StarSQL™ for Windows User's Guide

Version 6.4

StarQuest

Statement of Limitations on Warranty & Liability

StarQuest Ventures, Inc. makes no representations or warranties about the suitability of the software and documentation, either expressed or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. StarQuest Ventures shall not be liable for any damages suffered by licensee as a result of using, modifying, or distributing this software or its derivatives.

StarSQL™ is a trademark of StarQuest Ventures, Inc. All trademarks or registered trademarks are the property of their respective owners.

© Copyright 1996–2019 by StarQuest Ventures, Inc.
All rights reserved.

v6.4_12.23.2019

Contents

CHAPTER 1

Introduction

StarSQL is available as an ODBC driver for Windows- and UNIX-based computers, and as a JDBC driver for any computer that has the Java Runtime Engine (JRE) or Java Virtual Machine (JVM) installed. StarSQL includes a 64-bit version of the driver so you can unleash the power of 64-bit operating systems and database applications.

The StarSQL software is not copy protected, rather the usage is limited based upon the maximum number of concurrent connections ("CCs") licensed. The StarQuest license allows for you to install and run any of the StarSQL drivers on any number of client computers, subject to terms of the license grant. CCs may be made available for clients on a single computer ("Node-locked License") or any computer on the network ("Floating License"). A license to use StarSQL allows use of the 32-bit and 64 bit ODBC driver or the StarSQL for Java JDBC driver.

The StarSQL for Windows Driver

The StarSQL driver resides on the Windows computer and enables ODBC applications to transparently access data that resides in a DB2 database on a host computer.

The StarSQL driver performs the following functions to enable ODBC applications to communicate with a DB2 host:

- provides a programmatic interface for executing dynamic SQL functions
- provides transparent translation of character encoding between different client and host operating systems

Using your choice of ODBC-enabled applications, the Windows-based computer uses the StarSQL driver to establish a connection to a DB2 host. The connection to the host system can be via TCP/IP or SSL, or via a StarQuest StarPipes Gateway. This document focuses on using StarSQL over a TCP/IP network.

StarSQL for Windows Features

StarSQL for Windows provides the following major features:

- support for both 32-bit and 64-bit architectures
- support for LOB (large object) data types
- support for global (two-phase) transactions
- support for encrypted passwords
- support for SSL encrypted connectivity
- international language support, including enhanced support for East Asian multi-byte and double-byte character sets
- utilities for recording and viewing DRDA traces
- support for the latest versions of DB2 and Windows

StarSQL Includes 32-bit and 64-bit Versions

StarSQL includes the 32-bit driver and a 64-bit ODBC driver that can take advantage of Windows x64 architecture. You can install and run both the 32-bit and 64-bit version of the StarSQL driver on the same 64-bit Windows computer. Install and run only the 32-bit version of the StarSQL driver on a 32-bit Windows computer. "Installing StarSQL" on page 18 contains details about installing one or both versions of the driver software.

LOB Support

Large objects (LOBs) can be stored directly in the DBMS, such as to manage objects such as text documents, images, or streaming video. The StarSQL for Windows driver supports working with DB2 LOB data types. See "System Requirements" on page 10 for details about which database versions support LOB data types. If you have applications or hosts that do not support LOB data types, set the LongStrParams expert setting in the StarSQL data source to provide backwards compatibility. The help system for the StarSQL Data Source Configuration Wizard provides details about the LongStrParams setting.

Global Transaction Support

In a distributed processing environment where a unit of work spans more than one database, using the two-phase commit protocol can help ensure the integrity of the transaction and databases. When the two-phase commit protocol is in effect, the transaction must complete successfully before it is committed. If the transaction fails, all of the updates are rolled back and the databases remain in the state they were before the transaction was begun. StarSQL for Windows includes a Transaction Log Manager application to help you manage two-phase commit transactions over a TCP/IP network, as described in "Supporting Two-Phase Commit" on page 52.

StarSQL 6.4 introduces support for XA Two-Phase transactions

Encrypted Password Support

The default behavior of StarSQL for Windows is to send passwords in clear text to the host computer. Many versions of DB2 support password encryption or successfully negotiate the use of password encryption (see "System Requirements" on page 10 for details about which versions). If the host database supports encrypted passwords, you can enable the password encryption feature as described in the StarSQL Help that is accessible from the StarSQL Data Source Configuration Wizard.

SSL Encryption

Secure Sockets Layer (SSL) is a protocol that provides a secure connection over an insecure network. It is commonly used by Internet applications such as web-based banking and commerce; whenever you see a URL beginning with https, you know that your web browser is communicating using SSL. SSL can also be used to secure DRDA connections, and is supported by recent versions of DB2 (see "System Requirements" on page 10). If the host is not enabled for SSL, you can use a StarQuest StarPipes Gateway to provide SSL support.

International Language Support

StarSQL uses a data-driven architecture to support data character conversions, which allows support for specific languages and character encoding schemes to be added without requiring any changes in the StarSQL driver source code.

DRDA Tracing Utilities

StarSQL for Windows includes a DRDA Trace Recorder application that allows you to capture trace information about the DRDA communications. A StarSQL DRDA Trace Viewer utility also is provided for viewing the trace file output.

StarSQL System Overview

StarSQL is a Unicode driver that is compliant with the ODBC v3.0 specification, and is backwards compatible with IBM DB2 v2 CLI. The StarSQL driver software is installed on the client computers to allow access to data that resides in a DB2 database on a host computer. StarSQL requires a physical network and network software for communication between the desktop and the host.

Before setting up StarSQL for Windows on the desktop, it is necessary to prepare both the host and the network as described in "Preparing Hosts for StarSQL Access" on page 33. You will need information about the network and the host databases to configure data sources after StarSQL is installed.

Because the StarSQL configuration involves the desktop, the network, and the host, several individuals may be involved with setting up the StarSQL environment. In a large organization, this might include a database and/or network administrator, systems programmer, and applications developer.

System Requirements

The StarSQL **32-bit** ODBC driver is supported on the following 32-bit Windows platforms:

- Windows 7, 8, 8.1, 10

The StarSQL **32-bit and 64-bit** ODBC drivers are supported on all the following 64-bit platforms:.

- Windows Server 2008R2, 2012, 2012R2, 2016, 2019
- Windows 7, 8, 8.1, 10 x64 Edition

If the computer will run applications that participate in the two-phase commit process, refer to "Supporting Two-Phase Commit" on page 52 for details about the system requirements.

Host Connectivity

The StarSQL ODBC driver works within a TCP/IP network to connect a Windows client directly to a DB2 host. You can also use a SSL (Secure Socket Layer) connection when supported by the host database, or in conjunction with a StarPipes Gateway. .

Host Databases

You can use the StarSQL driver to connect to any of the following host databases:

- DB2 z/OS v8.1 and later
- DB2 for i (formerly known as DB2/400, DB2 UDB for iSeries, and DB2 for i5/OS) running IBM i (i5/OS) V5R4 and later
- DB2 for Linux, UNIX, and Windows (DB2 for LUW) v8.2 and later

Refer to the StarSQL Readme file for additional information about other PTFs or fixes that may be required for proper operations.

Documentation

StarSQL for Windows includes the following documentation in addition to this *User's Guide.* You can access most of the documentation components from the Information submenu of the StarSQL program group. The online help systems are Windows-based compiled help files (.chm) that are installed in the \StarSQL\Programs directory if you choose to install all of the StarSQL components. The online help can be accessed from the respective software component by pressing F1, clicking a Help button, or selecting Help Topics from the About menu if there is a menu bar.

Table 1. StarSQL for Windows Documentation

Documentation Component	**Description**
DRDA Trace Viewer Help	Filename: Trcview.chm Available from the StarSQL DRDA Trace Viewer utility.
StarQuest License Configuration Help	Filename: starlic.chm Available from the StarQuest License Configuration application.
Resource Manager help	Filename: sqrmMMC.chm The StarSQL Resource Manager is available from the Services and Applications item of the Microsoft Management Console. The Resource Manager help is embedded in the Microsoft Management Console help.

Documentation Component	Description
StarSQL Help	Filename: sqwizard.chm Available from the StarSQL Program Group and from the StarSQL Data Source Configuration Wizard. The StarSQL Wizard appears when you add or edit a StarSQL data source using the ODBC Data Source Administrator.
Release Notes	Filename: readme.html The Release Notes contain important information about installing and using StarSQL in specific environments, known limitations, and a history of changes to the driver.

In addition to the StarSQL product documentation, The StarQuest Info Center provides StarSQL Quick Start Guides (go to http://www.starquest.com/docs/Supportdocs/browseQuickStarts.shtml) that provide step-by-step instructions for quickly installing and using the StarSQL ODBC and JDBC drivers. The Info Center also provides technical documents for specific functions and issues.

Contacting StarQuest

Please use the following methods to contact StarQuest Ventures if you need to obtain a license key, or have suggestions or need information about StarQuest products.

Support

If you do not want to use the online licensing feature of the License Configuration utility (see "Licensing StarQuest Products" on page 21) , you can obtain a license key for your product by sending an email to support@starquest.com with the following information:

- TCP/IP address or Host ID of the computer on which the license will be installed
- Number of connections purchased
- Company Name
- Contact Name

- Phone Number
- Email Address

StarQuest Support will send a reply email that provides the license key for your organization's use of the product. Since the license is unique to the computer on which it will be installed, you must contact StarQuest should you need to move the license from one computer to another.

Additional technical support may be available subject to the prices, terms, and conditions specified in your organization's maintenance contract with StarQuest Ventures, Inc.

Sales and Service

If you have ideas for product enhancements or need more information about StarQuest products, please contact us via any of the following methods.

Address	StarQuest Ventures, Inc. 548 Market St, #22938 San Francisco, CA 94104-5401
Telephone	415-669-9619 Option 1: Sales Option 2: Technical Support
Fax	415-669-9639
Email	support@starquest.com
World Wide Web	www.starquest.com

CHAPTER 2

Installing StarSQL for Windows

This chapter describes how to install StarSQL for Windows. It addresses:

- Migration issues applicable if you are upgrading from a prior version
- Installing the 32-bit version of StarSQL and/or the 64-bit version
- Licensing StarSQL
- Deploying StarSQL to multiple desktops

Be sure to review the Release Notes included in the distribution for important information about installing or upgrading the StarSQL for Windows driver.

Upgrading StarSQL

Versions of the StarSQL software are categorized as major releases, point releases, and hot-fixes. Major releases usually are designated with a new major release number such as v4.1, v5.3, or v5.5. Point releases provide defect corrections to a major release, and are designated with a version number such as v5.21, v5.34, 5.51 and so on. If a particular problem arises between scheduled releases, one or more of the StarSQL components may be provided as a "hot-fix" until the fix is incorporated into a point release or a major release.

To upgrade from StarSQL5.51 or v6.0, you can just run the `setup.exe` program from the StarSQL installer image as described in "Installing StarSQL" on page 18. To upgrade from StarSQL 5.0 through 5.50, use the Add/Remove Control Panel to uninstall the older version of StarSQL. To upgrade from a v2.x, 3.x, or 4x version of StarSQL, choose Uninstall StarSQL from the StarSQL Program Group to un-install the older version before you install StarSQL v6.1.

Major releases of StarSQL also may require changes to SQL catalog packages on the host. StarSQL uses the SQL packages to provide added functionality. (Point releases and hot-fixes usually do not involve changes to the SQL packages.) It is recommended that the SQL packages be rebound whenever introducing a major version of StarSQL into the computing environment or when directed by the instructions included in the Readme file associated with all StarSQL installation images.

Significant changes were made to catalog packages and dynamic SQL packages with the v5.1 and v5.3 releases of StarSQL. Users may get different results from ODBC catalog functions when using a version of StarSQL prior to v5.1 with packages that are bound with StarSQL v5.1 or later.

Table 2 shows the level of functionality supported with SQL packages created by different versions of StarSQL. In general, packages bound by later versions of StarSQL are backward compatible with earlier versions of StarSQL, unless specifically noted. If an earlier version of StarSQL cannot function properly with packages that are bound with a later release of StarSQL, you can maintain separate package collections or upgrade all clients to use the latest version of the driver.

