before migration" on page 85 for more information.

- Start the multiplex write server in single node mode. See "Start the multiplex write server in single node mode" on page 86 for more information.

- Shutdown your 12.x server and backup your database. See "Shutdown your 12.x server" on page 86 for more information.

- Run iqunload with the appropriate parameters to migsrate your databases. See "Use iqunload to migrate your database" on page 87 for more information.

- Use the -iqro 1 -iqmpx_sn 1 switches to start the writer in READONLY mode. See "Start the 15.0 migrated database" on page 90.

- Run sp_iqcheckdb to verify your migrated databases, then backup the databases. See "Verify the migrated databases" on page 90 for more information.

- Synchronize the secondary nodes from the coordinator node. See "Manually synchronize the secondary nodes" on page 91 for more information.

- Start the secondary nodes, then import the schema and data for the local stores into the coordinator. See "Start the secondary nodes" on page 93 for more information.

- Set the failover mode for the failover node. See "Set the failover mode" on page 93 for more information.

- Complete any necessary post-migration tasks. See "Post-migration tasks" on page 93 for more information.

Synchronize and shutdown multiplex query nodes

When you migrate a multiplex, Sybase recommends that you check the SQL Remote and multiplex server log files for synchronization problems.

❖ **Correcting synchronization problems**

1   Start the multiplex server.

2   Start SQL Remote on all nodes in the multiplex, if not already started. 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. Note:
    the server console log file, by default, is created in *$ASDIR/logfiles*.

For example:

```
Now accepting requests

OS 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 the sleeping to stop to give SQL Remote a chance to process
any messages. SQL Remote will go back to sleep once it finishes
processing. 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   Please 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. For details, see "Verify the 12.x databases before migration" on page 85.

Migrate your local stores

If you created local stores for your 12.x multiplex query nodes, you need to move these stores before migrating data to Sybase IQ 15.0. The iqlsunload tool connects to a Sybase IQ 12.7 database with an IQ Local Store and creates scripts that unload its schema and data.

Local Store migration means consolidating node-specific information into either the existing 12.7 Writer or the new IQ 15.0 Main Store. You can customize the steps in the process to meet your data requirements.

If existing query nodes have node or department-specific information, you can use tablespaces and partitioning to achieve the same results. For more information, see Data storage in Chapter 3, "Multplex Overview," in the *New Features Guide*.

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

Query server data partitioning

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

❖   **Partitioning query server data**

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

1   Unload the schema and data from the query nodes.

The *reload_schema.sql* script produced for each query node will contain 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 IQ 15.0 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 of the query node data sets.

Query server data overlap

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

❖ **Addressing overlapping query server data**

1 Run iqlsunload against all your query servers with local stores that have the data you wish to consolidate.

2 Modify the *reload_schema.sql* and *reload_data.sql* files to have the new table names. Leave the *extract_data.sql* file alone as if references the table found in the query server's local store.

3 Run the *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 the following:

```
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"),
```

Edit the *reload_schema.sql* and change it as follows:

```
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"),
```

The *extract_data.sql* file contains the following:

```
--
-- 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 = '';
SET TEMPORARY
```

Leave the previous *extract_data.sql* code unchanged in order to extract the sales_order table from the query server.

Suppose that *reload_data.sql* contains the following:

```
-- 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 the following:

```
-- 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;
```

The preceding 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**

Follow these steps to unload the schema and data for local objects in the 12.7 IQ Local Store and move them to another node for migration.

1   If using an older version, you *must* upgrade to the minimum version Sybase IQ 12.7 ESD #5 before local store migration.

2   The iqlsunload utility is installed with Sybase IQ 15.0 in the directory *$IQDIR15/lsunload*. You must source the *ASIQ-12_7.sh* or *.csh* file before running the iqlsunload utility.

3   Run the 12.7 iqlsunload utility against each query server with a local store. For syntax and examples, see "Sybase 12.7 Local Store Unload utility (iqlsunload)" on page 64.

4   Edit the *reload_schema.sql* script as follows:

   •   Delete unwanted objects.

   •   Uncomment any objects that you want to reload that are commented out in the *reload_schema.sql* script.

   •   Add commands to define any objects that you defined in sp_mpxcfg_<*servername*> procedures.

5   Edit the *extract_data.sql* script to remove objects that you do not wish to migrate. These objects are generally the same ones that you removed from *reload_schema.sql*.

6   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.

7   Run the *reload_schema.sql* and *load_data.sql* scripts against the 12.7 write server.

   Note If you prefer, you may wait until the write server has been migrated to 15.0 and run the *reload_schema.sql* and *load_data.sql* scripts against the new 15.0 coordinator.

Verify the 12.x databases before migration

Sybase recommends that you run sp_iqcheckdb to verify the consistency of your 12.x databases before you migrate.

Note In this step, run the IQ 12.6 or 12.7 version of sp_iqcheckdb, depending on the version from which you are migrating. These versions of sp_iqcheckdb use input parameters, rather than database options, to specify the type of database consistency checking.

❖ **Running sp_iqcheckdb**

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 by typing the following:

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

4 Issue a COMMIT statement.

Note For information on interpreting the sp_iqcheckdb results and corrective action, refer to Chapter 13, "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.

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, then shutdown, start up 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 the following 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* that you will use to put the server into a good state 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.

Shutdown your 12.x server

At the command prompt, issue a stop_asiq command to shutdown your 12.x server. You can also stop a server by using the STOP ENGINE command from Interactive SQL or any front–end client.

Use iqunload to migrate your database

Run iqunload with the appropriate parameters to migrate your databases. Multiplex migration differs from Simplex migration and requires additional parameters.

