HP NonStop ODBC Server Installation and Management Manual

Abstract

This manual describes installation and configuration of the HP NonStop™ ODBC Server, a product that allows programs developed for the Microsoft ODBC or SQL Server interfaces to access an HP NonStop SQL/MP database on a HP system. The manual also describes management of the NonStop ODBC Server and its resources after installation.

Product Version

NonStop ODBC Server 2.0

Supported Release Version Updates (RVUs)

This guide supports D32.00 and all subsequent D-series RVUs, and G01.00 and all subsequent G-series RVUs until otherwise indicated in a new edition.

Part Number	Published
429395-002	September 2003

Document History

Part Number	Product Version	Published
130619	NonStop ODBC Server 2.0	November 1996
133900	NonStop ODBC Server 2.0	June 1997
136331	NonStop ODBC Server 2.0	April 1998
429395-001	NonStop ODBC Server 2.0	May 2001
429395-002	NonStop ODBC Server 2.0	September 2003

HP NonStop ODBC Server Installation and Management Manual

2. Installing the NonStop ODBC Server (continued)

3. Using the NonStop ODBC Server Configuration Manager

3. Using the NonStop ODBC Server Configuration Manager (continued)

4. Managing the NonStop ODBC Server

4. Managing the NonStop ODBC Server (continued)

5. Managing NonStop ODBC Server Resources

5. Managing NonStop ODBC Server Resources (continued)

A. Summary of System Installation Defaults

Index

Figures

Figures (continued)

Tables

What's New in This Manual

Manual Information

Abstract

This manual describes installation and configuration of the NonStop™ ODBC Server, a product that allows programs developed for the Microsoft ODBC or SQL Server interfaces to access a NonStop SQL/MP database on a HP system. The manual also describes management of the NonStop ODBC Server and its resources after installation.

Product Version

NonStop ODBC Server 2.0

Supported Release Version Updates (RVUs)

This guide supports D32.00 and all subsequent D-series RVUs, and G01.00 and all subsequent G-series RVUs until otherwise indicated in a new edition.

Part Number	Published
429395-002	September 2003

Document History

Part Number	Product Version	Published
130619	NonStop ODBC Server 2.0	November 1996
133900	NonStop ODBC Server 2.0	June 1997
136331	NonStop ODBC Server 2.0	April 1998
429395-001	NonStop ODBC Server 2.0	May 2001
429395-002	NonStop ODBC Server 2.0	September 2003

New and Changed Information

This publication has been updated to reflect new product names:

- Since product names are changing over time, this publication might contain both HP and Compaq product names.
- Product names in graphic representations are consistent with the current product interface.

- The technical content of this publication is not updated and reflects the state of the product at the D32.00 and all subsequent D-series RVUs, and G01.00 and all subsequent G-series RVUs.

This edition contains the following changes:

- The Transaction Auto Commit option is removed throughout the manual because the driver cannot set this option globally at the Data Source Administrator Level. This has been done in the following sub-sections of the manual:
    - Under Installing the Client on the PC, on page 2-31.
    - Under Examples of Installation, on pages 2-39 and 2-42.
- The IOP Name option for parallel TCP/IP is added under Create a New System Catalog With NOSCOM on page 2-39 and under Step Three: Add the Net Service Configuration on page 3-42.
- Since Sybase no longer provides DBLIB libraries or releases DBLIB based software, references to DBLIB and NETLIB being provided by Sybase or Microsoft are removed from this manual. Further, the note under Major Steps for Installing the NonStop ODBC Server on page 2-2, is updated to state that ODBC/MP and Sybase no longer support Sybase DBLIB or DBLIB-based applications.
- All references to Windows 3.1, Windows 3.1 NEC and 3.11 Workgroup are removed because they are no longer supported. The reference to the 16-bit driver is also removed because it is considered matured.

About This Manual

This manual describes the installation, configuration, management, and tuning of the HP NonStop ODBC Server and its components.

The NonStop ODBC Server is database connectivity software that allows programs written for use with the Microsoft ODBC (Open Database Connectivity) product, as well as those written for use with either the Microsoft SQL Server or the Sybase SQL Server, to access NonStop SQL/MP databases. Currently only Sybase or Microsoft SQL Server 4.9.2 DBLIB features are supported through NonStop ODBC.

