Novell®

# SQL *Connector*

## Installation Guide

**Printing Date:** **October 1, 1999**

© Copyright 1999 Novell, Inc. and B2Systems, Inc. All rights reserved. Printed in the USA.

The software described in this document is furnished under a license, and may be used or copied only in accordance with the terms of that license. No part of this document may be reproduced in any form or by any means without the written permission of Novell, Inc. and B2Systems, Inc.

The information in this document is subject to change without notice, and should not be construed as a commitment by Novell, Inc. or B2Systems, Inc. Every effort has been made to ensure that the information contained herein is accurate and complete. However, Novell, Inc. and B2Systems, Inc. assume no responsibility for any errors that may appear in this document.

SQL *Connector* is a trademark of Novell, Inc. and B2Systems, Inc.

NetWare is a registered trademark of Novell, Inc. Microsoft Windows NT is a registered trademark of Microsoft Corporation, and Microsoft SQL Server is a trademark of Microsoft Corporation.

Other product names are trademarks or registered trademarks of their respective holders, and are mentioned for reference only.

# About This Manual

## Purpose of this Manual

This manual describes the installation of SQL *Connector* on client (first tier), data broker (second tier) and data driver (third tier) operating systems. SQL *Connector* installations must be carefully planned to insure success. This manual provides directions and guidelines for the multiple tiers which must be configured for an SQL *Connector* installation.

## Intended Audience

This document is intended for system administrators who will be installing SQL *Connector*. A working knowledge of Java, ODBC, JDBC, database systems, operating systems and networking is required to install and test SQL *Connector*.

## Structure of this Manual

This manual consists of chapters which describe how to install SQL *Connector*. There are separate chapters for each tier in the installation process.

## Associated Documents

The SQL *Connector* document set contains these manuals:

- SQL *Connector Overview*
- SQL *Connector Installation Guide*
- SQL *Connector Administration Guide*
- SQL *Connector SQL Grammar Manual*
- SQL *Connector ODBC Programmer's Guide*
- SQL *Connector JDBC Programmer's Guide*

## Conventions

When there are differences in commands, examples, or syntax between operating systems, the following abbreviations are used:

| Abbreviation | Meaning |
|---|---|
| NetWare | the Novell NetWare operating system |
| Windows | the Microsoft Windows 95/98/NT operating systems |

# Table of Contents

**About This Manual . . . . . I–3**

**Table of Contents . . . . . I–4**

**1 Installation Planning . . . . . I–5**
1.1 Components . . . . . I–5
1.2 Architecture . . . . . I–5
1.3 Prerequisites . . . . . I–7
1.3.1 Client Components . . . . . I–7
1.3.2 Data Broker and Local Data Driver Components . . . . I–7
1.3.3 Remote Data Driver Components . . . . . I–7

**2 Broker Installation Procedure . . . . . I–8**
2.1 Introduction . . . . . I–8
2.2 Configuration Procedure . . . . . I–8
2.2.1 Startup Procedure . . . . . I–9
2.2.2 Shutdown Procedure . . . . . I–9
2.2.3 Internet Host Names and Addresses . . . . . I–9
2.2.4 Data Driver Setup . . . . . I–9
2.3 Administration . . . . . I–10
2.4 Verification . . . . . I–10

**3 Client Installation . . . . . I–11**
3.1 Introduction . . . . . I–11
3.2 Procedure . . . . . I–11
3.2.1 ODBC Driver Installation . . . . . I–11
3.2.2 JDBC Driver Installation . . . . . I–12
3.2.3 NetWare Administration Extensions . . . . . I–12
3.2.4 Verification . . . . . I–13

**4 Remote Data Driver Installation . . . . . I–14**
4.1 Configuration Information . . . . . I–14
4.2 Windows NT Remote Data Driver Installation . . . . . I–14
4.2.1 Introduction . . . . . I–14
4.2.2 Database Support . . . . . I–15
4.2.3 Procedure . . . . . I–15
4.2.4 Verification . . . . . I–16

**A NetWare Administration . . . . . I–17**
A.1 Overview . . . . . I–17
A.2 Command Information . . . . . I–17

# 1

# Installation Planning

## 1.1 Components

SQL *Connector* is a Data Request Broker that provides a multiple-tier, multiple-database enterprise environment for connecting ODBC (Open Database Connectivity) applications and JDBC (Java Database Connectivity) applications between client systems and database servers.