Table 2. Functional Package Differences Among StarSQL Versions

From	To	New Functionality
v4	v5.1	Packages must be rebound to use LOB data types. Upgrade all clients or maintain a separate package collection for clients running a StarSQL release prior to v5.1
v5.1	v5.3	Rebind packages to support use of jumbo packages, or to access DB2 for z/OS when the host is configured to use a multi-byte character set. See the topics for UseJumboPackages and CustomizePrdid in the StarSQL help for more information about these expert settings.
v5.31	v5.37	Rebind packages if you upgrade from a release earlier than v5.31. Client computers that use StarSQL v5.1 or earlier should use a different package collection.
v5.37	v5.6x	Clients running StarSQL 5.38 through StarSQL 5.6x can share package collections.
v6.0		Clients running StarSQL 6.0 and later require an upgraded package collection.

You can use the StarAdmin utility (available as a separate download) to rebind all packages, or you can drop the packages on the host and reconnect with the new version of StarSQL.

Data Type Mapping Differences

StarSQL v5 added support for DB2 LOB data types. The DB2 types are mapped to ODBC types as shown in Table 3:

Table 3. Mapping of DB2 Data Types to ODBC Data Types

DB2 Type	StarSQL v5 ODBC Type
BLOB	SQL_LONGVARBINARY
CLOB	SQL_LONGVARCHAR
DBCLOB	SQL_LONGVARCHAR

As shown in Table 4, StarSQL changes the mapping for DB2 long strings—they are no longer differentiated from short strings:

Table 4. Mapping of DB2 Strings

DB2 Type	StarSQL v4.x ODBC Type	StarSQL v5 ODBC Type
VARCHAR	SQL_VARCHAR	SQL_VARCHAR
LONG VARCHAR	SQL_LONGVARCHAR	SQL_VARCHAR
VARGRAPHIC	SQL_VARCHAR	SQL_VARCHAR
LONG VARGRAPHIC	SQL_LONGVARCHAR	SQL_VARCHAR
VARCHAR FOR BIT DATA	SQL_VARBINARY	SQL_VARBINARY
LONG VARCHAR FOR BIT DATA	SQL_LONGVARBINARY	SQL_VARBINARY

These mappings affect ODBC catalog query results. For example, SQLColumns for a DB2 LONG VARCHAR returns ODBC type SQL_VARCHAR for a package bound by StarSQL v5. If StarSQL v4.x uses packages bound by StarSQL v5, catalog queries will return the new StarSQL v5 results, but otherwise the driver should run as before.

Parameters for SQL statements are also affected. If an application binds a parameter as SQL type SQL_LONGVARCHAR, it is sent as a DB2 CLOB. StarSQL v4.x sends SQL_LONGVARCHAR as a DB2 VARCHAR string.

StarSQL has an optional data source setting, LongStrParams (see the LongStrParams topic of the StarSQL Data Source Configuration Wizard help system), for applications that require backwards compatibility for SQL_LONGVARCHAR (or SQL_LONGVARBINARY) parameters. This data source setting does not affect the types returned for result set columns, or for types returned by catalog queries.

StarSQL no longer returns type SQL_FLOAT from SQLGetTypeInfo. StarSQL maps DB2 REAL to SQL_REAL, and DB2 DOUBLE to SQL_DOUBLE, and only these two floating point types are returned by SQLGetInfo. StarSQL still accepts SQL_FLOAT, as a synonym for SQL_DOUBLE, as a type, for example, for SQLBindCol.

For an introduction to LOB support in DB2 for z/OS, see the IBM Redbook "Large Objects with DB2 for z/OS and OS/390" (SG24-6571), available at http://www.redbooks.ibm.com.

Installing StarSQL

StarSQL includes the 32-bit driver and a separate 64-bit ODBC driver that can take advantage of Windows x64 architecture. You can install either or both the 32-bit and 64-bit versions of the StarSQL ODBC driver on the same 64-bit Windows computer. Install and run only the 32-bit version of the StarSQL driver on a 32-bit Windows computer. If you want to install only the 32-bit version of StarSQL, skip to the section "Running the Setup Program".

StarSQL Architecture Considerations

You can install only the 64-bit version, or both the 32-bit and 64-bit versions of the StarSQL ODBC driver on the same 64-bit Windows computer. The versions operate independently of each other, using data sources that are configured explicitly for either the 32-bit or 64-bit environment.

Table 5 summarizes how the 32-bit and 64-bit versions of StarSQL for Windows differ. As indicated in the table, data sources that are defined for use by the 32-bit architecture cannot be used by 64-bit applications and vice-versa.

There is a 32-bit and a 64-bit version of the ODBC Administrator. You must configure the data source names (DSNs) using the version of the ODBC Administrator that is appropriate for the applications that will use the DSN. StarSQL (32-bit) uses DSNs created by the 32-bit version of ODBC Administrator, and StarSQL (64-bit) uses DSNs created by the 64-bit version of ODBC Administrator. Selecting ODBC Administrator from the StarSQL or StarSQL 32 program group starts the 32-bit ODBC Administrator and selecting the shortcut from the StarSQL (64-bit) program group starts the 64-bit version of ODBC Administrator.

Table 5. Comparison of 32-bit and 64-bit StarSQL ODBC Drivers

Feature	StarSQL 32-bit	StarSQL 64-bit
Windows Program Group Name (x86)	StarSQL	not applicable
Windows Program Group Name (x64)	StarSQL (32-bit)	StarSQL (64-bit)
Driver Name (x86)	StarSQL 32	not applicable
Driver Name (x64)	StarSQL 32	StarSQL (64-bit)
Default Installation Directory on Windows x86	Program Files\StarQuest\StarSQL	not applicable
Default Installation Directory for Windows x64	Program Files (x86)\StarQuest\StarSQL	Program Files\StarQuest\StarSQL
Data Sources for Windows x86	Uses DSNs configured with 32-bit ODBC Administrator (Windows\System32)	not applicable
Data Sources for Windows x64	Uses DSNs configured with 32-bit ODBC Administrator (Windows\SysWOW64)	Uses DSNs configured with 64-bit ODBC Administrator

Running the Setup Program

The StarSQL installation package contains two subdirectories, x86 and x64, that contain the files for installing the 32-bit (x86 subdirectory) and 64-bit (x64 subdirectory) versions of the driver. Change to the appropriate subdirectory and run the **setup.exe** program in that directory to install the corresponding version of StarSQL.

The Setup program displays dialog boxes that guide you through the installation. This section describes the installation options to help you respond to the choices presented during the installation. After you successfully install and configure StarSQL for Windows on one computer, you can replicate the data source configuration to other computers as described in "Sharing Data Source Definitions" on page 27.

The Setup program provides a choice of installing StarSQL as a Typical installation or a Custom installation. The Typical installation option installs the following components:

- Driver core
- TCP/IP and SSL network connectivity
- DRDA Trace Recorder
- Connectivity for all supported hosts
- License Utility
- StarSQL Help

If you choose the Custom installation option, you can install the following components *in addition* to those installed with the Typical installation:

- support for two-phase commit transactions (this option does not appear when installing the StarSQL 32-bit driver on a Windows x64 computer)
- the DRDA Trace Viewer

The Custom installation lists the major components and you can select and de-select specific subcomponents that you want to install.

Licensing StarQuest Products

All StarQuest products are licensed for use. Each product setup contains a client module used to configure the specific license option used to enforce the use of the product. The licensing options allow you to use a node-locked license or a floating license.

Node-Locked License

A node-locked license allows you to use the Product on a single computer: Node-locked licenses are only available for computers using Microsoft Windows Operating Systems. With a node-locked license:

- The computer is identified by a unique Host ID.
- The product can run only on the identified computer.
- The product usage may not exceed the limits allowed by the license.

In addition to the setup of the StarQuest product you will be provided with a unique registration code that should be used for the activation of the software license. It is also possible to use the client module to display the HOSTID to request a software license via email or telephone.

The software license should be activated online using the supplied registration code, or manually after communicating a HOSTID with StarQuest and receiving an email response containing a license string.

Floating License

A floating license allows multiple computers using a StarQuest product to share use of the software license. The software license can be used on any computer within a network provided that the number of concurrent requests does not exceed the limit allowed by the license. All StarQuest products for UNIX and Mac OS X ***must*** use a floating license. StarQuest products for Windows ***may*** use a floating license. Generally, only one license server need be installed on a network, to service any number of clients.

For a floating license, in addition to the setup of StarQuest product you will be provided with a StarLicense Server setup and a unique registration code to activate the license server. The *StarLicense Server for UNIX User's Guide* contains details about installing and configuring the server software, but in general:

- The StarLicense Server software should be installed on a network server.
- The network server is usually identified by a unique, static IP address.

- The StarLicense Server should be activated online or via e-mail.
- The StarLicense Server controls the total number of concurrent connections within the network.

For a client to obtain a license from a StarLicense server the parameters of the network server where the StarLicense Server is installed are specified in the appropriate client license module on any computer using the StarQuest product.

The StarPipes Gateway can also be configured to function as license server.

Configuring a License

After the StarSQL for Windows software is installed, a License Configuration dialog appears so you can enter a node-locked license key or specify which StarLicense server(s) you want to use. The 32-bit and 64-bit versions of StarSQL use the same license when installed on the same Windows x64 computer.

Refer to the appropriate section below, depending on whether you want to configure the client to use a floating license on a StarLicense server within the same TCP/IP network or use a node-locked license on the local computer.

Configuring a Client to Use a Node-Locked License

You can use the online licensing feature to obtain and enter a node-locked license key to the License Configuration, or you can manually enter a license key, as described in the following sections.

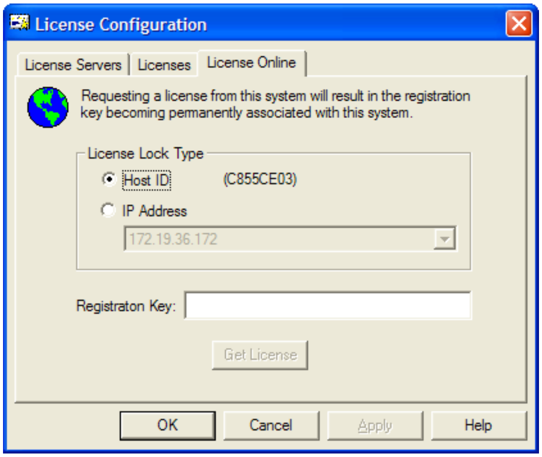
Obtaining a License Key Using the Internet

The License Online tab of the License Configuration dialog provides the easiest method for licensing StarQuest products. From the License Online tab you use the Internet to request and add a license key to the computer on which the License Configuration program is running. Request the license from the computer that will use the license, and have the Registration Key for the product available. The Registration Key is sent to the designated contact via email.

To obtain a license key you need to choose whether you want to associate the license with a static TCP/IP address or the Host ID of the computer on which the License Configuration program is running, as shown in Figure 1. Click on the radio button that reflects the type of license lock you want to use, enter the Registration Key that StarQuest Support provided, and click the **Get License** button to send the request.

When the request successfully completes, the license(s) for the software you are registered to use appear in the License Keys list of the Licenses tab. The Registration Key may produce several License Keys, depending on the products you are registered to use.

Figure 1. Licensing StarQuest Products using the Internet



License Configuration

Requesting a license from this system will result in the registration key becoming permanently associated with this system.

License Lock Type
● Host ID (C855CE03)
○ IP Address
172.19.36.172

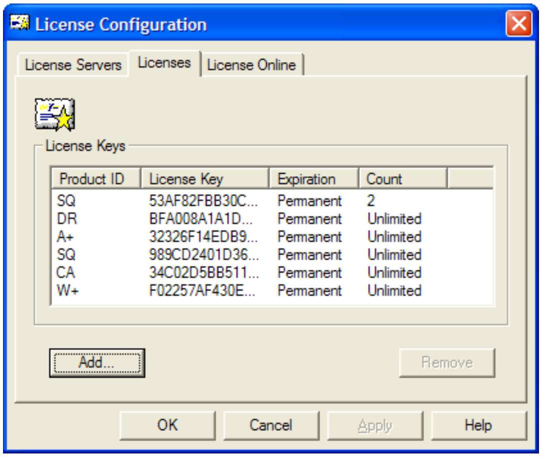
Registraton Key:

You also can request a license via email from StarQuest Customer Support. See "Support" on page 12 for the address and a list of the information you need to supply.