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 complete syntax, see "Sybase IQ 15.0 Unload utility (iqunload)" on page 53.

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

- You need to 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 that the name of the server match the naming conventions for the multiplex nodes. Once the 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 needs to be visible and accessible by all nodes of the multiplex. This is important because main store filename defaults to *new_system_main.iq*. This path is relative to the catalog database file (.db). Later, when the 15.0 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 are unable to 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, then the server name is enforced. If you are unsure of the servername, check the administrative script *start_server* in the database directory.

- The DBF= argument must specify the actual path that you 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 15.0.0.5120
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_sche
ma_reload".
Successfully backed up file
"/sunx5prod/users/marshall/mpx127/main.db" by renaming
it to
"/sunx5prod/users/marshall/mpx127/main.db.before_schem
a_reload".
Successfully backed up file
/sunx5prod/users/marshall/mpx127/main.iqmsg" by
renaming it to
"/sunx5prod/users/marshall/mpx127/main.iqmsg.before_sc
hema_reload".
```

```
Successfully reloaded schema of database
"/sunx5prod/users/marshall/mpx127/main.db".
```

Start the 15.0 migrated database

Use this command to start the migrated database in readonly mode with Sybase IQ 15.0:

```
start_iq -iqro 1 -iqmpx_sn 1
```

Update the appropriate configuration settings and database options. See the New Features Guide and Release Bulletin for changes to legacy features and options.

Verify the migrated databases

Run sp_iqcheckdb to verify the consistency of the migrated databases.

Note In this step, you run the IQ 15.0 version of sp_iqcheckdb, which uses input parameters, rather than database options, to specify the type of database consistency checking.

❖ **Running sp_iqcheckdb**

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 15.0, 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 output to a file by typing the following:

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

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

4   Issue a COMMIT statement.

Because we started the server in readonly mode, the server cannot complete some post migration bookkeeping tasks, and the verification will report 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. For information on interpreting the sp_iqcheckdb results and corrective action, refer to Chapter 2, "System Recovery and Database Repair," in *Sybase IQ Troubleshooting and Recovery Guide*. 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 readonly (-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 then restart it normally without any special switches.

Manually synchronize the secondary nodes

To start the secondary nodes, you first need to install Sybase IQ 15.0 and then synchronize from the coordinator node. When you migrate a query node, it will become a reader node.

❖ **Synchronizing secondary nodes manually**

1   Back up the query node files. It is recommended 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
```

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 11.0.0.5099
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/stop administration scripts on the secondary nodes to use the new name.

To retain the same catalog database file names:

a   Rename the synchronized coordinator catalog database file name. Assuming the coordinator file was called *main.db* and the secondary server was called *q1.db*

For example:

```
mv main.db q1.db
```

b   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:

(Your shell may differ.)

```
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 the preceding steps on the remaining secondary nodes that you want to migrate.

Start the secondary nodes

You must complete the steps in "Manually synchronize the secondary nodes" on page 91 and all nodes of the multiplex must be up and running in order to execute the following steps.

After you synchronize the secondary nodes, you can start them. For details about how the administration scripts have changed, see *New Features Guide*.

In general, secondary servers can be started with the command line startup utility. For example:

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

In this syntax, *<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 mode

For MPX migration the failover node, you need to set the failover mode. This is done while connected to the coordinator and *server name* is one of the reader nodes. Syntax for this is:

```
ALTER MULTIPLEX SERVER servername ASSIGN AS FAILOVER
SERVER
```

Post-migration tasks

After you migrate you databases, you may want to perform these post-migration tasks.

Recreating indexes for EUC_TAIWAN data

In Sybase IQ 15.0, the character encoding specification for the EUC–TAIWAN collation has been corrected and now uses the EUC_TW character set. Due to the correction, you must recreate indexes on data in version 12.7 or earlier databases that use the EUC_TAIWAN collation to make them work with Sybase IQ 15.0.

Updating configuration files

Be sure to 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. Be sure that you 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 above.

Preserving database options

Sybase IQ preserves the settings of all 12.7 database options that are still valid in Sybase IQ 15.0 in migrated databases. For more information, see Deprecated features in Chapter 2, "Behavior Changes in Sybase IQ 15.0,"and Database option changes in Chapter 2, "Behavior Changes in Sybase IQ 15.0," in the *New Features Guide*.

Backing up your databases after migration

Sybase recommends that you back up your databases again with the BACKUP statement. For complete syntax, see the *Sybase IQ System Administration Guide*.

This backup after the migration is recommended but not required. If you use the IQ BACKUP statement instead of a system–level backup, you can run backups and queries concurrently.

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

If you do not use multiplex capability, your migration is complete. The following sections apply only to multiplex migration.

CHAPTER 7 **Configuring Sybase IQ**

About this chapter   This chapter tells how to configure Sybase IQ.

Contents

Topic	Page
Using configuration files	98
Configuring client connectivity	99

Using configuration files

If you use an extensive set of options to start your server, you may find it useful to store them in a configuration file.

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, then the value in command line overrides the one 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 *Sybase IQ Utility Guide* for a complete list of switches.

Default configuration file

$IQDIR15/scripts directory includes a default configuration file, *default.cfg*. This file is also the source of parameters for the Service Manager, the *params.cfg* file used by Sybase Central and in multiplex configurations. You can maintain consistency by editing parameters in *default.cfg*.

If you need to increase the amount of virtual memory, for example, you can 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. For details see the *SQL Anywhere Server Database Administration Guide*. If you specify log file parameters (-o *logfile*) in encrypted files, the log is not available to the IQ Agent or start_iq, which is unable to 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
```

Configuring client connectivity

Sybase IQ supports ODBC and JDBC connectivity. Sybase IQ 15.0 uses SQL Anywhere to store catalog information.