SQL *Connector* includes *Client*, *Data Broker* and *Data Driver* components. These components can reside on the same or different systems within a network. The *client components* support the connection of SQL *Connector* to end user applications or application development tools that use ODBC or JDBC. The *Data Broker components* support client connectivity over a network, query parsing, optimization and distribution. The Data Broker components also support database administration for maintaining the SQL *Connector* Data Sources.

The *Data Driver components* support local or remote connectivity to physical databases. These components are summarized in the following tables. Note that flat files can be used transparently as a precursor to using a physical database.

| Component Name | Component Functionality | Usage |
|---|---|---|
| **Windows 95/98/NT Client Components** | | |
| SQL *C*–ODBC | Microsoft® ODBC™ Driver API | optional |
| SQL *C*–JDBC | JavaSoft™ Java™ SQL classes | optional |
| **NetWare Data Broker Components** | | |
| SQL *C*–DSA | Data Source Administrator | required [a] |
| SQL *C*–DRB | Data Request Broker | required |
| **Local and Remote Data Driver Components** | | |
| SQL *C*–ORA-DD | Oracle® Database Connection | optional |
| SQL *C*–ODBC–DD | ODBC Database Connection | optional |

a. Required for database administrators, not required for developers or users.

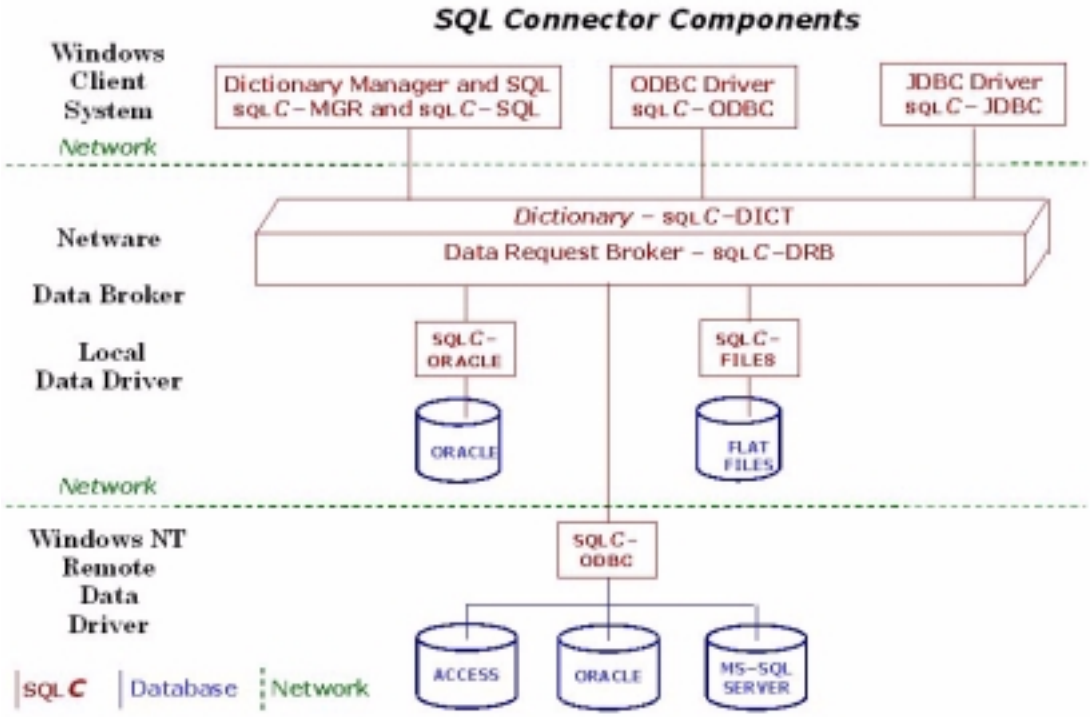
## 1.2 Architecture

The SQL *Connector* architecture includes the Client, Data Broker and Data Driver components listed in the above table. The architecture is designed for maximum flexibility in meeting the needs of enterprise wide application development and deployment. The components have been modularized so that they can be installed on multiple tiers within an enterprise network.