Manually Configuring a Client License

The License Configuration dialog also lets you add a License Key for the computer on which the License Configuration program is running. If the License Configuration dialog is not already active, select the **License Configuration** shortcut from the **StarSQL** program group, and click the Licenses tab of the License Configuration dialog.

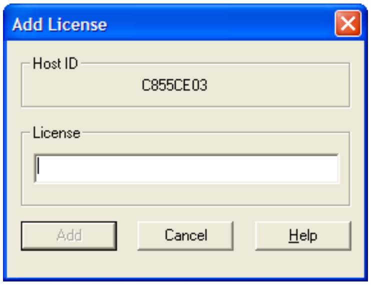
Figure 2. Manually Adding a License Key



Product ID	License Key	Expiration	Count
SQ	53AF82FBB30C...	Permanent	2
DR	BFA008A1A1D...	Permanent	Unlimited
A+	32326F14EDB9...	Permanent	Unlimited
SQ	989CD2401D36...	Permanent	Unlimited
CA	34C02D5BB511...	Permanent	Unlimited
W+	F02257AF430E...	Permanent	Unlimited

Click the Add button to display the Add License dialog. Enter the license key that was provided by StarQuest and click the Add button of the Add License dialog to save the license key to the Licenses tab.

Figure 3. Adding a License Key



Host ID: C855CE03

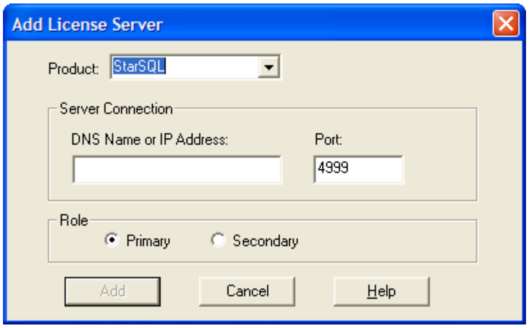
License:

Configuring a Client to Use a Floating License

Rather than using a node-locked license, StarSQL users and applications can check out a license from a StarLicense Server. StarSQL is designed to look first on the local computer for a license and if none is found, attempt to connect to a StarLicense server named **starlic** that is listening for license requests on communications port 4999. Use the StarLicense Manager utility to configure one or more StarLicense servers as appropriate for your environment.

To have the StarSQL client computer check out a license from a StarLicense server, click the License Servers tab of the License Configuration dialog, and then click the Add button.

Figure 4. Adding a License Server



Add License Server

Product: StarSQL

Server Connection

DNS Name or IP Address:

Port: 4999

Role

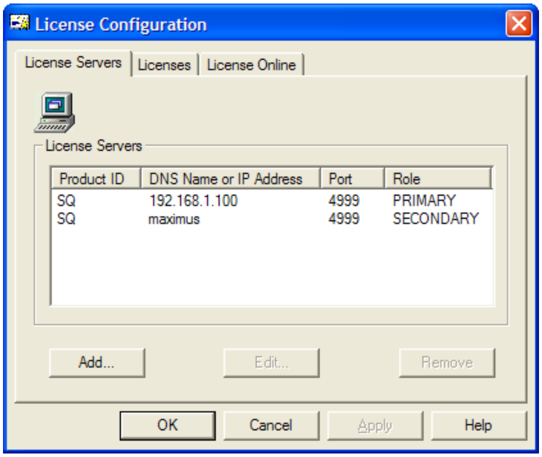
● Primary ○ Secondary

Select StarSQL or StarSQL for iSeries as the product for which you need a license, and supply the connection information appropriate for your StarLicense server(s). Click the Add button in the Add License Server dialog to save the information to the License Servers tab.

Using Multiple StarLicense Servers

If there is more than one StarLicense server available in a network, you can choose which server the computer uses as the primary server for obtaining a license. If the primary StarLicense server cannot provide a connection, a Secondary StarLicense server can issue a license. Select a server and click Edit to change which server acts as the Primary and Secondary server for issuing licenses to this computer.

Figure 5. Selecting a StarLicense Server to Use



Product ID	DNS Name or IP Address	Port	Role
SQ	192.168.1.100	4999	PRIMARY
SQ	maximus	4999	SECONDARY

You can define the license configuration and then import it to other computers, as described in "Sharing the StarLicense Server Configuration" on page 29.

Using a StarPipes Gateway for Licensing

If a StarPipes Gateway is configured with StarSQL or StarSQL for iSeries licenses, those licenses can be assigned to StarSQL client computers connecting to a DB2 host through the StarPipes Gateway. No license configuration is necessary on the client computer.

Deploying StarSQL in an Enterprise

StarSQL includes two programs that can help you manage a large number of client computers that need to run the StarSQL driver. The `DSIMPORT.EXE` program imports ODBC Data Source Name (DSN) definitions, and the `SQIMPORT.EXE` program allows you to import StarLicense information.

You can use these programs in conjunction with the `SETUP.EXE` program, and they also can be run from the client computer after StarSQL is installed.

Note

Windows x64 Users: Note that there are 32-bit and 64-bit versions of the `SETUP.EXE`, `DSIMPORT.EXE`, and `SQIMPORT.EXE` programs. If you install only the 32-bit version of StarSQL, the files are installed by default to C:\Program Files\StarQuest\StarSQL. If you install both the 32-bit and 64-bit versions of StarSQL, the 64-bit programs are installed by default to C:\Program Files\StarQuest\StarSQL and the 32-bit programs are installed by default to C:\Program Files (x86)\StarQuest\StarSQL. Be sure to use the version of the program file that is appropriate for the targeted environment.

Sharing Data Source Definitions

The `DSIMPORT.EXE` program looks for a file named `DSIMPORT.TXT` that contains the DSN definitions to import. You can import the DSN information as you install StarSQL on client computers, or you can run the `DSIMPORT.EXE` program after StarSQL is installed to update the DSN definitions on the client computer.

The format of the `DSIMPORT.TXT` file is the same for DSN definitions that are exported from a 32-bit or 64-bit environment and can be used with either the 32-bit or 64-bit version of `DSIMPORT.EXE`.

Note

It is important to specify the correct driver name in File data sources and in DSN-less connection strings. The 32-bit version of the StarSQL v5.5 driver is named StarSQL 32 and the 64-bit ODBC driver appears with the name StarSQL (64-bit).

Saving DSN Definitions

You must first create and save the data source definitions that you want to share with other computers.

1. From a Windows computer that has StarSQL installed, create and test one or more ODBC system data sources (see "Using the Data Source Configuration Wizard" on page 49). Be sure to use either the 32-bit or 64-bit version of ODBC Administrator, depending on whether you want to create a DSN for use with StarSQL 32 or StarSQL (64-bit).

2. Use REGEDIT or another registry editor to access the following `ODBC.INI` object in the Windows registry:

   For **User** data sources (32- and 64-bit):

   ```
   HKEY_CURRENT_USER\Software\ODBC\ODBC.INI
   ```

   For **System** data sources:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI
   ```

   or, for 32-bit data sources on a 64-bit computer:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI
   ```

3. Select the desired ODBC.INI object or specific data source object from the registry, choose File–>Export.

4. Name the text file `DSIMPORT.TXT` and be sure the file type is “Save as type Win9x/NT4 Registration Files (REGEDIT4)”. Click Save to export the object.

If you save the entire ODBC.INI object to `DSIMPORT.TXT` and have non-StarSQL data sources present, only the StarSQL data sources will be imported by the `DSIMPORT.EXE` program.

Importing DSNs During Installation of StarSQL

To install StarSQL on a large number of client computers, create a shared directory on a network drive that can be accessed from each of the client computers and copy the StarSQL distribution files to the shared directory.

To replicate the DSN definitions as you install StarSQL for Windows on client computers:

1. Copy the `DSIMPORT.TXT` file that contains the DSN definitions to the shared network directory that contains the StarSQL installer image. The `DSIMPORT.TXT` file must be located in the same directory as the StarSQL `SETUP.EXE` program. (The format of the `DSIMPORT.TXT` file is the same and can be used with either the 32-bit or 64-bit version of the `SETUP.EXE` program.)

2. Connect to the shared network directory that contains the StarSQL installer image from each of the client computers and run the 32-bit or 64-bit version of the `SETUP.EXE` program as appropriate for the target environment.

The presence of the `DSIMPORT.TXT` file causes the Setup program to run the `DSIMPORT.EXE` program, which imports the data source definitions to the Windows registry of the client computer as StarSQL is installed. Note that data sources are imported from the `DSIMPORT.TXT` file only when performing a new installation of StarSQL on a computer, not during an update of the StarSQL software.

Updating DSNs After Installation

You also can run the DSIMPORT.EXE program to update the ODBC DSN definitions of a client computer that is running StarSQL.

1. Copy the DSIMPORT.TXT file that contains the DSN definitions you want to import to the StarSQL Programs directory of the target computer. The DSIMPORT.TXT file must be located in the same directory as the DSIMPORT.EXE program. Table 5 on page 19 shows the default installation path used for each version of the StarSQL driver.
2. Double-click the DSIMPORT.EXE program to import the StarSQL DSN definitions to the local computer.

If you regularly add or change the DSNs that you want StarSQL client computers to use you may want to maintain the customized DSIMPORT.TXT file on a shared network drive and modify network login scripts to automatically get a fresh copy of the DSIMPORT.TXT file and run the DSIMPORT.EXE program.

Sharing the StarLicense Server Configuration

If a client is not configured with a local license key, the StarSQL driver attempts to connect to a StarLicense server with a host name of **starlic** that is listening for requests on communication port 4999. The online help for the StarLicense Manager program contains information about configuring StarLicense servers and changing the default behavior.

To configure client computers to use particular StarLicense servers, you can define the license configuration and then import the license configuration file to multiple computers. The SQIMPORT.EXE program looks for a file named STARLIC.LIC that contains the license configuration to import. You can import the StarLicense server information as you install StarSQL on client computers, or you can run the SQIMPORT.EXE program after StarSQL is installed to update the license configuration on the client computer.

Saving the License Configuration

When you use the License Configuration utility to configure a license server connection, it creates a file named starlic.lic. If you want to proliferate that license configuration to other client computers, perform the following steps.

1. From a Windows computer that has StarSQL installed, select Programs–>StarSQL–>License Configuration to start the License Configuration utility.

2. In the License Servers tab of the License Configuration utility, define the license servers you want to use (see "Configuring a Client to Use a Floating License" on page 25).

3. Click the OK button to save the license configuration. The License Configuration utility saves the file as `starlic.lic` in the \Programs directory where StarSQL is installed. Table 5 on page 19 shows the default installation path used for each version of the StarSQL driver.

Importing a License Configuration During Installation

After you save the license configuration you can import it as you install StarSQL on client computers. If you have not already done so, create a shared directory on a network drive that can be accessed from each of the client computers and copy the StarSQL distribution files to the shared directory.

1. Copy the `starlic.lic` file that contains the license configuration to the shared network directory that contains the StarSQL installer image. The `starlic.lic` file must be located in the same directory as the StarSQL SETUP.EXE program for the installer to import the license configuration during installation. If you are deploying both the 32-bit and 64-bit versions of StarSQL, copy the `starlic.lic` file to both the x86 and x64 directories.

2. Connect to the shared network directory that contains the StarSQL installer image from each of the client computers and run the `SETUP.EXE` program from the appropriate directory (x86 for the 32-bit version of StarSQL, and x64 for the 64-bit version).

The presence of the `starlic.lic` file causes the Setup program to run the `SQIMPORT.EXE` program, which imports the license configuration to the client computer as StarSQL is installed.

Updating the License Configuration After Installation

You also can run the `SQIMPORT.EXE` program to update the StarLicense server configuration of a client computer that is running StarSQL.

1. Copy the `starlic.lic` file that contains the server configuration that you want to import to the StarSQL Programs directory of the target computer. The `starlic.lic` file must be located in the same directory as the `SQIMPORT.EXE` program. Table 5 on page 19 shows the default installation path used for each version of the StarSQL driver.