Sybase IQ versions prior to 12.0 used Adaptive Server Enterprise catalog server. ODBC applications used as client front–end tools with older versions of Sybase IQ will continue to run in IQ version 12.x, but third–party and customer–written Open Client™ DB–Library and Client–Library applications are unlikely to perform as expected.

When developing Open Client applications to run with Sybase IQ 12.x, avoid using catalog tables or system stored procedures supported by Adaptive Server Enterprise but not Sybase IQ. See Appendix A, "Compatibility with Other Sybase Databases," in the *Sybase IQ Reference Manual*.

Sybase IQ is compatible with clients (like Open Client–Library and DB–Library) that use TDS. Open Client version 11.1.1 supports TDS 5.x and applications would therefore work with Sybase IQ, but only if these applications use the system tables, views and procedures that are found in Sybase IQ. System procedures, catalog tables, and views available in Sybase IQ are listed in the *Sybase IQ Reference Manual*. There are no restrictions accessing data in the IQ Store through any supported interface.

Connecting using JDBC

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

Sybase Central and Interactive SQL (iqisql) can use either JDBC or ODBC connectivity. For an overview of using JDBC in Sybase IQ, see "Data Access Using JDBC" in the *Sybase IQ System Administration Guide*.

Notes

- The SQL Anywhere JDBC driver is the default driver for Sybase Central and dbisql.

- IQ Agent can only use jConnect with JDBC to connect to a multiplex server. The jConnect JDBC drivers have been certified with multiplex and non–multiplex servers.

Connecting using ODBC

Open Database Connectivity (ODBC) is a standard application programming interface (API) developed by Microsoft. It allows a single application to access a variety of data sources for which ODBC–compliant drivers exist. The application uses SQL as the standard data access language.

ODBC conformance

ODBC drivers manufactured by different vendors may vary widely in the functions they provide. Sybase IQ supports ODBC 3.5.2. ODBC 3.x, drivers are classified based on the features they provide:

Conformance Level	Sybase IQ Support
Core conformance	All Core level 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.
Level 2 conformance	All Level 2 features, except for the following:
	• 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 request and SQL queries.

Notes

- 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 at http://www.microsoft.com.

Installing ODBC drivers

When you install Sybase IQ on your UNIX or Linux server, the installation procedure also installs the ODBC driver, which can be directly accessed by applications. If you are using an ODBC application that uses *libodbc.so* (*libodbc.so.1*) or *libodbcinst.so* (*libodbcinst.so.1*), simply create symbolic links to that point to *$SYBASE/IQ-15_0/lib/libdbodbc11.so.1* for single threaded or *$SYBASE/IQ-15_0/lib/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.

Note The filenames cited above have platform–specific suffixes. The *so* suffix shown is specific to the Sun Solaris system. See "Linking ODBC applications on UNIX" in Chapter 7 of the *SQL Anywhere Server Programming Guide* for a list of files included in the driver managers for supported UNIX platforms.

The ODBC Driver shipped with Sybase IQ connects clients on a Windows platform or the platform of the IQ server to the IQ server. To connect clients on other UNIX or Linux platforms to your server, download and install a platform–specific Sybase IQ ODBC Driver. Check the EBF/Update information on the Web for the appropriate driver.

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, you need to download and install the 32-bit ODBC driver kit. Check the EBF/Update information on the Web for the appropriate driver.

Using UNIX or Linux–based query tools through ODBC

UNIX or Linux platforms do not provide a standard ODBC driver manager, but you can use the driver manager emulation capabilities provided by the Sybase IQ ODBC driver to connect to the driver.

Tools (like Brio) that do not require extensive driver manager services can use the symbolic links provided with Sybase IQ to connect directly to the driver. Tools (like Whitelight) that require the presence of a driver manager may ship with a driver manager.

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

Third–party ODBC applications

For information on third–party vendor applications that are certified with Sybase IQ, refer to the section "Sybase certifications on the Web" in the *Sybase IQ Release Bulletin*.

Creating ODBC data sources

You need an ODBC data source on the client computer for each database you want to access using ODBC. A data source describes how to connect to data on a network. For example, a data source may include the name of a database, the server where it resides, and the network used to access the server.

On UNIX or Linux, ODBC data sources are held in a file named *.odbc.ini*. You can edit this file with any text editor to specify data sources. For details, see "Using ODBC data sources on UNIX," in the *Sybase IQ System Administration Guide*.

You can also use the cross–platform iqdsn utility to create data sources. See "The Data Source utility" in the *Sybase IQ Utility Guide*.

Note You cannot connect to a Sybase IQ 12.6 or 12.7 server using an ODBC Data Source Name created for a 12.5 server, even if you specify the same server name, port number, and database name. Use the ODBC Data Source Administrator to remove DSNs created in 12.5 and add new DSNs.

You may use files as data sources instead of databases. File data sources are stored as files with the extension *.dsn*. For information about creating a file data source, see *Sybase IQ System Administration Guide*.

❖ **Testing an ODBC data source**

1   Start the database.

2   In the ODBC Data Source Administrator, select your new data source from the list of User Data Sources.

3   Click Configure.

4   On the ODBC Configuration dialog box, click Test Connection.

   If you cannot access the data source, check that you have filled out the various tabs with correct file and path names.

Storing connection information

If you are running a UNIX client, an *.odbc.ini* file on your system stores the information used to access each database. (Note that the filename begins with a period.)