*Local Data Drivers* execute on the same system as the Data Broker. *Remote Data Drivers* execute on systems that are networked to the Data Broker system. This version of SQL *Connector* supports the following local and remote Data Drivers:

| System | Oracle | ODBC |
|---|---|---|
| **Novell Netware** | Local | Remote |

The complete component architecuture is shown below:




**Note:** The ODBC Client Component and the ODBC Data Driver are separate modules. The ODBC Client Component is an interface between end–user client applications and the Data Broker. The ODBC Data Driver is a data source (like Oracle, Sybase, etc.) for the Data Broker. Multiple ODBC data sources can connect through the Data Broker into a single ODBC client application.

# 1.3 Prerequisites

## 1.3.1 Client Components

The client requires:

1. Windows 95/98/NT client operating system (includes ODBC–32).
2. For Java/JDBC application development, Java Development Kit (JDK) 1.1.8 or higher, available at: http://www.javasoft.com/products/jdk/1.1/.
3. For ODBC application development, ODBC-32 2.5 or higher.

## 1.3.2 Data Broker and Local Data Driver Components

The Data Broker requires:

- NetWare 5.1 .
- Administration access for installation.
- NetWare modules dsapi.nlm, netdb.nlm, nlsapi.nlm, nlslsp.nlm and lsapi.nlm must be loaded.
- NetWare Oracle module oci7stub.nlm must be loaded.

The following local Data Drivers are supported on NetWare 5.1:

- Oracle 8.1.5

## 1.3.3 Remote Data Driver Components

The Data Drivers are available for use by SQL *Connector* in connecting to physical databases. These Data Drivers require the versions of operating systems listed in the following table

| System | ODBC |
|---|---|
| **Windows NT 4.0** | 2.5 or higher |

**Note:** The version numbers listed in the above table represent the highest version numbers of the databases that have been verified for proper operation with SQL *Connector*. Lower versions of the databases may function correctly if the database vendor has maintained backwards compatibility. Contact Customer Support for additional information.

# 2

# Broker Installation Procedure

## 2.1 Introduction

The SQL *Connector* architecture includes Client, Data Broker and Data Driver components as discussed in the previous chapter. These components can be installed on multiple systems within an enterprise network.

The SQL *Connector* Data Broker is a set of programs that respond to data requests from SQL *Connector* Clients and send data requests to SQL *Connector* Local and Remote Data Drivers. The SQL *Connector* Local and Remote Data Drivers provide database access to local and remote databases

The installation procedure is designed for a multiple tier configuration. The Data Broker and Local Data Drivers are typically installed on a single Netware server (middle tier). The Client components can be installed on multiple end-user systems (first tier), and the Remote Data Driver can be installed on multiple Windows NT database servers (third tier).

When Netware is installed, the SQL *Connector* Data Broker and Local Data Driver files are placed in the directory SYS:SYSTEM\SQLC, and the SQL *Connector* Client and Remote Data Driver installation files are placed in the directory SYS:PUBLIC\SQLC. The PUBLIC copies can be used multiple times as needed to install Client components and Remote Driver components on other systems.

The remainder of this chapter discusses configuration, administration and verification procedures for the Data Broker.

The following chapters discuss installing the Client and Data Driver components on other systems (first and third tiers).

## 2.2 Configuration Procedure

When Netware is installed, the Data Broker is automatically installed and configured. The installation has performed these actions:

- Create and enable SQL *Connector* startup and shutdown files.
- Edit the network file sys:etc\services.
- Extends the NDS User Schema to support SQL *Connector* environment variables. See "NetWare Administration Extensions" for additional information.