2. Double-click the `SQIMPORT.EXE` program to import the StarLicense configuration to the local computer.

If you regularly add or change the StarLicense servers that you want StarSQL client computers to use you may want to maintain the customized `starlic.lic` file on a shared network drive and modify network login scripts to automatically get a fresh copy of the file and run the `SQIMPORT.EXE` program.

CHAPTER 3

Preparing Hosts for StarSQL Access

This chapter describes how to prepare host systems to enable StarSQL to provide access to the host databases.

It covers:

- Preparation required for all hosts
- Preparing a DB2 for z/OS host
- Preparing a DB2 for i host
- Preparing a DB2 for Linux, UNIX, and Windows (LUW) host

These sections cover details of the DB2 environment that are pertinent to StarSQL. For complete documentation of installation and configuration on the host, consult IBM’s DB2 documentation, especially IBM’s *DRDA Connectivity Guide*.

Contact StarQuest Customer Support (see page 12) for assistance if you plan to use StarSQL to connect to DB2 UDB using the SNA network protocol.

Preparation Required for All Hosts

Regardless of the host platform, you will need the information described in Table 6 to configure an ODBC data source that will use StarSQL to access the DB2 host. You may need to obtain this information from the DB2 administrator. The section "Configuring Data Sources" on page 49 contains details about configuring data sources.

Table 6. Host Information Required for Data Source Configuration

Data Source Information Item	Information Needed
Package Collection Name	The location of the SQL packages that StarSQL requires. The Package Collection Name on a DB2 for i host is the name of the library that contains the StarSQL packages. For DB2 for z/OS, it is the name of the virtual collection associated with these packages.
Database Server Name	The relational database name. On different DB2 hosts this may be referred to as Location Name, Global Resource Name, RDB name, Database Name, or dbname.
User ID and Password	A valid user id and password for logging into the database. The user account information is not stored in the DSN.
TCP/IP Connection Information	The host name and the port used for DRDA communications.

In addition to properly preparing the host, each StarSQL user must have an account on the host and have permission to access the necessary packages.

User Accounts

To connect to a DB2 database, each StarSQL user needs an account on the host database. An account consists of a user ID and password.

You need to provide the user account information to each StarSQL user who needs to connect to the database.

Permissions

Usually a database administrator (DBA) is responsible for packages on the host, including binding them and granting permissions to use them. Depending on the host platform and the type of package used by the ODBC application, the DBA may need to grant StarSQL users explicit permissions to access data used by the application.

Preparing DB2 on a z/OS Host

Preparing DB2 on a z/oS host for access with StarSQL primarily involves configuring the Distributed Data Facility (DDF), which is a component of DB2 for z/OS. Its primary task is to process DRDA requests. DDF must be active for a desktop to connect to DB2 using StarSQL or any other DRDA requestor or client.

Configuring DDF

If your organization has not implemented distributed database capabilities, DDF may not be configured and activated. The DSNTINST CLIST provides two panels—DSNTIPR and DSNTIP5 for customizing a DB2 for z/OS subsystem to use native DRDA TCP/IP support. The DSNTIP5 panel is specific for TCP/IP. However, to use native TCP/IP support, you also must have APPC support configured and active because DB2 uses the network ID and the LU name to identify units of work. You specify the LU name that identifies the DB2 subsystem to VTAM and to uniquely identify logical units of work, in the DSNTIPR panel.

The values specified on these panels are used to generate the JCL that stores them in the DB2 bootstrap data set (BSDS) communication record.

If you are installing DB2, use the DDF panel DSNTIPR and DSNTIP5 to provide the following parameters. To change the DDF parameters after installation, run a customized configuration job DSNTIJUZ to update the BSDS.

- DDF Location Name. This name must be specified for the Database Server Name of the ODBC data source that StarSQL uses to connect to the host.
- Password used when connecting DB2 to VTAM, if a password is required.
- IP port to use for TCP/IP access. To enable support for TCP/IP, set the DRDA port in the DDF to 446.
- IP port to use for two-phase commit. The RESYNC PORT parameter in the DSNTIP5 panel specifies a TCP/IP port number for processing requests for two-phase commit re synchronization. The RESYNC PORT must be different than the DRDA PORT, and it must match the port

number specified for two-phase-commit recovery operations in the StarSQL Resource Manager. The StarSQL Resource Manager uses a default port value of 5020.

For more information about establishing connectivity between your desktop and DB2 for z/OS with TCP/IP, consult the Installation Guide for your version of DB2 for z/OS.

For more information about configuring DDF, consult IBM's DB2 for z/OS installation documentation and the IBM Redbook, *Distributed Functions of DB2 for z/OS and OS/390* . For more information about establishing connectivity between client computers and DB2 for z/OS over a TCP/IP network, the IBM Redbook, *WOW! DRDA Supports TCP/IP: DB2 Server for OS/390 and DB2 Universal Database*, may be particularly useful. Refer to "Documentation" on page 11 for details about these publications.

Starting DDF

Use the following command, which requires authority of SYSOPR or higher, to start DDF:

```
-START DDF
```

When DDF starts successfully, the following messages are displayed:

```
DSNL003I - DDF IS STARTING
DSNL004I - DDF START COMPLETE LOCATION locname LU
netname.luname
```

If DDF has not been properly installed, the START DDF command fails and displays the following message:

```
DSN9032I - REQUESTED FUNCTION IS NOT AVAILABLE
```

If DDF has already been started, the START DDF command fails and displays the following message:

```
DSNL001I - DDF IS ALREADY STARTED
```

The following command shows whether DDF is running and, if so, the parameters that it is using:

```
-DIS DDF
```

Supporting Password Management Using DRDA Flows

Password Management using DRDA flows is supported for a network using the TCP/IP protocol. To support the ability of StarSQL users to change their host passwords through StarSQL on DB2 for z/OS, set Extended Security to YES (EXTSEC=YES). This can be done using either

- the DSNTIPR (DDF) panel on the DB2 installation dialog.
- a customized configuration job DSNTIJUZ, with the option EXTSEC=YES specified.

Using StarSQL with Stored Procedures

Stored procedures are application programs that reside on the host and are invoked via DB2. They are usually written in a traditional programming language like COBOL, RPG, or C. They may contain SQL statements for accessing the DB2 database or they may be used to access non-DB2 resources.

You invoke a stored procedure using the SQL Call statement and receive output data in a result set or in output parameters. The Call statement is executed as any other SQL, using SQLExecute or SQLExecDirect.

Registering Stored Procedures

The stored procedure should be registered on the host so that calling application can obtain information using the SQLProcedures and SQLProcedureColumns functions.

Use the CREATE PROCEDURE command to register the stored procedure in the system. The CREATE PROCEDURE command automatically updates the SYSIBM.SYSROUTINES catalog table.

```
CREATE PROCEDURE SYSPROC.STARPING (
IN REGION CHAR(8) CCSID EBCDIC,
IN PROGRAM CHAR(8) CCSID EBCDIC,
IN TRANSID CHAR(4) CCSID EBCDIC,
IN COMMLEN SMALLINT,
INOUT COMMAREA VARCHAR(32700) FOR BIT DATA,
OUT RC INTEGER,
OUT ABCODE CHAR(4) CCSID EBCDIC
)
PARAMETER STYLE GENERAL
LANGUAGE C
EXTERNAL NAME 'STARPING'
RESULT SETS 0
DETERMINISTIC
NO SQL
NO DBINFO
NO COLLID
ASUTIME NO LIMIT
NO WLM ENVIRONMENT
STAY RESIDENT NO
PROGRAM TYPE MAIN
SECURITY DB2
```

```
    COMMIT ON RETURN YES
```

Calling Stored Procedures

Stored procedures on DB2 for z/OS may return a result set.

Each of the following SQL statements for calling the sample stored procedure is valid:

```
Call MyProc (1, 'A', ?, ?)
Call MyProc( parm1=1, parm2='A', parm3=?, parm4=?)
Call MyProc( parm1=1, parm2='A', ?, ?)
Call MyProc( ?, ?, ?, ?)
Call MyProc( 1, 'A', parm3=?, parm4=?)
```

Configuring SSL for DB2 for z/OS

To configure a DB2 for z/OS host system for SSL communications, you must be using DB2 for z/OS 9.1 or later. Refer to the section *Encrypting your data with Secure Socket Layer support* in the DB2 for z/OS documentation.

You must configure AT-TLS (z/OS® Communications Server IP Application Transparent Transport Layer Security) as well as configuring the DB2 server.

The listening port number (typically 448) is specified in the DRDA SECURE PORT field of the Distributed Data Facility Panel 2 (DSNTIP5) during DB2 installation. After initial installation, you can update the SECPORT parameter of the DDF statement in the BSDS with the change log inventory (DSNJU003) stand-alone utility.

The following IBM Redbooks may also be of assistance:

- *Securing and Auditing Data on DB2 for z/OS* (SG24-7720)
- *DB2 9 for z/OS: Distributed Functions* (SG24-6952-01)

Preparing a DB2 for i Host

This section discusses setting up a DB2 for i host for supporting a connection through StarSQL for Windows.

It covers:

- creating a library/collection for SQL packages
- determining the RDB name
- enabling DRDA over TCP/IP for OS/400
- configuring SSL
- Registering stored procedures

For complete information about setting up DB2 for i, consult IBM’s documentation.

Creating a Library for SQL Packages

On DB2 for i, required SQL packages are stored in a collection, or library. You may need to create the collection or library on the host.

Use the CRTLIB command to create a new library for the SQL packages used by StarSQL. You can do this from a 5250 terminal session with a user ID that has QSECOFR privileges. The library does not have to be a SQL collection, but it must be accessible to all StarSQL users.

For example, the following command creates a new library named STARSQL:

```
CRTLIB STARSQL
```

Record the name of the library as it must be specified as the SQL Package Collection ID in the data source configuration.

Determining the RDB Name

Determine the Relational Database (RDB) name of the DB2 for i host. From the AS/400 command line, enter:

```
WRKRDBDIRE
```

Look for an entry with a Remote Location value of *LOCAL. If such an entry does not exist, create it with the 1=ADD option. A common convention is to use the same name as the AS/400 system name for the RDB name.

Make a note of the RDB name as you need to specify it as the Database Server Name in the data source configuration.

Enabling DRDA Over TCP/IP

The Distributed Data Management (DDM) server allows client computers to access the DB2 functions. The DDM server supports remote SQL access, record level access, and remote journals. To initiate a DDM server job using TCP/IP communications a DRDA application or DDM source system connects to the well-known port number for TCP/IP, port 446 or 447. The DDM listener program, upon accepting the connection request, issues an internal request to attach the client's connection to a DDM server job.

The DDM listener program runs in a batch job in the QSYSWRK subsystem. There is one listener program that serves potentially many DDM server jobs. If you have access to iSeries Navigator you can verify whether DDM is configured by selecting TCP/IP from the Network–>Servers menu.

Follow the steps below if you need to configure an AS/400 host to accept DRDA requests over TCP/IP:

1. Log on to the AS/400 host.
2. Change the DDM TCP/IP Attributes to automatically start the listener program by entering:

   ```
   CHGDDMTCPA AUTOSTART(*YES)
   ```

3. Start the TCPIP DDM Server by entering:

   ```
   STRTCPSVR SERVER(*DDM)
   ```

When you are logged on to the AS/400 host, you can examine which port DB2 for i is using to listen for DRDA requests using either the WRKSRVTBLE or the WRKTCPSTS command.

To use the WRKSRVTBLE command:

1. Enter WRKSRVTBLE.
2. Look for the DRDA entry with the port number.

To use the WRKTCPSTS command:

1. Enter WRKTCPSTS.
2. Choose option 3, "Work with TCP/IP connection status."
3. Find the entry with port "drda" and press "F14=Display port numbers." The default port number for DRDA is 446.