To connect with ODBC data sources, the location of your *.odbc.ini* file must be referenced by one of the following variables. Sybase IQ searches the directories specified by the variables below in the following order:

• $ODBCINI – must contain the exact full path name of the *.odbc.ini* file.

• $HOME

•   current directory

• $PATH

Sybase IQ clients ignore the following environment variables when searching for *.odbc.ini*:

• $ODBC_HOME

• $ODBC_INI

Once you have created a data source entry, you can connect to your database, by entering the dbisql command at the command prompt and specifying the data source entry name in a connection string. Sybase IQ finds the rest of the connection information in the *.odbc.ini* file. For example:

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

For more information about dbisql and its options, see *Sybase IQ Utility Guide*.

Note For Sybase IQ Version 12.5 and higher, by default any database that is started from a connection string is *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. (This does not apply in the case of multiplex IQ databases, which are started with Sybase Central.)

To have the database continue running after connections disconnect, as in Sybase IQ releases prior to 12.5, you must specify the connection parameter AutoStop=No in your connection string or data source.

For example, the following data source fragment instructs the client library to keep the database loaded after the connection is dropped:

```
[dbcli7 Connection Parameters]
ServerName=testsrv
Autostop=No
UserID=DBA
Password=sql
```

If you want to connect without using *.odbc.ini*, you can enter an Interactive SQL command that specifies the entire entry, like the following. While it is shown here on multiple lines, you must enter the entire command at the command prompt on one line.

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

Connecting using OLE DB

OLE DB is a data access model from Microsoft. It uses the Component Object Model (COM) interfaces. Unlike ODBC, OLE DB does not assume that the data source uses a SQL query processor.

This release of Sybase IQ includes an OLE DB provider. If you use the Sybase IQ OLE DB provider, ODBC is not required in your deployment.

OLE DB requires a Windows client. However, you can access both Windows and UNIX servers using OLE DB.

For more information, see *SQL Anywhere Server Programming Guide*. For information on connecting to a database using OLE DB, see "Connecting to a database using OLE DB" in Chapter 3, "Sybase IQ Connections," in the *Sybase IQ System Administration Guide*.

Note Sybase IQ support for certain features used with OLE DB differs from SQL Anywhere support. Be aware of these differences when using the SQL Anywhere documentation:

- 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

If you need to insert from an Adaptive Server Enterprise database to a Sybase IQ database, or you want a simple connection using iqisql, each server must have an entry in the interfaces file on the client computer. Interfaces file entries, also called **server objects**, also simplify database startup.

Use iqdsedit (IQ Directory Services Editor) to create entries in the interfaces file. You must be the owner of the Sybase home directory (*$SYBASE*) in order to run iqdsedit. For more about iqdsedit, see "Configuring IQ servers with DSEDIT" in Chapter 15, "Sybase IQ as a Data Server," in the *Sybase IQ System Administration Guide*.

Note Sybase IQ 15.0 includes a simplified version of the TDS tools **iqdsedit** and **iqisql.** To use advanced features, like encrypted connections, you will need the full version included with ASE, SDK, or Replication Server.

❖ **Adding a server object**

The iqdsedit utility lets you view and edit server entries in the interfaces file using a GUI based on X11/Motif in UNIX platforms. Be sure to make a copy of the interfaces file before you make any changes.

1   Start the Open Client Directory Service Editor.

```
% $IQDIR15/bin64/iqdsedit
```

2   The default interfaces file and configuration file display.

Click OK to continue or Exit to quit.

3   Select Add new server entry.

4   Type the server name and click Add new network transport. *The server name in your* iqdsedit *entry must be the same as the database name*.

5   Select the TCP transport and enter the host name and port number. (The defaults are usually sufficient.) Click OK.

6   Click OK.

7   The server is now listed under Available servers.

8   Click Close session to make new server entries usable.

9   Click Exit to close iqdsedit.

10 Click Yes to "Are you sure you want to exit iqdsedit?"

Running client and server on the same system

Shared memory is the default communications mechanism when the client and server are on the same system. It is configured automatically, and starts up automatically, on both UNIX and Windows platforms. Sybase IQ uses a shared memory segment and several semaphores for communication between the client and server on the same machine.

See the "Connecting using command line utilities" in Chapter 3, "Sybase IQ Connections," in the *Sybase IQ System Administration Guide* for more 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.

In order 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 the following TCP/IP implementations:

- **NetWare**   TCP/IP For NetWare.

- **Windows**   Microsoft Winsock version 2.0.

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.

Although the default packet size for TCP/IP is 1460 bytes, a larger packet size may improve query response time, especially for queries transferring 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.

For more information about server startup switches, see the *Sybase IQ Utility Guide*. For more information about connection parameters, see the *Sybase IQ System Administration Guide*.

Connecting 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.

When connecting across a firewall, you must use a set of communication parameters in the CommLinks connection parameter of your application's connection string.

- 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.

For more information on these parameters see the section "Network communications parameters" in Chapter 4, "Connection and Communication Parameters," in the *Sybase IQ System Administration Guide*.