**Note:** The Internet hosts file must be manually edited. See "Internet Host Names and Addresses" for additional information.

### 2.2.1 Startup Procedure

The configuration procedure automatically creates a startup file sqlc.ncf and edits autoexec.ncf to call the startup file. The startup file has the following commands:

```
search add SYS:\SYSTEM\SQLC
load SQLCMON <username> <password>
load VTXNETD -p1958 -n
load SQLCODBC
```

**Note:** The NetWare modules dsapi.nlm, lsapi.nlm, netdb.nlm, nls.nlm, and nlsapi.nlm must be loaded prior to loading sqlcmon.nlm.

### 2.2.2 Shutdown Procedure

The configuration procedure automatically creates a shutdown file sqlckill.ncf.

The shutdown file has the following commands:

```
unload sqlcodbc
unload vtxodbc
load vtxkill -p1958 <server name>
unload vtx16
pause
unload vtxnetd
unload sqlcmon
```

This shutdown file will be automatically called when SQL *Connector* is shutdown by executing the following commands.

```
load SQLCADM
SQLCADM> shutdown
```

### 2.2.3 Internet Host Names and Addresses

The SQL *Connector* components (Client, Broker, Data Drivers) use Internet host names and/or Internet Protocol (IP) addresses (nnn.nnn.nnn.nnn) for communications. All of the components will work with addresses used in place of node names. Node names are typically designated system-wide using a naming service such as DNS (Domain Naming Service). If DNS is not enabled, then the following system files *must be* edited to map host names and IP addresses for the SQL *Connector* components to communicate.

| Operating System | File Name and Location |
|---|---|
| Windows NT | %SYSTEM_ROOT%\system32\drivers\etc\hosts |
| Windows 95/98 | %WINDIR%\hosts |
| Netware 5.x | SYS:ETC\HOSTS |

### 2.2.4 Data Driver Setup

The system file sys:etc/services is automatically edited to include the Data Driver TCP/IP service name and port number. The default port numbers are:

```
ODBC:       sqliodbcserv    8106/tcp
```

See Chapter 4 for more information on setting Data Driver TCP/IP port numbers.

## 2.3 Administration

The following programs are used by the SQL *Connector* Data Request Broker:

```
VTXNETD.NLM
VTX16.NLM
VTXODBC.NLM
VTXKILL.NLM
SQLCMON.NLM
SQLCODBC.NLM
SQLC.NLM
SQLCADM.NLM
```

`VTXNETD.NLM` is a multi-threaded network listener that processes database requests from ODBC and JDBC client applications. The listener will assign a unique thread to each client application. All client applications must use a TCP/IP port number which matches the "p" argument specified when `VTXNETD.NLM` is loaded. The *ODBC Programmer's Guide* and the *JDBC Programmer's Guide* provide additional information on setting the port number used by the client application.

`VTX16.NLM` is a multi-threaded request processor that translates between network requests and database requests, and provides data buffering over the network. For example, multiple (small) rows returned by an SQL SELECT statement will be combined into a single (large) buffer for transmission over the network.

`VTXODBC.NLM` is a network processor that handles NetWare clients that make ODBC requests, such as those from the Netscape Enterprise Server.

`VTXKILL.NLM` is a utility for stopping `VTXNETD.NLM`.

`SQLCMON.NLM` is a database monitor that assigns each client thread to its own database broker and monitors the activity of each thread. The database broker consists of a database engine that connects to one or more local or remote physical databases and includes an internal memory pool for each database engine.

`SQLCODBC.NLM` is a processor that connects network ODBC requests to the database engine.

`SQLC.NLM` is a database engine that provides the physical database connections and processes the physical database requests (`SQL SELECT, INSERT, UPDATE, DELETE, etc.`) from each client thread.

`SQLCADM.NLM` is a database engine administrator which provides control, logging and status information for each database engine in the database monitor. See Appendix A for information about running `SQLCADM.NLM`.