Configuring SSL on DB2 for i

To configure a System i host system to use the Secure Sockets Layer (SSL) protocol you must have the following components:

- Digital Certificate Manager - 5722-SS1 (v5rx), 5761-SS1 (v6r1), or 5770-SS1(v7r1) option 34
- TCP/IP Connectivity Utilities - 5722-TC1(v5r4), 5761-TC1 (v6r1), or 5770-SS1 (v7r1)

- IBM HTTP Server - 5722-DG1 (v5rx), 5761-DG1 (v6r1) or 5770-DG1 (v7r1)

Following are general procedures for configuring SSL on the iSeries host. Refer to your IBM documentation for details, especially the AS/400 documentation and the IBM Redbook *IBM iSeries Wired Network Security OS/400 V5R1 DCM and Cryptography Enhancements* (GSG24-6168).

1. Start the Admin HTTP instance and use a browser to configure the Digital Certificate Manager.
2. Create a local Certificate Authority or obtain a certificate from a public Internet Certificate Authority.
3. Create a *SYSTEM certificate store.
4. Use “Manage Applications” to assign a server certificate to the OS/400 DDM/DRDA server.
5. After you assign the certificate, restart the DDM/DRDA server:

   ENDTCPSVR *DDM

   STRTCPSVR *DDM
6. If necessary, set the port on which the DDM/DRDA server listens for SSL conversations. Use WRKSRVTBLE to view and modify service table entries; the entry for SSL is ddm-ssl, and the default value is 448.

Registering Stored Procedures on DB2 for i

This section describes issues regarding stored procedures that are specific to AS/400 hosts. Refer to "Using StarSQL with Stored Procedures" on page 37 for general information about using stored procedures.

Following is sample SQL for registering a stored procedure on an AS/400 system. It assumes that the COBOL program MYLIB.MYPRGM already exists on the AS/400. This statement modifies the QSYS2.SYSPROCS and QSYS2.SYSPARMS catalog tables for you.

```
CREATE PROCEDURE MYLIB.MYPROC (INOUT PARM1 CHAR(10))
EXTERNAL NAME MYLIB.MYPGM LANGUAGE COBOL GENERAL
```

In the above example, the procedure name is MYLIB.MYPROC, which references the COBOL program MYLIB.MYPGM. The program takes one input parameter called PARM1 which is a char field of length 10. This procedure does not return a result set.

Refer to the *IBM SQL Reference and SQL Programming Guide* for the appropriate version of IBM i (i5/OS) for more information on registering a stored procedure and the full syntax of the CREATE PROCEDURE statement.

Considerations for Specific IBM i Releases

In general, it is a good idea to stay current on PTF packages and the DB2 Group PTF, and to use the latest version of StarSQL. Refer to the StarSQL Release Notes (readme.html) for specific issues for particular versions of IBM i.

Preparing a DB2 LUW Host

This section provides details for setting up a DB2 for Linux, UNIX, and Windows host to support a connection through StarSQL for Windows. It covers:

- enabling DRDA support for TCP/IP
- configuring SSL support
- enabling encryption
- locating the database name

For complete information about setting up DB2 LUW, consult IBM's DB2 LUW installation documentation.

Enabling DRDA Support for TCP/IP

When using TCP/IP to connect to a DB2 LUW host, make sure that the host has a static IP address. You may experience problems if DB2 LUW is installed on a computer using DHCP. To be recognized, DB2 LUW requires an entry in the DNS (Domain Name Server) or an entry in the HOSTS file.

Although you can configure StarSQL to communicate with DB2 using any available port, port 446 is the standard port used for DRDA communications and is the default value that StarSQL uses if another is not specified. DB2 LUW uses a default value of 50000 for DRDA communications; this allows DB2 processes on UNIX to run without requiring that the instance owner user ID have root authority.

If there is a firewall that monitors network traffic to the DB2 host, be sure that it allows DRDA communications to pass through the port that you configure for DRDA requests.

Using db2 Commands to Specify the DRDA Port

Issue the following db2 commands to change the port that DB2 uses for DRDA communications.

1. Enter the following command to determine on which port DB2 is listening to for TCP/IP communications.

   ```
   db2 get dbm configuration
   ```

   In the dbm configuration, look for “TPC/IP Service Name (SVCENAME).” It will have a value similar to “db2c_DB2.” This is the symbolic name of the connection port.

2. Find the symbolic name of the connection in the services file. The services file is located in the path \WINDOWS\system32\drivers\etc\services on a Windows computer.

3. Edit the services field and change the value of the connection port to 446. Change the value of the interrupt port to 447.

4. Restart the DB2 instance for the changes to take effect.

To verify that DB2 is listening on the correct port, from either the client or the server enter:

```
telnet <host> 446
```

If DB2 is listening on that port, no error is returned to the telnet window. If DB2 is not listening on that port, you will see an error similar to the following and may need to contact your DB2 administrator for the correct port number to use:

```
Could not open connection to the host, on port 446: Connect
failed.
```

Click the Close (X) icon to close the telnet window.

Configuring SSL for DB2 UDB 9.7 & later:

Refer to the *Configuring Secure Sockets Layer (SSL) support in a DB2 instance* chapter in the DB2 documentation for details.

The following is an example of using self-signed certificates.

Use GSKit to create a keystore file (certificate database) and a certificate. Export the certificate if desired:

```
C> cd C:\Program Files\ibm\gsk8\bin\gsk8capicmd
```

```
C> gsk8capicmd -keydb -create -db "mydbserver.kdb" -pw
"mypassword" -stash
```

```
C> gsk8capicmd -cert -create -db "mydbserver.kdb" -pw "
mypassword " -label "SelfSigned" -dn
"CN=myhost.mydomain.com,O=MyCompany,OU=CustomerSupport,L=C
alifornia,ST=ON,C=CA"
```

```
C> gsk8capicmd -cert -extract -db "mydbserver.kdb" -pw
"mypassword" -label "SelfSigned" -target
"MYHOSTserver.arm" -format ascii -fips
```

On UNIX, make sure that the DB2 instance owner has read access to the keystore file.

Update DB2 and restart it:

```
C> db2 update dbm cfg using SSL_SVR_KEYDB "C:\Program
Files\ibm\gsk8\bin\mydbserver.kdb"
```

```
C> db2 update dbm cfg using SSL_SVR_LABEL SelfSigned
```

```
C> db2 update dbm cfg using SSL_SVR_STASH "C:\Program
Files\ibm\gsk8\bin\mydbserver.sth"
```

```
C> db2 update dbm cfg using SSL_SVCENAME 50029
```

```
C> db2set -i db2 DB2COMM=SSL,TCPIP
```

```
C> db2stop
```

```
C> db2start
```

Note that there is a problem in DB2 UDB 9.7 fp3 (resolved in fp4); see *IC72728: THE PORT NUMBER FOR SSL_SVCENAME IN THE SERVICE FILE IS USED FOR SVCENAME*. The value specified for SSL_SVCENAME is being used for both SSL and non-SSL listeners, causing a conflict. The workaround is to set the SVCENAME parameter using an explicit port number:

```
C> db2 update dbm cfg using SVCENAME 50000
```

Configuring SSL for DB2 UDB 9.1 & 9.5:

The procedure is similar to the above instructions for DB2 UDB 9.7, except:

- The gskit is v7 rather than v8 (e.g. run C:\Program Files\ibm\gsk7\bin\gsk7capicmd instead of C:\Program Files\ibm\gsk8\bin\gsk8capicmd)
- Rather than using DBM configuration, the DB2 SSL parameters are stored in a configuration file sslconfig.ini located in the following directory:

  •Linux and UNIX: INSTHOME/cfg/SSLconfig.ini

  •Windows: INSTHOME/SSLconfig.ini

  where INSTHOME is the home directory of the instance.

Here is a sample sslconfig.ini:

```
DB2_SSL_KEYSTORE_FILE=C:\Program
Files\ibm\gsk7\bin\mydbserver.kdb

DB2_SSL_LISTENER=50448

DB2_SSL_KEYSTORE_PW=mypassword

DB2_SSL_KEYSTORE_LABEL=SelfSigned
```

After configuring gsk7 and creating sslconfig.ini, enter **`db2set -i db2 DB2COMM=SSL,TCPIP`** and restart DB2

Enabling Encryption

If you configure the StarSQL driver to send encrypted user IDs and passwords (see the "UseEncryption" expert setting in the StarSQL Help), be sure to enable the database for encryption. On a DB2 for Linux, UNIX, and Windows host you enable encryption by setting the Server Connection Authentication (SRVCON_AUTH) parameter to “Server encrypt” and restarting the instance.

Using db2 Commands to Enable Encryption

Issue the following db2 commands to enable encryption.

1. To enable encryption from a db2 command window, enter the following command:

   ```
   db2 update dbm cfg using SRVCON_AUTH
   SERVER_ENCRYPT
   ```

2. Stop and restart the instance by issuing the following commands:

```
db2stop
db2start
```

Locating the Database Name

When an administrator sets up a UDB system, they assign names to DB2 databases. You will need to enter the Database Server Name when you configure data sources on the desktop. You can display a list of the databases that have been created by issuing the following command:

```
db2 list database directory
```

Contact your database administrator if you need to determine which databases should be accessible to StarSQL clients.

Preparing a Derby Network Server Host

The Derby Network Server is part of the Derby software distribution and provides a framework for multi-user connectivity to Derby databases across a network. Derby must be started in the Network Server mode (vs. the embedded mode) for client computers to connect to a Derby database using StarSQL for Java.

The default of the Derby Network Server is to start with user authentication disabled. Refer to the Derby documentation for information about enabling user authentication and running under the Java security manager to help avoid security problems.

Setting Network Server Properties

Typically the Java system property `derby.system.home` specifies the system directory, which contains subdirectories that hold the databases that are available to the Derby Network Server. If you do not explicitly set the `derby.system.home` property when starting Derby, the default is the directory in which Derby was started. You need to specify at least the name of the database you want to access with the StarSQL for Java driver and which port to use. Contact your database administrator if you do not know which Derby databases should be accessible to StarSQL for Java clients.

The Network Server properties can be specified three ways:

- on the command line
- in the `.bat` or `.ksh` files, loading the properties by executing **java -D**
- in the `derby.properties` file

Properties in the command line or in the .bat or .ksh files take precedence over the properties in the derby.properties file. Arguments included on commands that are issued on the command line take precedence over property values.

The properties that are particularly of interest for using the StarSQL for Java driver are those that allow remote connections and determine which protocol to use.

```
derby.drda.host=hostname
derby.drda.portNumber=portnumber
derby.drda.sslMode=SSL | TLS || SSLv3 | TLSv1
```

Enabling Remote Connections

The default of the Derby Network Server is to start with user authentication disabled. Before you enable remote connections ensure that you are running under the security manager and that user authorization is enabled. Refer to the Derby documentation for details about enabling user authentication and running under the Java security manager.

The `derby.drda.host` property causes the Network Server to listen on a specific interface, allowing multiple instances of Network Server to run on a single computer with a unique host:port combination. By default the Derby Network Server listens only on the loopback IP address (127.0.0.1) and port 1527, which restricts access to the local computer. To make the Derby Network Server accessible to other computers on the network, specify a particular interface (host name or IP address) and specify a port number other than 1527 on which to listen for connections. The Derby Network Server must listen for connections on the same port that the StarSQL for Java driver is configured to use, which is port 446 by default.

The following shows how to use a java command to start the Derby Network Server to listen on port 446 for connection requests from any host name or IP address.

```
java -jar $DERBY_HOME%\lib\derbyrun.jar server start
          -h 0.0.0.0 -p 446
```

The -h and -p options can be specified regardless of how you choose to start Derby Network Server. Refer to the Derby documentation for details about the additional methods, such as running the startNetworkServer script or using a java command to directly invoke the NetworkServerControl class.

Configuring Support for the SSL Protocol

The Derby Network Server supports network security with Secure Socket Layer/Transport Layer Security (SSL/TLS). With SSL/TLS the client-server communication protocol is encrypted and both the client and the server may, independently of each other, require certificate-based authentication of the other part. You can configure SSL for the Derby Network Server to be Off, or to use SSL encryption with no peer authentication (basic), or to use SSL encryption and peer authentication (peerAuthentication). To enable SSL support, use the `ssl` option on the command line when you start Derby or specify the `derby.drda.sslMode` property in `derby.properties`.