ODBC clients are supported on Windows workstations; SQL Server clients are supported on DOS, Windows, OS/2, and UNIX workstations. The HP product, the NonStop ODBC Server, runs primarily on the NonStop system, although one of its components, the NonStop ODBC driver, runs on the client workstation.

Audience

This manual is written for system administrators and database administrators who:

- Initially install the NonStop ODBC Server or migrate from an earlier version of the NonStop ODBC Server
- Prepare HP NonStop SQL/MP databases for access with the NonStop ODBC Server
- Prepare application programs to run with the NonStop ODBC Server
- Modify ODBC or SQL Server application programs to run with the NonStop ODBC Server

End users of applications that access NonStop SQL/MP databases can also use this manual for its language and feature descriptions.

Readers of this manual should know:

- The ODBC CORE SQL dialect of the SQL language or the SQL Server TRANSACT-SQL dialect
- The fundamentals of NonStop SQL/MP
- The fundamentals of the HP NonStop Kernel
- The operation of the PC or workstation applications being used

Organization of This Manual

This manual contains the following sections:

- Section 1, Introduction, gives an overview of the installation process, configuration, and maintenance procedures.

- Section 2, Installing the NonStop ODBC Server, describes how to install a NonStop ODBC Server system, including procedural steps for installing on the NonStop system and on the PC or UNIX workstation, configuration of the NonStop ODBC Server, and troubleshooting techniques during installation.
- Section 3, Using the NonStop ODBC Server Configuration Manager, describes a graphical user interface (GUI) tool provided with the NonStop ODBC Server for doing system configuration directly from a PC.
- Section 4, Managing the NonStop ODBC Server, explains how to manage a NonStop ODBC Server system effectively, describing the configuration of server classes; techniques for improving performance, including the use of statement caching; security implications; using error and event messages to diagnose problems; and using the trace features provided by the NonStop ODBC Server.
- Section 5, Managing NonStop ODBC Server Resources, describes facilities provided by the NonStop ODBC Server to account for the use of NonStop ODBC Server resources and to govern that use to prevent excessive cost of unmonitored queries.
- Section A, Summary of System Installation Defaults, lists the default values, set at system installation time by the installation process, for named attributes of the NonStop ODBC Server.

Related Manuals

Two other manuals in the NonStop ODBC Server library are:

- *NonStop ODBC Server Reference Manual* — This manual contains reference information for the NonStop ODBC Server, describes the ODBC or SQL Server features and statements that the NonStop ODBC Server supports, and describes how the NonStop ODBC Server accommodates the differences between ODBC or SQL Server and NonStop SQL/MP
- *NonStop ODBC Server Messages Manual*—documents the error and warning messages generated by the NonStop ODBC Server components.

ODBC Documentation

For information about Microsoft ODBC, version 2.0, see the *Microsoft ODBC 2.0 Programmer’s Reference and SDK Guide*, © 1994, Microsoft Corporation, ISBN 1-55615-658-8 (available at most bookstores that carries computer-related books).

For additional information about ODBC CORE SQL, see the *X/Open and SQL Access Group SQL CAE specification* (1992).

For information about workstation tools that are compatible with the NonStop ODBC Server, see the “NonStop ODBC Server Data Sheet,” which your HP analyst can obtain for you.

SQL Server Documentation

For information about SQL Server, see the Microsoft SQL Server library. If you are using the Sybase SQL Server software, use the Sybase manuals rather than the Microsoft manuals.

NonStop SQL/MP Documentation

All users need the following NonStop SQL/MP manuals:

- *NonStop SQL/MP Reference Manual*—provides basic NonStop SQL/MP reference information. The manual includes all standard NonStop SQL/MP statements and discusses essential concepts for using the statements, such as locking, parallelism, buffering, DDL/DML concurrency considerations, and DEFINEs.
- *NonStop SQL/MP Messages Manual*—lists the NonStop SQL/MP messages for all NonStop SQL/MP components. It also includes file-system messages that can be issued only on NonStop SQL/MP objects

If you are unfamiliar with NonStop SQL/MP, the *Introduction to NonStop SQL/MP* provides feature, functional, and conceptual overviews.