**Note:** The `shutdown` option of `SQLCADM` will unload `SQLCMON.NLM` and `VTXNETD.NLM`.

## 2.4 Verification

After the SQL *Connector* Data Broker and local Data Driver components are installed and loaded (see following chapters), they can be tested by creating Data Sources and using the odbcping, odbcjoin, jdbcping and jdbcjoin programs, which are documented in the *ODBC Programmer's Guide* and the *JDBC Programmer's Guide*. These programs can be found on SYS:PUBLIC\SQLC\SAMPLES.

# 3

# Client Installation

## 3.1 Introduction

The SQL *Connector* ODBC Driver (SQL *C*–ODBC) and the SQL *Connector* JDBC Driver (SQL *C*–JDBC) are Client components that connect to the SQL *Connector* Data Request Broker (SQL *C*–DRB). SQL *C*–JDBC requires Javasoft JDK or JRE 1.1.8 or higher (see Chapter 1 for additional information).

The SQL *Connector* NetWare Administrator Extensions allow a NetWare Administrator to set SQL *Connector* environment variables.

## 3.2 Procedure

For ODBC application *development* on Windows, the following steps are necessary:

- SQL *C*–ODBC driver installation

For ODBC application *deployment* on Windows, the following steps are necessary:

- SQL *C*–ODBC driver installation

For JDBC application *development* on Windows, the following steps are necessary:

- Java Development Kit installation (download from www.javasoft.com)
- SQL *C*–JDBC driver installation

For JDBC application *deployment* on Windows, the following steps are necessary:

- Jave Runtime installation (download from www.javasoft.com)
- SQL *C*–JDBC driver installation

### 3.2.1 ODBC Driver Installation

The SQL *C*–ODBC installation procedure can be started by executing the setup file on the software distribution path. The file can be executed by double-clicking the file name using the Windows Explorer or by browsing for the file using the Start –> Run Menu command. The setup file has the following location:

```
SYS:PUBLIC\SQLC\Client\ODBC\disk1\setup.exe
```

The installation procedure has no options. The Destination Directory location is:

```
C:\Program Files\SQLC\ODBC
```

Files are copied from the installation directory to the system directory and to the Destination Directory, and ODBC entries are made in the system registry.

The following files are installed in the `Windows NT system32` directory:

```
tod32.dll:      SQL Connector ODBC driver
vtx3.dll:       Communications layer between ODBC and network
```

The SQL *C*–ODBC Driver can then be used in client ODBC applications. See the *ODBC Programmer's Guide* for additional information.

Readme and uninstall information is copied to the Destination Directory.

## 3.2.2 JDBC Driver Installation

### 3.2.2.1 Windows

The SQL *C*–JDBC installation procedure can be started by executing the setup file on the software distribution path. The file can be executed by double-clicking the file name using the Windows Explorer or by browsing for the file using the Start –> Run Menu command. The setup file has the following location:

```
SYS:PUBLIC\SQLC\Client\JDBC\setup.exe
```

The first prompt is the Destination Directory. The default location is:

```
C:\Program Files\SQLC\JDBC
```

The destination directory can be changed by browsing for an alternate location, which will be created if it does not exist.

The installation procedure will create a JAR file in the Destination Directory, e.g.:

```
sqlcjdbc.jar
```

This jar file should be placed in the Java CLASSPATH so it can be accessed by Java programs such as third party application development environments that will use the SQL *Connector* JDBC classes. See the *JDBC Programmer's Guide* for additonal information.

Uninstall information is also placed in the same directory.

### 3.2.2.2 NetWare

The same jar file can be copied to a NetWare system for executing Java JDBC applications or browser applets. The file should be copied to SYS:\JAVA\CLASSES, and the NetWare CLASSPATH should be modified to include the jar file. See the *JDBC Programmer's Guide* for additional information.