For example, the following command would start Derby Network Server using the "basic" SSL encryption option:

```
java -jar derbyrun.jar server start -ssl basic
```

You also need to use the **keytool** to create a server key pair, and specify the key store and password when starting the server. For SSL operation the server always needs a key pair. If the server runs in peer authentication mode, then each client needs its own key pair. The **keytool** is located in the JRE bin directory. Entering the following command prompts for the information needed to generate a key pair.

```
keytool -genkey myDatabase -keystore serverKeyStore.key
```

The key pair is located in a file which is called a key store and the JDK's SSL provider needs the system properties `javax.net.ssl.keyStore` and `javax.net.ssl.keyStorePassword` to access the key store. Specify the key store and password when starting the server, such as shown below:

```
java -Djavax.net.ssl.keyStore=serverKeyStore.key \
     -Djavax.net.ssl.keyStorePassword=myPassword \
     -jar %DERBY_HOME%\lib\derbyrun.jar server start
                      -h 0.0.0.0 -p 446 -ssl basic
```

Refer to the Derby documentation at http://db.apache.org/derby/docs/10.4/adminguide/cadminsslkeys.html for additional details about key and certificate handling.

CHAPTER 4

Configuring Data Sources

To establish a connection between the desktop and a DB2 database, you configure a data source using the ODBC Data Source Administrator. To access more than one database, configure a data source for each database. You need to obtain information about the host, as listed in "Preparation Required for All Hosts" on page 33, from the DB2 administrator, and some information about the network from the network administrator.

This chapter covers:

- Using the StarSQL Data Source Configuration Wizard
- Supporting Two-Phase Commit

Using the Data Source Configuration Wizard

To configure a data source after StarSQL is installed, start the ODBC Data Source Administrator. The ODBC Administrator is typically available from the Administrative Tools item of the Windows Control Panel.

If you are running Windows x64, there is a 32-bit ODBC Administrator (located in Windows\SysWOW64) and a 64-bit ODBC Administrator (located in Windows\System32). If you have both versions of StarSQL installed, the ODBC Administrator shortcut in the StarSQL (32-bit) program group starts the 32-bit ODBC Administrator and the shortcut in the StarSQL (64-bit) program group starts the 64-bit version of the ODBC Administrator.

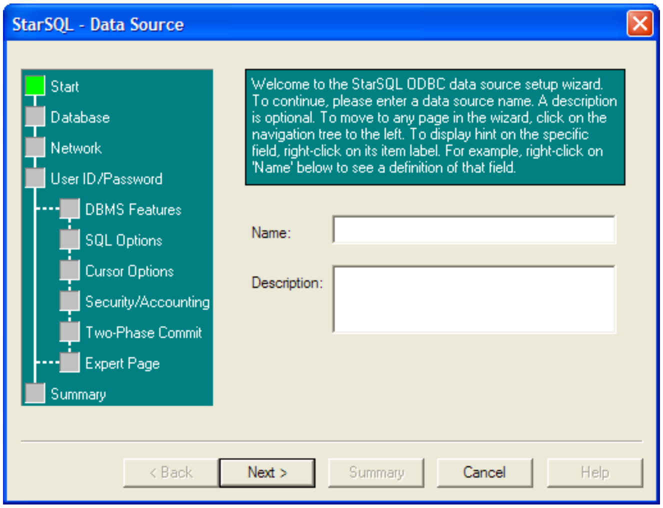
1. From the ODBC Data Source Administrator, click on the User DSN, System DSN, or File DSN tab to configure the appropriate type of data source.

   A *User DSN* is local to a machine and can be used only by the current user. A *System DSN* is local to a machine rather than dedicated to a user. The system or any user having privileges can use a data source set up as a System DSN. A

File DSN is data source information that is stored in a file instead of in the registry, which allows it to be shared by multiple users or multiple systems if the file is stored on a network file server.

2. To create a new data source, click the Add button, select StarSQL 32 or StarSQL (64-bit) as the driver, depending on which version of StarSQL you want to use the DSN with, and click Finish to display the StarSQL Data Source Configuration Wizard.
   To modify an existing data source, select a Data Source Name and click Configure, which displays the StarSQL Data Source Configuration Wizard.

Figure 7. StarSQL Data Source Configuration Wizard



Welcome to the StarSQL ODBC data source setup wizard. To continue, please enter a data source name. A description is optional. To move to any page in the wizard, click on the navigation tree to the left. To display hint on the specific field, right-click on its item label. For example, right-click on 'Name' below to see a definition of that field.

The Data Source Configuration Wizard presents dialogs to obtain values for at least the following required data source settings:

- Data Source Name
- Database Server Name
- Package Collection Name
- Network Information

Note

Do not specify a Data Source Name that includes parentheses [( )]. The ODBC driver manager does not allow parentheses in data source names. If you configure a data source with parentheses in the name, the DSN will not be usable and can be removed only by manually editing the Windows registry.

Click the Help button to display information about using and responding to the dialogs. (If the Help button is disabled, click on Start to display the first panel of the Data Source Configuration Wizard.) You also can right-click on any field name for a description of that field.

The first four panels of the ODBC Data Source Configuration Wizard lead you through configuration of the required settings. When you provide the user id and password, you can choose to connect to the host database at that time. (The user id and password are used only for connecting to the host—they are not stored in the data source.)

If the connection succeeds, the Wizard will automatically obtain information from the host database platform and set the remainder of the configuration settings. You can view the default settings by clicking the Summary button in the bottom of the Wizard window.

If the connection fails, or if you want to explicitly set the advanced settings rather than use default values, click on the Advanced button. The Advanced configuration settings of the Data Source Wizard allow you to specify more detailed information about the ODBC connections, but are intended to be modified only by SQL experts.

Sharing Data Source Definitions Among StarSQL Users

After you create one or more data sources, you can use the `DSIMPORT.EXE` program that is included with StarSQL to replicate data source definitions to computers on which StarSQL is installed.

Note

Windows x64 Users: Be sure to use the appropriate version of `DSIMPORT.EXE` to import data sources. Use the 32-bit version of `DSIMPORT.EXE`, which is located in the C:\Program Files (x86)\StarQuest\StarSQL, to import 32-bit data sources, and the 64-bit version of `DSIMPORT.EXE`, located in the C:\Program Files\StarQuest\StarSQL, to import 32-bit or 64-bit data sources. The format of the `DSIMPORT.TXT` file is the same for DSN definitions exported from either a 32-bit or 64-bit environment and can be used with either the 32-bit or 64-bit version of `DSIMPORT.EXE`.

1. After you create one or more data source definitions, use REGEDIT or another registry editor to access the following `ODBC.INI` object in the Windows registry. On a 64-bit computer the 64-bit data sources are stored in the standard ODBC object and 32-bit data sources are stored under a mirror ODBC object in the Wow6432Node, as detailed below.

   For **User** data sources (32- and 64-bit):

   ```
   HKEY_CURRENT_USER\Software\ODBC\ODBC.INI
   ```

   For **System** data sources:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI
   ```

   or, for 32-bit data sources on a 64-bit computer:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI
   ```

2. Select the desired ODBC.INI object or specific data source object from the registry, choose File–>Export.

3. Name the text file `DSIMPORT.TXT` and be sure the file type is “Save as type Win9x/NT4 Registration Files (REGEDIT4)”. Click Save to export the object.

4. Move the customized `DSIMPORT.TXT`, and the appropriate version (32-bit or 64-bit) of the `DSIMPORT.EXE` file to a shared network directory that all StarSQL users can access.

5. Double-click the `DSIMPORT.EXE` file to run the utility and import the StarSQL DSN definitions from the `DSIMPORT.TXT` file to the registry. The data source entries will appear in either the standard ODBC object or the mirrored Wow6432Node ODBC object, depending on whether you execute the 32-bit or 64-bit version of `DSIMPORT.EXE`.

Note that data sources are imported from the DSIMPORT.TXT file only when performing a fresh installation of StarSQL on a computer, not during an update of the StarSQL software.

Supporting Two-Phase Commit

In a distributed processing environment where a unit of work spans more than one database, using the two-phase commit protocol can help ensure the integrity of the transaction and databases. When the two-phase commit protocol is in effect, the transaction must complete successfully before it is committed. If the transaction fails, all of the updates are rolled back and the databases remain in the state they were before the transaction was begun.

The following describes using two-phase commit with the Microsoft Transaction Server (MTS) and Distributed Transaction Coordinator (DTC), and the StarSQL Resource Manager. This support has been available in StarSQLfor Windows since v3. StarSQL v6.4 adds support for XA Distributed Transactions, which is available for StarSQL on both Windows and UNIX, and is documented as a technical document on the StarQuest Info Center website

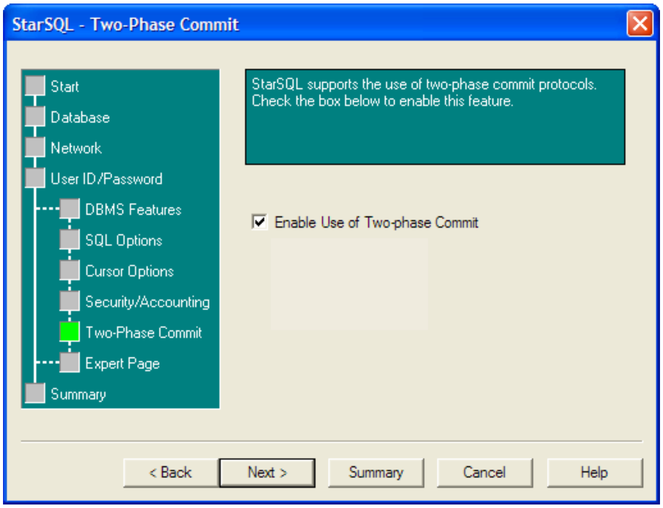
Using Two-Phase Commit with Microsoft DTC

To enable applications to use the two-phase commit protocol with the Microsoft DTC, you must perform a Custom installation of StarSQL (see "Installing StarSQL" on page 18) to install the StarSQL Resource Manager. In a 64-bit environment, you should install the 64-bit version of StarSQL; the two-phase commit feature is not available when installing the 32-bit version of StarSQL in a 64-bit environment.

After installing StarSQL for Windows, use the ODBC Data Source Administrator to enable the client computer to use the two-phase commit protocol, as described in the following steps.

1. From the ODBC Data Source Administrator (see "Using the Data Source Configuration Wizard" on page 49), select the System Data Source and click Configure.
2. Click on the Two-Phase Commit option under the User ID/Password category. Note that the appearance of this dialog has changed and now includes radio buttons to choose between TCP/IP DRDA (Microsoft DTC) and XA Distributed Transactions.

Figure 8. Configuring Two-Phase Commit



StarSQL supports the use of two-phase commit protocols. Check the box below to enable this feature.

☑ Enable Use of Two-phase Commit

3. Click the checkbox next to Enable Use of Two-Phase Commit to enable the option.
4. Select the radio button for TCP/IP DRDA (Microsoft DTC)
5. Click Summary, or click Next until the last option panel appears, and then click OK to save the DSN configuration.

The following sections describe the host requirements and setup for using the two-phase commit protocol.

Two-Phase Commit

To use two-phase commit,, the host system must support two-phase commit transactions. See "System Requirements" on page 10 for details about which host systems support two-phase commit.

StarSQL for Windows also includes a Resource Manager Administration Tool to help you monitor and manage two-phase commit transactions. There are two versions of the Resource Manager—one for 32-bit computers and one for 64-bit computers. When using two-phase commit on a 64-bit computer, install only the 64-bit version of the Resource Manager. Only one version of the Resource Manager and MMC Plug-in can be installed at one time, and the 64-bit version of the Resource Manager can be used by both 32-bit and 64-bit ODBC applications.