Example

In the following 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)
```

APPENDIX A   **Silent Installations**

About this chapter

This chapter tells you how to install Sybase IQ without having to interact with the standard installation wizard.

Contents

Topic	Page
Overview	110
Installing IQ silently	110
Uninstalling IQ silently	114
Sample configuration files	117

Overview

Your installation media includes scripts that perform standard IQ installs and uninstalls without the need to interact with the product. These scripts are configured to perform typical installations by default; custom installation tasks require additional configuration.

The installation media includes installation scripts, an appropriate JRE, CLASSPATH settings, java command string, and pointers to the text file that contains the feature selections and configuration options.

Note Client and Server silent installation scripts share the same filenames. Be sure that you are running the correct script when you are installing or uninstalling IQ components.

Installing IQ silently

IQ silent installation scripts support typical and custom installations. See "Editing the installation parameters" on page 111 for information about changing the installation defaults; see "Running a silent install" on page 112 for information about running a silent install.

Task summary

The following list summarizes the tasks required to run a silent install.

- Mount the DVD and set up the sybase account. If your DVD drive is set up for automount, skip this step. If you need to mount the DVD drive, see "Mounting the installation media" on page 18.

- Create a sybase user account to provide consistent ownership and privileges for all installation, configuration, and upgrade tasks.

  If a sybase account already exists, skip this step, but verify that you can use this account to log in the system. For more information, see "Setting up the sybase account" on page 19.

- Set the *$SYBASE* environmental variable to directory where you plan to install Sybase IQ. See "Setting the $SYBASE environmental variable" on page 19 for more information.

- Update the installation parameters, if necessary. Installation parameters are stored in *SilentInstall.txt* and match the defaults in a typical installation. If you want to change any installation options, you need to edit this file and save your changes on the file system. See "Editing the installation parameters" on page 111 for more information.

- Run the silent install scripts. Be sure to complete all preinstallation tasks before you run the installer. See "Running a silent install" on page 112 and "Completing the installation" on page 113 for more information.

- Set the environment variables. Installing IQ also installs some environment files that set variables that are required to run Sybase IQ Server. See "Setting environment variables" on page 114 for more information.

Editing the installation parameters

Installation parameters are stored in *SilentInstall.txt*. *SilentInstall.txt* is configured to perform typical installations by default; custom installs require additional configuration.

SilentInstall.txt contains the same default selection and configuration options as the interactive installer. If you change the target installation directory, for example, use an editor to change the default parameters. See "SilentInstall.txt" on page 117 to review the parameters in this file.

- Comments in the file are prefixed with a # sign

- Modifiable parameters appear between *"quotes"*

- Feature selections start with a *–P*

- Variable options are prefixed with a *–V*

If you modify the file, read the comments before the section you want to change, update the parameters, then save *SilentInstall.txt* to the local file system.

Running a silent install

Launch *SilentInstall.sh* from a console to start the install. You must include a license acceptance switch as an argument when you run this script. Other arguments depend on whether you plan to run a "Typical install" on page 112 or "Custom install" on page 112.

Command line arguments

You can display options for this script by running the *SilentInstall.sh* script with the *–h* or *–?* option in a console.

- Display help and exit *–h* or *–?*

- Include user specified JVM_ARGS *–jvm "JVM_ARGS"*

- Specify the location of the java folder *–java <path>*

- Specify the location of Silent Install text file*–txt <path>*

- Specify the bit size of the installed files *–bs <32|64>*

- Specify the location of the silent install console log file *–cl <path>*

- Specify the path to root directory of installer files *–src <path>*

- Agree to the Sybase IQ License *–I_accept_sybase_license*

- Specify the desired install location for Sybase IQ *–dst <path>*. This parameter is required if the *$SYBASE* variable is not set.

Typical install

You can run a typical silent install directly from the installation CD, and must include the *license acceptance* switch:

```
% ./<DVD_mount_directory>/ SilentInstall.sh -
I_accept_sybase_license
```

If you do not have the *$SYBASE* environment variable set, you must also specify the *-dst* parameter.

Custom install

If you want to perform a custom install, the command you use to run *SilentInstall.sh* must include the *–txt* switch that points to the location of the modified *SilentInstall.txt*. The command must also include the *license acceptance* switch:

```
% ./SilentInstall.sh -txt <directory
name/SilentInstall.txt -I_accept_sybase_license
```

If you do not have the *$SYBASE* environment variable set, you must also specify the *-dst* parameter.

Note

•   If you change the license acceptance and destination parameters in the
    SilentInstall.txt file, you do not need to specify those options on the
    command line.

•   You can monitor silent installs by running a tail command on the logfile.
    The default name and location of this file is */tmp/IQ15ConsoleNNNN.log*
    where *NNNN* is the PID. This location and logfile name can be changed by
    adding *–cl* and the log location as a command line parameter when
    launching the silent install.

Completing the installation

•   Log on as the root user and unmount the DVD (if you had to mount to
    begin the installation):

```
% cd/
umount <DVD_mount_directory>
```

•   Remove the DVD from the drive.

•   If you installed Sybase Central, see the *Sybase IQ System Administration
    Guide* for instructions on configuring and running the IQ Agent. You must
    configure and run the IQ Agent in order to manage multiplexes with
    Sybase Central.

•   Set the environment variables as described in "Setting environment
    variables" on page 114, then log out.

Note After you install the software, Sybase strongly recommends that you
check the online support Web site for software updates. If a software update
(EBF) has been released, it contains bug fixes made after this product shipped.

Completing the installation

After you run the install, remove the DVD from the drive.

If you installed Sybase Central, see the *Sybase IQ System Administration Guide* for instructions on configuring and running the IQ Agent. You must configure and run the IQ Agent in order to manage multiplexes with Sybase Central.

Note After you install the software, Sybase strongly recommends that you check the online support Web site for software updates. If a software update (EBF) has been released, it contains bug fixes made after this product shipped.

Setting environment variables

You must set the following environmental variables to run Sybase IQ Server.

❖ **Setting the environmental variables**

• The environment files that set Sybase IQ variables also set the variables needed by Open Client.

bash or Korn (ksh) shell users should enter:

```
% . $IQDIR15/IQ-15_0.sh
```

tcsh or C (csh) shell users should enter this command:

```
% source $IQ1DIR15/IQ-15_0.csh
```

Note These procedures set the *$IQDIR15* environment variable used to start Sybase IQ.

Uninstalling IQ silently

IQ silent uninstallation scripts support typical and custom uninstallations. See "Editing the uninstall parameters" on page 115 for information about changing the uninstall parameters; see "Running the uninstall" on page 115 for information about running the uninstaller.

Note You must run a silent uninstall from the installation directory, not the installation media.

Editing the uninstall parameters

SilentUninstall.txt contains parameters that tell the uninstaller what components you want to remove. *SilentUninstall.txt* is configured to remove all components by default; removing specific components requires additional configuration.

SilentUninstall.txt contains the same default selection and configuration options as the interactive uninstaller. If you only want to uninstall some components, edit the file, and change the default parameters:

- Features with parameters set to *true* will be uninstalled

- Features with parameters set to *false* will not be uninstalled

If you modify the file, read the comments for the section(s) you want to change, update the parameters, then save *SilentUninstall.txt* to the local file system. See "SilentUninstall.txt" on page 121 to review the parameters in this file.

Running the uninstall

Use *SilentUninstall.sh* to remove some or all IQ components from the system. How you run the uninstall depends on whether you want to run a *complete* or *partial* uninstall.

Uninstalling all components

If you installed IQ with all defaults, you can run a typical silent uninstall directly from the installation directory. To run the uninstall, open a console, and use the following command:

```
$SYBASE/uninstall/IQ/SilentUninstall.sh
```

Uninstalling some components

To perform a custom uninstall, the command you use to run *SilentUninstall.sh* must include the *–txt* switch that points to the location of the modified *SilentUninstall.txt*:

```
$SYBASE/uninstall/IQ/SilentUninstall.sh -txt
<directory name>\SilentUninstall.txt
```

Uninstalling IQ components

SilentUninstall.bat is the name of the script that you use to uninstall IQ components. How you run the uninstall depends on whether you are "Uninstalling all components" on page 116 or "Uninstalling some components" on page 116.

Uninstalling all components	If you installed IQ with all defaults, you can run a typical silent uninstall directly from the installation directory. To run the uninstall, open a command window, and use the following command:

```
%SYBASE%\uninstall\IQ\SilentUninstall.bat
```

Uninstalling some components	To perform a custom uninstall, the command you use to run *SilentUninstall.bat* must include the *–txt* switch that points to the location of a modified *SilentUninstall.txt*:

```
%SYBASE%\uninstall\IQ\SilentUninstall.bat -txt
<directory name>\ SilentUninstall.txt
```

Running silent uninstall scripts on Microsoft Vista	You must have sufficient user permissions to run applications in Microsoft Vista. If you do not have administrator privileges and want to run a silent uninstall, open a command prompt as an Administrator, then execute the silent uninstall from the command line.

Sample configuration files

Use this section to review the settings in the configuration files, SilentInstall.txt and SilentUninstall.txt.

SilentInstall.txt

Use this example to review the server settings in a typical *SilentInstall.txt* file. Client versions follow the same conventions.

```
#############################################################################
############################ SILENT INSTALLER ###############################
#############################################################################

#############################################################################
 # IMPORTANT NOTE BEFORE STARTING (If you have previously installed IQ):
 #  1. Ensure IQ Server is not running.
#############################################################################

#############################################################################
 # By setting the following argument to "true", you are agreeing to the
 # Sybase License Agreement.
#############################################################################
 -V AgreeToSybaseLicense="false"

#############################################################################
 # By setting the following argument to "false" you wish to use the Evaluation
 # version.  By setting the following argument to "true" you wish to use the
 # Licensed Version.  You can specify a license file below if you select a
 # licensed version.
#############################################################################
 -V Variable_IQDemoSelect_LicensedYesRB="true"

#############################################################################
 # NOTE: The following two variables apply to WINDOWS Platform Only
 #
 # The install will try to detect if Sybase Open Client is in the environment.
 # To prevent conflicts, IQ will append the IQ executable directory to the end
 # of the search PATH when it finds Sybase Open Client in the environment.
 #
 # In rare cases, appending to the end of the PATH can fail due to the length
 # of the PATH exceeding the maximum system length, or if an invalid or corrupt
 # entry preceeds IQ.  If IQ is missing from the path after the install
 # completes, please shorten or clean the current PATH and then reboot before
 # attempting a reinstall.
```

```
 #
 # If you wish to modify the install behavior you can uncomment and modify
 # the two variables below:
 #     Variable_OverridePrependOrAppendToPath"true" to override install
 #                     behavior
 #                     "false" to use install
 #                     behavior
 #     Variable_IQAppendPath_Prepend_IQBinDir"true" to prepend
 #                     "false" to append
###########################################################################
 ### -V Variable_OverridePrependOrAppendToPath="false"
 ### -V Variable_IQAppendPath_Prepend_IQBinDir="true"