Other HP Documentation

The following manuals contain information about the HP systems and software products used with the NonStop ODBC Server:

- *File Utility Program (FUP) Reference Manual*—describes the utility program used for managing files on NonStop systems.
- *Guardian User's Guide*—describes how to use the HP NonStop operating system.
- *TACL Reference Manual*—presents the syntax and operations of the standard commands and functions available in the command interpreter.
- *NonStop Transaction Management Facility (TMF) Planning and Configuration Guide* and *NonStop Transaction Management Facility (TMF) Operations and Recovery Guide*—describe how to use NonStop Transaction Management Facility to protect a database against disk, system, or program failures.
- *TCP/IP Management Programming Manual*—describes the programmatic interface to the HP TCP/IP data communication software.
- *Multilan/TLAM Programming Manual*—describes the programmatic interfaces provided by the HP Tandem LAN Access Method (TLAM) application programs.
- *PTrace Reference Manual*—describes how to use the PTrace utility to display trace files created through the use of the Subsystem Control Facility (SCF).
- *TS/MP Pathsend and Server Programming Manual* and the *Pathway/iTS TCP and Terminal Programming Guide*–describes how to develop stored procedures in the Pathway environment.

- The Subsystem Control Facility (SCF) manual for the communication protocol you are using.

Sun Workstations

For users who are accessing the NonStop ODBC Server from a Sun workstation, the following manuals contain background information about Sun UNIX:

- *SunOS Reference Manual*—describes UNIX calls and commands available from a Sun interface terminal.
- *System and Network Administration Manual*—contains administrative information about using a Sun system.

These manuals are included with a Sun workstation and are available from Sun Microsystems.

Notation Conventions

Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

> This requirement is described under Backup DAM Volumes and Physical Disk Drives on page 3-2.

General Syntax Notation

This list summarizes the notation conventions for syntax presentation in this manual.

UPPERCASE LETTERS. Uppercase letters indicate keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

```
MAXATTACH
```

lowercase italic letters. Lowercase italic letters indicate variable items that you supply. Items not enclosed in brackets are required. For example:

`file-name`

computer type. `Computer type` letters within text indicate C and Open System Services (OSS) keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

```
myfile.c
```

italic computer type. *Italic computer type* letters within text indicate C and Open System Services (OSS) variable items that you supply. Items not enclosed in brackets are required. For example:

```
pathname
```

[ ] Brackets. Brackets enclose optional syntax items. For example:

```
TERM [\system-name.]$terminal-name
```

```
INT[ERRUPTS]
```

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
FC [ num  ]
   [ -num ]
   [ text ]
```

```
K [ X | D ] address
```

{ } Braces. A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }
                  { $process-name  }
```

```
ALLOWSU { ON | OFF }
```

| Vertical Line. A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

… Ellipsis. An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address [ , new-value ]…
```

```
[ - ] {0|1|2|3|4|5|6|7|8|9}…
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char…"
```

Punctuation. Parentheses, commas, semicolons, and other symbols not previously described must be typed as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;
```

```
LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example:

```
"[" repetition-constant-list "]"
```

Item Spacing. Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items:

```
$process-name.#su-name
```

Line Spacing. If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] LINE

   [ , attribute-spec ]…
```

!i and !o. In procedure calls, the !i notation follows an input parameter (one that passes data to the called procedure); the !o notation follows an output parameter (one that returns data to the calling program). For example:

```
CALL CHECKRESIZESEGMENT (  segment-id                   !i
                         , error        ) ;             !o
```

!i,o. In procedure calls, the !i,o notation follows an input/output parameter (one that both passes data to the called procedure and returns data to the calling program). For example:

```
error := COMPRESSEDIT ( filenum ) ;                     !i,o
```

!i:i. In procedure calls, the !i:i notation follows an input string parameter that has a corresponding parameter specifying the length of the string in bytes. For example:

```
error := FILENAME_COMPARE_ (  filename1:length          !i:i
                            , filename2:length ) ;      !i:i
```

!o:i. In procedure calls, the !o:i notation follows an output buffer parameter that has a corresponding input parameter specifying the maximum length of the output buffer in bytes. For example:

```
error := FILE_GETINFO_ (  filenum                       !i
                        , [ filename:maxlen ] ) ;       !o:i
```

Notation for Messages

This list summarizes the notation conventions for the presentation of displayed messages in this manual.