## 3.2.3 NetWare Administration Extensions

### 3.2.3.1 Installation

The NetWare Administration Extensions installation procedure can be started by executing the setup file on the software distribution path. The file can be executed by double-clicking the file name using the Windows Explorer or by browsing for the file using the Start | Run Menu command. The setup file has the following location:

```
SYS:PUBLIC\SQLC\NetWare\NDSAdmin\disk1\setup.exe
```

<SOFTWARE> is the distribution path specified during the first phase of the installation (see previous chapter). The Destination Directory location for uninstall information is:

```
C:\Program Files\SQLC\NDS
```

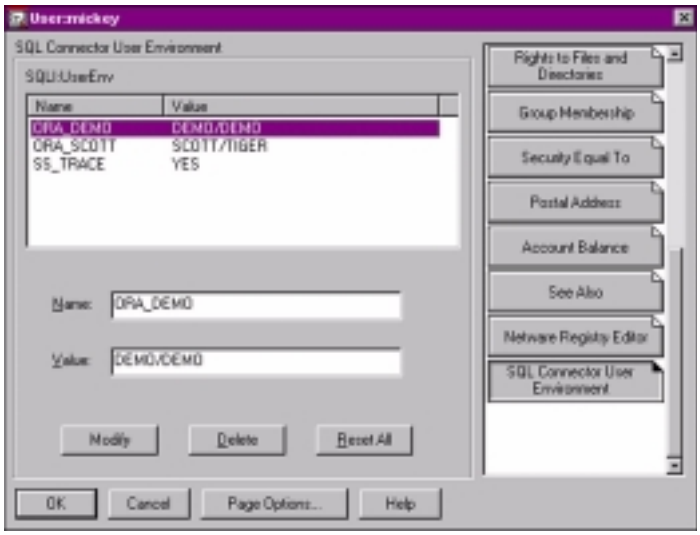
The installation procedure prompts for the location of the NetWare 5.x Administration program and then copies files to the Administration directory and make registry entries for NDS Snapins. The following files are installed in the Admininistration directory:

```
sqlcndserv.dll
snapins\sqlcuserenv.dll
```

### 3.2.3.2 Usage

**User Extension**

After the installation, the NDS User Object has been extended as shown in the following figure (double-click on a user name in NDS Admin to see this extension).




## 3.2.4 Verification

### 3.2.4.1 ODBC Driver Verification

The SQL *C*–ODBC driver can be tested using a sample program odbcping that is discussed in the *ODBC Programmer's Guide*. The sample program files are in the software distribution path as follows:

```
SYS:PUBLIC\SQLC\Samples\Windows\odbc\odbcping.c
SYS:PUBLIC\SQLC\Samples\Windows\odbc\odbcping.exe
SYS:PUBLIC\SQLC\Samples\NetWare\odbc\odbcping.c
SYS:PUBLIC\SQLC\Samples\NetWare\odbc\odbcping.nlm
```

The odbcping program will attempt an ODBC connection from a Netware or Windows client to an SQL *Connector* Data Source through a Data Broker running on a network server. See the *ODBC Programmer's Guide* for more details.

### 3.2.4.2 JDBC Driver Verification

The SQL *C*–JDBC driver can be tested using a sample program jdbcping that is discussed in the *JDBC Programmer's Guide*. The sample program files are in the software distribution path as follows:

```
SYS:PUBLIC\SQLC\Samples\Windows\jdbc\jdbcping.java
SYS:PUBLIC\SQLC\Samples\Windows\jdbc\jdbcping.class
SYS:PUBLIC\SQLC\Samples\NetWare\jdbc\jdbcping.java
SYS:PUBLIC\SQLC\Samples\NetWare\jdbc\jdbcping.class
```

The Windows and NetWare Java programs are identical. The jdbcping program will attempt a JDBC connection from a Netware or Windows client to an SQL *Connector* Data Source through a Data Broker running on a network server. See the *JDBC Programmer's Guide* for more details.