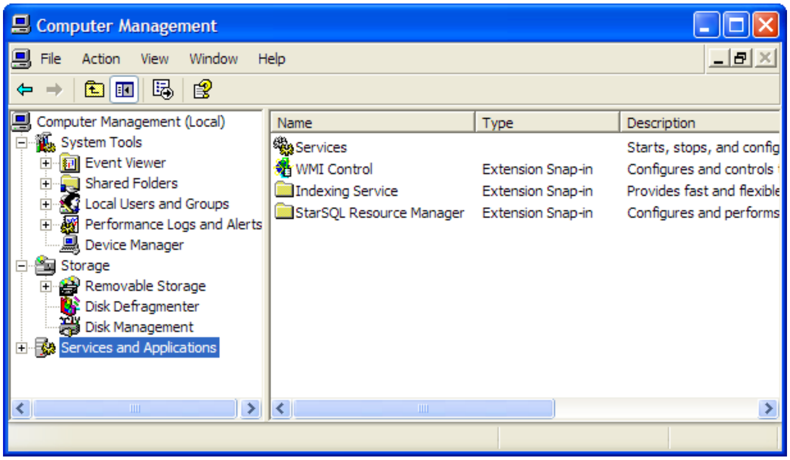
Using the StarSQL Resource Manager

The StarSQL Resource Manager provides support for distributed two-phase commit transactions. You must select the Custom installation for StarSQL to install the StarSQL Resource Manager service and Administration Tool. Note that the two-phase commit option does not appear when performing a Custom installation of the StarSQL 32-bit driver on a Windows x64 computer—use the 64-bit version of the Resource Manager on a x64 computer as it can be used by both 32-bit and 64-bit ODBC applications.

1. To access the StarSQL Resource Manager, start the Windows Computer Management tool. The Computer Management tool is available from the Administrative Tools category of the Windows Control Panel, or you can select My Computer and then right-click and select Manage from the context menu.

As shown in Figure 9, the StarSQL Resource Manager appears as a “snap-in” under the Services and Applications category of the Computer Management console.

Figure 9. StarSQL Resource Manager



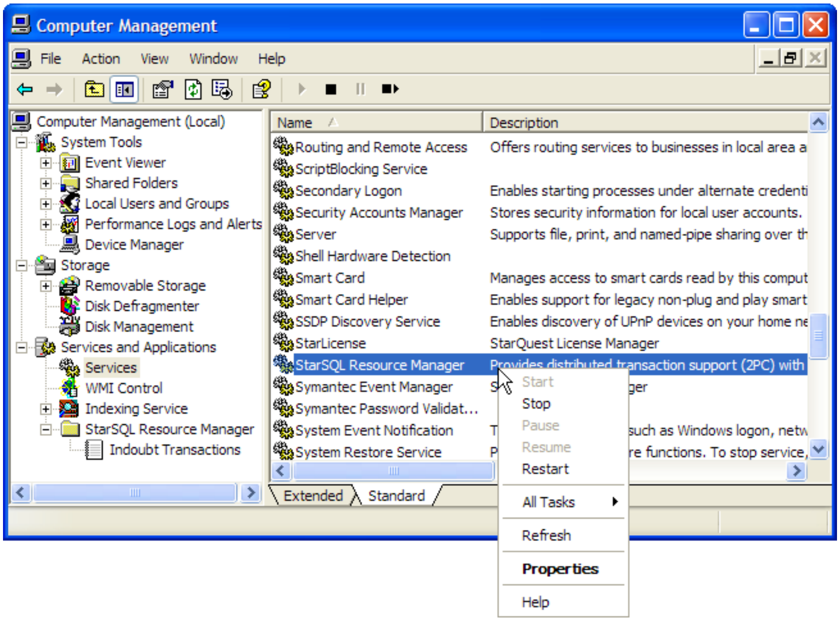
Name	Type	Description
Services		Starts, stops, and config
WMI Control	Extension Snap-in	Configures and controls
Indexing Service	Extension Snap-in	Provides fast and flexible
StarSQL Resource Manager	Extension Snap-in	Configures and performs

Press F1 or select the Help command to display the Microsoft Management Console help system—the StarSQL Resource Manager help appears as a book within the Microsoft Management Console help system.

The Resource Manager Service

To control the Resource Manager service, click on Services and select the StarSQL Resource Manager service in the right pane, as shown in Figure 10. The commands in the context menu allow you to start, stop, restart, and set startup properties for the service, just as you can for any Windows service.

Figure 10. Controlling the StarSQL Resource Manager Service



Name	Description
Routing and Remote Access	Offers routing services to businesses in local area a
ScriptBlocking Service	
Secondary Logon	Enables starting processes under alternate credenti
Security Accounts Manager	Stores security information for local user accounts.
Server	Supports file, print, and named-pipe sharing over th
Shell Hardware Detection	
Smart Card	Manages access to smart cards read by this comput
Smart Card Helper	Enables support for legacy non-plug and play smart
SSDP Discovery Service	Enables discovery of UPnP devices on your home ne
StarLicense	StarQuest License Manager
StarSQL Resource Manager	Provides distributed transaction support (2PC) with
Symantec Event Manager	S ger
Symantec Password Validat...	
System Event Notification	T such as Windows logon, netw
System Restore Service	P re functions. To stop service,

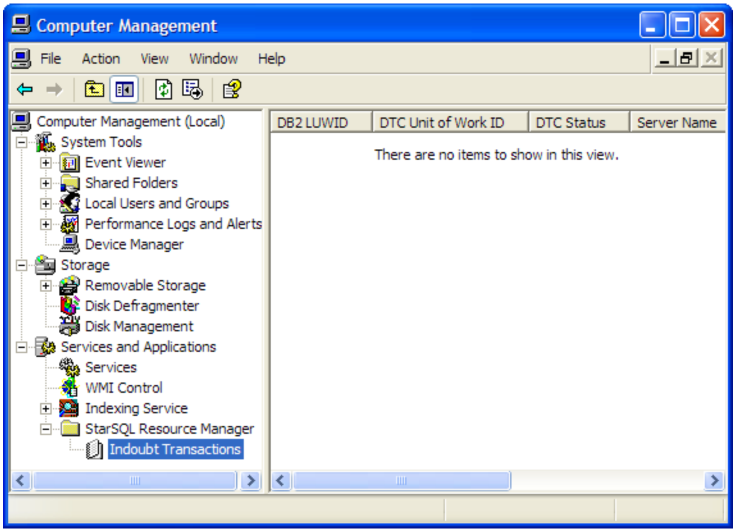
You can configure how the Resource Manager handles indoubt transactions by setting properties for the StarSQL Resource Manager application, as described in the next section.

The Resource Manager Administration Tool

The StarSQL Resource Manager works with the Distributed Transaction Coordinator (DTC) and DB2 to track the progress of each transaction. The components that manage the transaction status each maintain a log and communicate with each other to synchronize the status of the transaction.

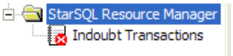
If the Resource Manager service is running and all transactions are processing normally, the message “There are no items to show in this view” appears when the Indoubt Transactions object is selected.

Figure 11. Transactions Processing Normally



There are no items to show in this view.

If the Resource Manager application cannot contact the Resource Manager service, a red X appears in the Indoubt Transactions icon, as shown below. Check the status of the Resource Manager service, as described in "The Resource Manager Service" on page 56.



Managing the Transaction Log

Under normal circumstances there is no need to manage the transaction logs—as a transaction progresses to being committed to the host database, the status record is automatically deleted from the transaction log. If the DTC commits a transaction, the

Resource Manager records the Committed status in the transaction log to communicate the outcome to DB2. If the Resource Manager cannot communicate with the DTC, it records the status as Unknown until the DTC reports the transaction should be committed or aborted.

If there is a critical failure that prevents DB2 from re synchronizing with the Resource Manager, or the Resource Manager from communicating with the DTC, you can manually clear transaction records from the log. For example, if you manually recover a transaction on the DB2 host, you also can delete the corresponding record from the Resource Manager transaction log as DB2 will no longer try to resynthesized that transaction with the Resource Manager. To delete a transaction record from the Resource Manager log, select the transaction from the list of Indoubt Transactions, right-click and select Delete.

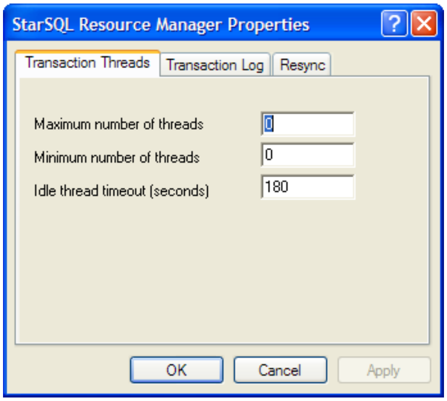
Note

Be extremely cautious about manually deleting a transaction record from the log. In accordance with the DRDA two-phase commit protocol, a transaction is assumed to be aborted if the components that are managing the transaction cannot obtain information about the outcome. If you clear a record from the log, DB2 will assume the transaction was aborted if it is in doubt about the transaction status when it re synchronizes with the Resource Manager.

Configuring the Resource Manager

Select the StarSQL Resource Manager object, right-click, and select the Properties command to configure how the Resource Manager handles indoubt transactions.

Figure 12. Configuring the Resource Manager



Maximum number of threads: 0
Minimum number of threads: 0
Idle thread timeout (seconds): 180

In the Resource Manager Properties you can specify:

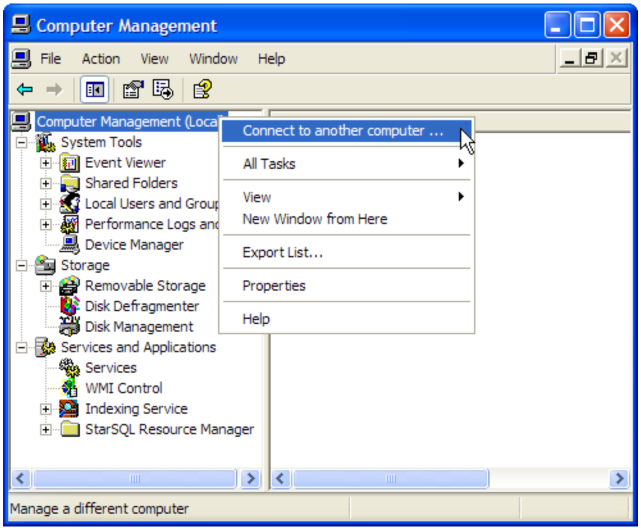
- the number of threads that are allocated to indoubt transactions, to control the number of indoubt transactions that can be active at one time
- the location and name of the transaction log file, and the maximum number of indoubt transactions that can be recorded in the log at one time
- the port number that DB2 uses to resynchronize the transaction status

If you change any of the Resource Manager properties, you must restart the service before the changes take effect.

Managing Remote Transaction Logs

As shown in Figure 13, from the Computer Management console Action menu, you can connect to another computer to remotely manage the StarSQL Resource Manager on that computer.

Figure 13. Connecting to StarSQL Resource Manager on a Remote Computer



Refer to the online help for the Microsoft Management Console for general information about using the Computer Management console.

CHAPTER 5

Binding Packages

A SQL package is an object that DB2 uses to process a SQL statement. Different packages are required to execute dynamic SQL, static SQL, and ODBC catalog functions.

StarSQL for Windows assumes that the dynamic SQL package and the catalog package already exist in the host database. On the initial connection with the host, StarSQL for Windows searches for the required packages, and if it does not find them, it creates them automatically.

The names and locations of the packages that are bound vary, depending on the values of certain configuration settings in the StarSQL for Windows Data Source definitions that are in effect when the initial connection is made.

If you are upgrading from a previous version of StarSQL for Windows, you may need to rebind existing packages on the host. See "Upgrading StarSQL" on page 15 for more information about migration issues.

The StarSQL for Windows user requires certain permissions to bind and use packages on the host. Typically, the packages will be created by an administrator and other users will be granted permission to use the packages.

Binding Packages with the Explain Option Enabled

DB2 provides an explain facility that provides detailed information about the access plan and environment of static or dynamic SQL statements. The captured information can help administrators understand how individual SQL statements are executed so the statement and database configuration can be tuned for performance. You can use a command-line tool or DB2 Visual Explain to display explain information.

The v5.4 release of StarSQL added support for a new keyword in the SWODBC.INI initialization file that binds all the StarSQL packages with the EXPLAIN bind option set to ALL. This includes dynamic SQL packages that are bound with StarAdmin, packages that are created and bound as needed by the StarSQL driver, and static SQL packages that are created with the StarScribe Package Editor.

The SWODBC.INI file is located in the \Programs subdirectory where the StarSQL software is installed. To enable the EXPLAIN bind option for all packages that are bound by StarSQL, add the following line to the [Defaults] section of the SWODBC.INI file:

```
[Defaults]
Explain=All
```

After you modify the initialization file to include the Explain=All statement, the explain information for packages that are subsequently bound by StarSQL can be displayed.