###########################################################################
 #
 # Sybase Software Asset Management License
 #
 # Sybase Data Integration Suite requires Sybase Software Asset Management
 # (SySAM) license key(s). If you have not yet obtained the license key(s), go
 # to
 # SPDC at http://sybase.subscribenet.com to obtain the license keys.
 #
 # You can specify the license file or license server where the license key(s)
 # were previously deployed.
 #
 #    -W sysamLicense.proceedWithoutLicense    - Select this if you want to
 #                                               continue installation without
 #                                               a
 #                                               license key.
 #    -W sysamLicense.useExistingLicenseServer - Valid values are "true" or
 #                                               "false". Set to "true" if you
 #                                               want to use previously
 #                                               deployed
 #                                               license server or "false" if
 #                                               want to specify the license
 #                                               file.
 #    -W sysamLicense.licenseFile              - Specify the Sybase Software
 #                                              Asset Management license file.
 #    -W sysamLicense.licenseServerHostName    - Enter the hostname where the
 #                                               license server was deployed.
 #    -W sysamLicense.licenseServerPortNumber  - Enter the license server port
 #                                               number. Leave the entry blank
 #                                              if you want to use the default
 #                                               port numbers.
 #
```

```
 -W sysamLicense.proceedWithoutLicense=true
### -W sysamLicense.useExistingLicenseServer=false
### -W sysamLicense.licenseFile=
### -W sysamLicense.licenseServerHostName=localhost
### -W sysamLicense.licenseServerPortNumber=

##############################################################################
 #
 #
 # Please select the product edition you would like to configure. For Enterprise
 # and Small Business Editions, you also need to select the license type for this
 # installation.
 #
 #     sysamProductType.productEdition - Valid values are "Enterprise Edition",
 #                                       "Small Business Edition", "Single
 #                                       Application Edition", or "Unknown".
 #     sysamProductType.licenseType    - Valid values are "CPU License (CP)",
 #                                       "Standby CPU License (SF)", "Development
 #                                       and Testing (DT)", "OEM Application
 #                                       Deployment CPU License (AC)", "OEM
 #                                       Application Deployment Standby CPU
 #                                       License (BC)", or "Unknown".
 #

 -W sysamProductType.productEdition="Enterprise Edition"
 -W sysamProductType.licenseType="Development and Testing (DT)"

##############################################################################
 # Sybase Software Asset Management email alerts
 #
 # If you would like to configure email alerts, set
 # -V Variable_IQSySAMEmail_AlertsYesRB=true
 # -V Variable_IQSySAMEmail_AlertsNoRB=false
 # and uncomment and set the other variables appropriately.
 #
 # Valid values for -V Variable_IQSySAMEmail_SeverityTF= are:
 # WARNING, INFORMATIONAL, and ERROR
##############################################################################

 -V Variable_IQSySAMEmail_AlertsYesRB=false
 -V Variable_IQSySAMEmail_AlertsNoRB=true
 # -V Variable_IQSySAMEmail_HostTF=smtp
 # -V Variable_IQSySAMEmail_PortTF=25
 # -V Variable_IQSySAMEmail_SenderTF=" "
 # -V Variable_IQSySAMEmail_RecipientTF=" "
 # -V Variable_IQSySAMEmail_SeverityTF=" "
```

```
############################################################################
 # Enable/Disable features to be installed by setting the value to
 # "true" or "false".
 #
 # Legal values are:
 #    true  - Indicates that the feature is selected for installation
 #    false - Indicates that the feature is not selected for installation
 #
 # In order to install a sub feature, all parent features must also be
 # selected.
 #
 # Note: You will need to remember these same features if running the
 #    Silent Uninstall.
 #
 # For example, to select "Server:" for installation, use
 #
 #    -P Server.active=true
 #
############################################################################

 # "SybaseIQ" Feature
 -P SybaseIQ.active="true"
      # "IQServer" Feature
      -P IQServer.active="true"
         # "IQOpenClient" Feature
         -P IQOpenClient.active="true"
      # "IQClient" Feature
      -P IQClient.active="true"
      # "IQODBC" Feature
      -P IQODBC.active="true"
 # "SybaseCentral" Feature
 -P SybaseCentral.active="true"
      # "SCJViewer" Feature
      -P SCJViewer.active="true"
      # "SCJPlugin" Feature - requires SCJViewer to be installed
      -P SCJPlugin.active="true"
 # "jConnect" Feature
 -P jConnect.active="true"
 # "SybaseSySAM" Feature
 -P SybaseSySAM.active="true"
      # "NetworkLicenseServer" Feature
      -P NetworkLicenseServer.active="false"
      # "LicenseUtilities" Feature
      -P LicenseUtilities.active="true"
```

```
##############################################################################
 # Please do not modify the values below.
##############################################################################
 -V Variable_Silent="true"
 -W InstallType.active="false"
 -P copyJRE.active="true"
 -P SilentUninstall.active="true"
 -P copyJRE1.active="true"
 -P SilentUninstall1.active="true"
 -silent
```

SilentUninstall.txt

Use this example to review the server settings in a typical *SilentUninstall.txt* file. Client versions follow the same conventions.

```
##############################################################################
########################### SILENT UNINSTALLER ###############################
##############################################################################

##############################################################################
 # Enable/Disable features to be uninstalled by setting the value to
 # "true" or "false".
 #
 # Legal values are:
 #    true  - Indicates that the feature is selected for removal
 #    false - Indicates that the feature is not selected for removal
 #
 # For example, to select "Server:" for removal, use
 #
 #    -P Server.activeForUninstall=true
 #
##############################################################################

 # "SybaseIQ" Feature
 -P SybaseIQ.activeForUninstall="true"
      # "IQServer" Feature
      -P IQServer.activeForUninstall="true"
         # "IQOpenClient" Feature
         -P IQOpenClient.activeForUninstall="true"
      # "IQClient" Feature
      -P IQClient.activeForUninstall="true"
      # "IQODBC" Feature
```