# 4

# Remote Data Driver Installation

## 4.1 Configuration Information

The SQL *Connector* Remote Data Drivers are Internet server programs that run on remote Windows NT systems. Each remote system has an Internet server process that *listens* for incoming requests from the SQL *Connector* Data Broker and *routes* the requests to the appropriate Data Driver, which connects to the physical database. The SQL *Connector* Data Request Broker and the SQL *Connector* Remote Data Drivers communicate using two pieces of information:

1. A designated TCP/IP port number or TCP/IP named service that corresponds with a designated port number. The service name or port number must match on the Broker (client) and Driver (server) sides of the connection. The SQL *Connector* Data Broker will send out TCP/IP requests using this service name or port number.
2. An internal remote Data Source. This Data Source stores the names of the remote physical database tables that have been chosen by the user of the Data Broker. The database table names must match on the Broker (client) and Driver (server) sides of the connection.

## 4.2 Windows NT Remote Data Driver Installation

### 4.2.1 Introduction

The Broker and Data Drivers communicate using a TCP/IP process that emulates the standard Internet service and configuration files (`inetd` and `inetd.conf`). Since there is no standard inetd server process on Windows NT systems, SQL *Connector* provides a substitute process named `sqlc_inetd.exe` and a substitute configuration file named `sqlc_inetd.conf` (the NT `services` file is not used).

This service (sqlc_inetd) runs on the same system as the Data Drivers. It listens for Broker service requests on the TCP ports associated with each of the services listed in its configuration file (`sqlc_inetd.conf`). When a request arrives, `sqlc_inetd` executes the server program associated with the service. The configuration file is as follows:

```
# Used by sqlc_inetd.exe to connect a broker to a database driver
# Note that sqlc_inetd (SQL Connector Listener service)
# must be restarted after this file is changed.
# Port_number  database_driver
8106        ODBC\sqlcserver_odbc.exe
```

The pathname of the Data Driver must be relative to the pathname of the server process (`sqlc_inetd.exe`). If the Data Driver is installed using a different relative pathname, then the above configuration file must be edited accordingly.

### 4.2.2 Database Support

The following remote Data Driver is currently available for Windows NT systems. Please contact Customer Support for an updated list.

- ODBC 2.5 or higher

The remote Data Driver receives requests through the Internet server process (service name is SQLC–server Listener, program name is sqlc_inetd.exe) that listens for incoming requests from the SQL *Connector* Data Request Broker and routes the requests to the physical database. The Internet server process may be started manually at any time or automatically when the system is booted. The installation procedure for each Data Driver will copy files to the installation directory and create registry entries. Each installation procedure will then inquire about the startup options.

The Data Driver installation procedures may be run by any user who has read/write access to the installation directory. The Internet server process must be started or stopped by a user with Administrator privilege.

### 4.2.3 Procedure

The installation procedure can be started by executing the setup file on the software distribution path (see previous chapter). The file can be executed by double-clicking the file name using the Windows Explorer or by browsing for the file using the Start –> Run Menu command. The installation procedures use the following setup files and default folder:

| Database | Program | Default Folder on System Disk |
|---|---|---|
| ODBC | \Drivers\NT\ODBC-OUT\disk1\setup.exe | \Program Files\SQLC\Drivers\ODBC |

The installation procedure will first prompt for the destination folder. The default location may be changed by browsing for another folder, which will be created if it does not exist.

After the files are copied, there will be a prompt for the startup choices for the NT service. The choices are:

```
Automatic – restarts when the system is booted
Manual – start from Services applet in Control Panel
```

The default choice is Automatic.

The last prompt is a choice to view the readme.txt file. After the last prompt, the installation procedure is finished.

The Windows NT services file should be checked to insure there are no conflicts with the TCP/IP port numbers defined in the sqlc_inetd.conf. If there are conflicts, then the port numbers in sqlc_inetd.conf should be changed to eliminate the conflicts.

### 4.2.4 Verification

The installation may be verified by starting the SQL *Connector* Listener service. The Service Icon in the Control Panel can be used to check the status of the service. The Data Driver is functioning correctly when the NT service can be started.