Catalog Package

The catalog package is needed to execute ODBC functions that query the system catalogs. The value of the SQL Catalog Schema in the StarSQL for Windows Data Source indicates the name of the catalog package.

Dynamic SQL Packages

There are several packages used for executing dynamic SQL. The default dynamic package varies by host platform; other dynamic packages will be used based on the BindRules, IsolationLevel, HeldCursors, and KeepDynamic settings configured in the data source. See the StarSQL for Windows Help for more information about these settings.

Table 7 shows the dynamic packages that can be bound; the default packages vary for different hosts and are shown in **boldface** type.

Table 7. Dynamic Packages

Package Name	Isolation Level and Held Cursors Setting
SWNC0000	Commit None/Held Cursors no (default on **DB2 for i**)
SWRC0000	Read Committed / Held Cursors no (default on **DB2 UDB** for Windows and UNIX)
SWRR0000	Repeatable Read / Held Cursors no
SWRU0000	Read Uncommitted / Held Cursors no
SWTS0000	Serializable / Held Cursors no
SWRC1000	Read Committed / Held Cursors yes
SWRR1000	Repeatable Read / Held Cursors yes
SWRU1000	Read Uncommitted / Held Cursors yes
SWTS1000	Serializable / Held Cursors yes

Table 8 through Table 10 show the packages that can be bound when using the KeepDynamic option, or if the DYNAMICRULES option is set to BIND. These tables are only applicable for hosts running DB2 for z/OS.

Table 8. Dynamic Packages on DB2 for z/OS with KeepDynamic=Yes

Package Name	Isolation Level and Held Cursors Setting
SWNC4000	Commit None/Held Cursors no
SWRC4000	Read Committed / Held Cursors no
SWRR4000	Repeatable Read / Held Cursors no
SWRU4000	Read Uncommitted / Held Cursors no
SWTS4000	Serializable / Held Cursors no
SWRC5000	Read Committed / Held Cursors yes (default on **DB2 for z/OS**)
SWRR5000	Repeatable Read / Held Cursors yes
SWRU5000	Read Uncommitted / Held Cursors yes
SWTS5000	Serializable / Held Cursors yes

Table 9. Dynamic Packages on DB2 for z/OS with DYNAMICRULES=BIND and KeepDynamic=No

Package Name	Isolation Level and Held Cursors Setting
SWRC2000	Read Committed / Held Cursors no
SWRR2000	Repeatable Read / Held Cursors no
SWRU2000	Read Uncommitted / Held Cursors no
SWTS2000	Serializable / Held Cursors no

Package Name	Isolation Level and Held Cursors Setting
SWRC3000	Read Committed / Held Cursors yes
SWRR3000	Repeatable Read / Held Cursors yes
SWRU3000	Read Uncommitted / Held Cursors yes
SWTS3000	Serializable / Held Cursors yes

Table 10. Dynamic Packages on DB2 for z/OS with DYNAMICRULES=BIND and KeepDynamic=Yes

Package Name	Isolation Level and Held Cursors Setting
SWRC6000	Read Committed / Held Cursors no
SWRR6000	Repeatable Read / Held Cursors no
SWRU6000	Read Uncommitted / Held Cursors no
SWTS6000	Serializable / Held Cursors no
SWRC7000	Read Committed / Held Cursors yes
SWRR7000	Repeatable Read / Held Cursors yes
SWRU7000	Read Uncommitted / Held Cursors yes
SWTS7000	Serializable / Held Cursors yes

Permissions for Packages

For Binding Packages

StarAdmin is an application (available as a separate download) used to bind packages on the host. As a convenience, you may want to use StarAdmin to bind all the required packages when setting up StarSQL for Windows to avoid permission issues when users try to bind them later.

If the required catalog and dynamic SQL packages do not already exist, StarSQL for Windows automatically creates them on the initial connection to the host.

The user who performs the initial connection or who is binding packages with StarAdmin, must have permission to bind packages in the host database. On DB2 for i, the DBA must grant CREATE permission for this user for the specified library. On other hosts, the DBA must grant CREATE IN COLLECTION and BINDADD permissions. The binding will fail if the user does not have these permissions.

After the StarSQL for Windows packages have been bound, the DBA administrator can revoke these permissions.

For Using Packages

After the packages have been bound, the DBA must grant StarSQL for Windows users permission to execute them. On DB2 for i, users must be granted *USE permission on the packages, which can usually be done by the package owner. On other hosts, the DBA must grant StarSQL for Windows users EXECUTE or RUN permissions on the packages.

Static and Catalog Packages

The StarSQL for Windows user executes applications that use the catalog package and any static SQL packages under the permissions of the package owner. No additional permissions are required to use these packages.

Dynamic SQL Packages

On all hosts, except for DB2 for z/OS, for applications that use dynamic SQL, the StarSQL for Windows user needs explicit permissions to read (SELECT) and write (UPDATE/INSERT/DELETE) the columns and tables accessed by the application. The DBA needs to grant these permissions to “public” or to the various groups.

On DB2 for z/OS, the required permissions depend on the setting of the DYNAMICRULES option. If the DYNAMICRULES option is set to RUN, which is the default, the StarSQL for Windows user needs explicit permissions to read and write the columns and tables accessed by the application. If the DYNAMICRULES option is set to BIND, the user has the permissions of the package owner, except that the following SQL statements cannot be executed regardless of the permissions of the package owner:

- SET CURRENT SQLID
- GRANT
- REVOKE
- ALTER

- CREATE
- DROP
- Any SQL statement that cannot be prepared as a dynamic SQL statement.

See the StarSQL for Windows help for instructions on setting the DYNAMICRULES option.

Granting Use Permissions

To grant EXECUTE authority on the SQL packages, you can use the StarAdmin application, or you can execute SQL statements similar to those shown in the following sections

For DB2 for z/OS

Using SPUFI on the host or the SQL passthrough mode of an ODBC-enabled application, execute the following SQL statements:

```
GRANT EXECUTE ON PACKAGE
STARSQL.SYSIBM, STARSQL.SWRC5000
TO PUBLIC
```

For DB2 for Linux, UNIX, and Windows

Using the DB2 Command Line Processor or the SQL passthrough mode of an ODBC-enabled application, execute the following SQL statements:

```
GRANT EXECUTE ON PACKAGE
STARSQL.SYSCAT, STARSQL.SWRC0000
TO PUBLIC
```

For DB2 for i

Using STRSQL (which is the interactive SQL interpreter, available in the Query Manager and SQL Development Kit for AS/400, Licensed Program 5722-ST1, 5761-ST1 or 5770-ST1), or using the SQL passthrough mode of an ODBC-enabled application, execute the following SQL statements:

```
GRANT EXECUTE ON PACKAGE
STARSQL.QSYS2, STARSQL.SWNC0000
TO PUBLIC
```

In addition, you can use AS/400 authority commands to grant *USE authority for all users (*PUBLIC) to the library packages:

1. From a 5250 session, enter **`WRKLIB <`***`package library name`***`>`**.

2. Select Option 12 (work with objects).

3. Select Option 2 (edit authority on each object one at a time).

4. Change **PUBLIC *EXCLUDE** to **PUBLIC *USE**.

Another alternative is to use the GRTOBJAUT command from a 5250 session to grant *USE authority for the library and EXECUTE authority for the packages to all users. The following example assumes that the package library is named “STARSQL.”

```
GRTOBJAUT OBJ(STARSQL) OBJTYPE(*LIB) USER(*PUBLIC)
AUT(*USE)
GRTOBJAUT OBJ(STARSQL/*ALL) OBJTYPE(*ALL)
USER(*PUBLIC) AUT(*USE).
```

AutoBind Option

The AutoBind option forces SQL packages to be bound at connect time (SQLConnect or SQLDriverConnect). You can set AutoBind directly in the Expert Setting Page or pass it in the connect string to *SQLDriverConnect()*.

⚠ **Caution**

The use of AutoBind always causes binding to occur and adversely affects connect time performance.

The AutoBind option settings are described in Table 11.

Table 11. AutoBind Option Settings

AutoBind Value	Description
0 (Default)	No auto-bind at connect time. The driver still binds packages on-the-fly if it does not find them on the server when it needs them.
1	Binds up to three packages at connect time, but uses NO REPLACE option on bind. If the packages already exist they will not be replaced. In the current version of StarSQL, AutoBind=1 ignores the NO REPLACE option and functions in the same manner as AutoBind=2.
2	Binds up to three packages at connect time and always replaces current packages, if there are any.

AutoBind Value	Description
3	Binds the Package File (for static SQL) only at connect time, always replacing the current package if there is any.
Y	Same as 3.
N	Same as 0.

Regardless of the AutoBind setting, the StarSQL driver binds packages dynamically if it does not find them on the host at the time they are needed.

The three packages that may be bound by AutoBind=1 and AutoBind=2 are:

- the dynamic package for the current transaction isolation level
- the catalog package
- the static package file (a static SQL package file is bound only if static SQL matching is enabled for the data source)

For specific information about the AutoBind options and the default settings, refer to the StarSQL Help system for the Data Source Configuration Wizard.

Glossary

APAR - Authorized Program Analysis Report A request for correction of a problem caused by a defect in a current unaltered release of a program.

APPL An APPL statement defines the DB2 subsystem to VTAM for the purposes of remote access. It is required for any configuration that involves a DB2 host on an IBM mainframe (z/OS).

authorization identifier On DB2 for z/OS, the authorization identifier is user's login ID or an assigned Authorization Identifier, which corresponds to a group with which the user is associated.

BSDS The BSDS (bootstrap data set) is a VSAM data set that contains name and status information for DB2, as well as RBA range specifications, for all active and archive log data sets, passwords for the DB2 directory and catalog, and lists of conditional restart and checkpoint records.

CCSID CCSID (Coded Character Set Identifier) represents a character set, code page, encoding scheme, and additional coding-related information. Used to support international code sets.

collection A collection is a location on the host for database objects, such as user tables and catalog tables, which are collected together under a single qualifying name.

CoS Class of Service (CoS) is the ability of switches and routers to prioritize traffic into different queues and classes.

data source A data source describes a connection to a database from an ODBC application.

DDF DDF (Distributed Data Facility) is the vehicle that DB2 uses to send and receive remote procedure calls.

DRDA DRDA (Distributed Relational database Architecture) supports access to distributed data by which an application can explicitly connect to another location, using an SQL statement, to execute packages that have been previously bound at that location.

DSN A DSN (Data Source Name) is an ODBC definition that refers to a particular database.

Dynamic SQL Dynamic SQL refers to SQL statements that are prepared and executed within an application program while the program is executing. A dynamic SQL statement can change during program execution.

held cursor A held cursor is a cursor that is not automatically closed when its transaction commits.

installer image An installer image is the set of StarSQL for Windows files and a Setup.exe program that installs the StarSQL for Windows software.

isolation level The isolation level refers to the degree of concurrency permitted in a transaction.

ODBC ODBC (Open Database Connectivity) is a call-level interface developed by Microsoft Corporation that allows a single application to access DBMSs from different vendors using a single interface.

Open Edition Open Edition is a component of the MVS operating system that supports TCP/IP processing for certain levels of OS/390.

package A package is an object on the host containing a set of SQL statements that have been bound statically and are available for processing.

PTF Program Temporary Fix - a method used by IBM for distributing fixes quickly.

RDB Name An RDB name is a unique identifier for an RDBMS within a network.

REGEDIT The registry editor supplied by the Windows operating system for manually editing the Registry.

system catalogs The system catalogs are tables in the host database that DB2 uses to keep track of the system. On some RDBMSs, they are called the system tables.

TCP/IP TCP/IP (Transmission Control Protocol / Internet Protocol) is a commonly - used network protocol.

user id A user id is a unique identifier that enables a user to logon to a host.

VTAM VTAM (Virtual Telecommunications Access Method) provides network communications on IBM z/OS systems.

VSAM Virtual Storage Access Method - A data storage system used in IBM z/OS. VSAM was designed to organize data more efficiently and to improve access time by searching indexes instead of actual files.

Index