```
      -P IQODBC.activeForUninstall="true"
 # "SybaseCentral" Feature
 -P SybaseCentral.activeForUninstall="true"
      # "SCJViewer" Feature - removal of SCJViewer requires that SCJPlugin
      # be removed as well
      -P SCJViewer.activeForUninstall="true"
      # "SCJPlugin" Feature
      -P SCJPlugin.activeForUninstall="true"
 # "jConnect" Feature
 -P jConnect.activeForUninstall="true"
 # "SybaseSySAM" Feature
 -P SybaseSySAM.activeForUninstall="true"
      # "NetworkLicenseServer" Feature
      -P NetworkLicenseServer.activeForUninstall="false"
      # "LicenseUtilities" Feature
      -P LicenseUtilities.activeForUninstall="true"

##########################################################################
 # Please do not modify the values below.
##########################################################################
 -G removeExistingResponse=yesToAll
 -G removeModifiedResponse=yesToAll
 -silent
```

APPENDIX B   **Troubleshooting SySAM**

About this Chapter

If SySAM 2 related errors are not addressed in a timely fashion, Sybase IQ may cease to function. This section includes information about troubleshooting the most common SySAM related errors.

For the most up to date information about troubleshooting SySAM, see the SySAM FAQ at http://www.sybase.com/products/allproductsa-z/sysam.

Contents

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 command. 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 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.

You can also execute sp_iqlmconfig to determine the expiration date. The Expiry Date column next to any row that shows graced status indicates the date and time the grace period expires. Sybase IQ disables the functionality corresponding to the license if the issue is not fixed within that time. Sybase IQ itself shuts down if the graced license is *IQ_CORE.*

Enabling and changing e-mail notifications

You can configure the e-mail notification setup using sp_iqlmconfig. Run sp_iqlmconfig command 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

If you call Sybase technical support for SySAM-related issues have the following information available:

- The Sybase IQ error log located in *$IQDIR15/logfiles/<servername>.stderr* directory.

- 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 – the server's properties file in the *$IQDIR15/logfiles/<servername>.properties* .

- 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/licenses* 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.

Diagnosing SySAM issues

Refer to the following table to locate specific SySAM problems.

Problem Descriptions

"Cannot find license type during the Sybase IQ installation" on page 126	"Sybase IQ cannot check out a license and starts with graced license instead" on page 129
"The installation program produces a warning about not finding a suitable license" on page 127	"Sybase IQ shows graced licenses even after the issue has been fixed" on page 131
"The license server executables and scripts are not installed" on page 127	"Sybase IQ cannot find license for an optional feature, even though the license exists" on page 131
"The license server does not start" on page 128	"Sybase IQ does not start with the expected edition or license type" on page 131
"The license server does not recognize the license files" on page 128	"Sybase IQ does not pick the correct license" on page 132
"Sybase IQ does not start, and generates a license check-out failure error" on page 129	

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.

The installation program produces a warning about not finding 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. For more information, see "Sybase IQ cannot check out a license and starts with graced license instead" on page 129.

You can determine the date based version of the license from the Version column of the output from sp_iqlmconfig; by looking at the license checkout message in the Sybase IQ error log and *<dbname>.iqmsg file*; or by examining the license itself. For more information see Appendix A Anatomy of a License in the *Sybase Software Asset Management Users Guide*.

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.

See also
- See Appendix B of the *Sybase Software Asset Management Users Guide* for an explanation of how SySAM uses date-based versioning.

- See Updating your Sybase licenses in Chapter 3 Getting and Using your Licenses, in the *Sybase Software Asset Management Users Guide* for how to update your license.

The 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.

The 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 for license server not to start 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.

The license server does not recognize the license files

If the license server refuses to serve the licenses, this may be because:

- 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 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 cannot check out a license and starts with graced license instead in this table. The troubleshooting tips for resolving licenses issued under grace apply to solve this issue as well.

Sybase IQ cannot check out a license and starts with graced license instead

Sybase IQ cannot check out a license and starts with graced license instead.

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 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 Appendix B, 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 Chapter 3, Getting and Using your License, 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.

See "Sybase IQ cannot find license for an optional feature, even though the license exists" on page 131 for more information.

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. They must match. This field is not considered when searching for a license without active and standby capability.

Sybase IQ does not start with the expected edition or license type

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

When Sybase IQ looks for a license, it does so in the following locations:

- Any saved values in the registry on Windows platforms, or the *.flexlmrc* file on UNIX platforms. Use 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

Symbols

$SYBASE variable   19
   setting   19

A

Adaptive Server Enterprise
   inserting data from   105
adding startup switches   58
after upgrading
   updating configuration files   94
autostarting database   63
AUTOSTOP parameter   103

B

backing up databases
   after upgrade   94
   procedure   94
BLANK PADDING
   migration   68
blank users   66

C

cache sizes   61
catalog cache   61
changing hardware platforms   51
client
   uninstalling   47
client and server, running in same system   106
client components   44
   installing   43
   jConnect   44
   ODBC   44
   Sybase Central   44

   Sybase IQ client   44
client installation
   client components   44
   disk space and memory requirements (Windows)
      46
   installer   45
   installer, Windows   46
   overview   44
   parent installation directory   45
   prerequisites   44
   running the installer   47
   setting environmental variables   46
   Sybase IQ installation directory   45
   typical installation   45
COMMLINKS parameter   103, 107
configuration files   98
   security   98
configuration parameters
   overriding   38
   set by **start_asiq**   38
connecting
   across firewalls   107
   using OLE DB   104
   with JDBC   99
   with ODBC   100
connection parameters   103
connections
   restricting   52
constraints
   migration   68
   unenforced   68
conventions
   documentation   xiii, xiv
   syntax   xiii
   typographic   xiv
creating
   interfaces file entries   105
   ODBC data sources   102