The SQL *Connector* Listener service can only be started by a user with Administrator privileges. If the privileges are not present, the error message is:

```
SQLC–server Listener – Start
Could not start the SQLC-server Listener service on \\<node>.
Error 0005: Access is denied.
```

If this error occurs, then try again as a user with Administrator privilege.

A Windows NT Data Driver is used whenever the SQL *Connector* Data Request Broker attempts a connection to a remote Data Driver on NT.

# A

# NetWare Administration

## A.1 Overview

SQLCADM.NLM is a database engine administrator which provides control, logging and status information for each database engine that is executing in the database monitor. After loading SQLCADM.NLM, a list of commands can be obtained by typing:

```
> help
```

The output is:

```
Commands are:
 Status             -- Display overall status of SQL Connector
 Purge              -- Kill non-connected SQL Connector engines
 Shutdown           -- Shutdown SQL Connector
 Engine first incr max -- Set SQL Connector engine parameters
   initial -- number of engines on startup
   incr    -- number of engines to start when non-available
   max     -- max number to run concurrently
 Logging {on|off}  -- Turn logging on or off
 Save               -- Persist settings
 Restore            -- Restore settings from the persistence file
 SetEnv name value -- Set environment variable
 GetEnv name        -- Display environment variable
 Help               -- Display this text
 Exit               -- To exit program
```

## A.2 Command Information

### A.2.1 Status

Command format: Status

The status command will display information about SQLCMON. This includes:

```
Number of free SQL Connector engines
Number of SQL Connector engines initially created
Number of SQL Connector engines per allocation cycle
Maximum number of simultaneous SQL Connector engines
Number of SQL Connector engines in use

Size of the initial SQL Connector engine memory pool
Size of additional SQL Connector engine memory pools
Number of additional SQL Connector engine memory pools
Maximum amount of memory per SQL Connector engine
```

### A.2.2 Purge

Command format: `Purge`

The purge command causes SQLCMON to kill and deassign SQLC engines.

### A.2.3 Shutdown

Command format: `Shutdown`

The Shutdown command performs a purge and then causes SQLCMON to exit. It will also cause SQLCADM to exit (since SQLCADM can not commuiuncate with SQLCMON once it shuts down).

### A.2.4 Engine

Command format: `Engine first incr max`

The Engine command controls the creation of SQLC engines. The `first` parameter is the number of SQLC engines that SQLCMON will create when it first starts. The `incr` parameter is the number of SQLC engines to start whenever there are not any available and one is required. The `max` parameter is the maximum number of SQLC engines that can be running at any time.

### A.2.5 Logging

Command format: `Logging on` or `Logging off`

The Logging command turns on or off event logging. SQLCMON will log key events to the log file `SQLCMON.LOG`. At the time an event is logged, it is also displayed on the console.

### A.2.6 SetEnv

Command format: `SetEnv name value`

The SetEnv command sets environment variables for SQLCMON. The `name` is case sensitive and the `value` is an uninterpreted string (no escape character processing). Environment variables are discussed in the *SQL Grammar Manual*. The environment variables become global values for all client applications.

In particular, most client application development environments use double quotes around table names and column names. To inform SQLC engines about this behavior, the following command can be used:

```
setenv  SS_QUOTED_NAMES  yes
```

### A.2.7 GetEnv

Command form: `GetEnv name`

The GetEnv command displays the environment value associated with `name`. This value is set with the SetEnv command. The `name` variable is case insensitive.

### A.2.8 Save

Command form: `Save`

The Save command persists the current parameter and environment to a data file. The data file is read when `SQLCMON` starts to initialize the internal state of `SQLCMON`.

### A.2.9 Restore

Command form: `Restore`

The Restore command resets the state of `SQLCMON` parameters and environment variables to the values that are saved in the persistent data file by the Save command.

### A.2.10 Help

Command form: `Help`

The Help command provides a limited explanation of all of the commands available to `SQLCADM`.

### A.2.11 Exit

Command form: `Exit`

The Exit command causes `SQLCADM` to terminate.