Bold Text. Bold text in an example indicates user input typed at the terminal. For example:

```
ENTER RUN CODE

?123

CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

Nonitalic text. Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

```
Backup Up.
```

lowercase italic letters. Lowercase italic letters indicate variable items whose values are displayed or returned. For example:

```
p-register

process-name
```

[ ] Brackets. Brackets enclose items that are sometimes, but not always, displayed. For example:

```
Event number = number [ Subject = first-subject-value ]
```

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
proc-name trapped [ in SQL | in SQL file system ]
```

{ } Braces. A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
obj-type obj-name state changed to state, caused by
{ Object | Operator | Service }

process-name State changed from old-objstate to objstate
{ Operator Request. }
{ Unknown.          }
```

| Vertical Line. A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
Transfer status: { OK | Failed }
```

% Percent Sign. A percent sign precedes a number that is not in decimal notation. The % notation precedes an octal number. The %B notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

```
%005400
```

```
%B101111
```

```
%H2F
```

P=%*p-register* E=%*e-register*

Notation for Management Programming Interfaces

This list summarizes the notation conventions used in the boxed descriptions of programmatic commands, event messages, and error lists in this manual.

UPPERCASE LETTERS. Uppercase letters indicate names from definition files. Type these names exactly as shown. For example:

```
ZCOM-TKN-SUBJ-SERV
```

lowercase letters. Words in lowercase letters are words that are part of the notation, including Data Definition Language (DDL) keywords. For example:

```
token-type
```

!r. The !r notation following a token or field name indicates that the token or field is required. For example:

```
ZCOM-TKN-OBJNAME      token-type ZSPI-TYP-STRING.          !r
```

!o. The !o notation following a token or field name indicates that the token or field is optional. For example:

```
ZSPI-TKN-MANAGER      token-type ZSPI-TYP-FNAME32.         !o
```

Change Bar Notation

Change bars are used to indicate substantive differences between this manual and its preceding version. Change bars are vertical rules placed in the right margin of changed portions of text, figures, tables, examples, and so on. Change bars highlight new or revised information. For example:

> The message types specified in the REPORT clause are different in the COBOL85 environment and the Common Run-Time Environment (CRE).
>
> The CRE has many new message types and some new message type codes for old message types. In the CRE, the message type SYSTEM includes all messages except LOGICAL-CLOSE and LOGICAL-OPEN.

1 Introduction

To use the HP NonStop ODBC Server, you must first install it. After it is in operation, you need to manage it to keep it running at an optimum level. This manual describes these activities.

Installation of the NonStop ODBC Server involves the following activities outlined in this section:

Management of the NonStop ODBC Server involves additional configuration of the system, improving performance by various techniques, using diagnostic messages and the Subsystem Control Facility (SCF) to check on system status and performance, and establishing security criteria.

Installation

Installation is performed systematically, as shown in the following steps:

1. Prepare for the installation by doing the following.
    - Read and understand the documentation.
    - Evaluate the desired characteristics of the system.
    - Check the system—both from the NonStop side and from the client platforms—to be sure all aspects are operational.
    - Perform any necessary backup operations.
2. Install the NonStop ODBC Server by doing the following:
    - Install the software.
    - Perform. an initial configuration of the software.
    - Start the SQL Communication Subsystem (SCS) and the NonStop ODBC servers.
3. Install the clients. This includes:
    - Installing the client connectivity software and the client application software.
    - Configuring the client.
    - Ensuring connectivity to both the NonStop system and the NonStop ODBC Server.

4. Complete the installation by doing the following

    - If needed, make a more detailed configuration of the NonStop ODBC Server on the NonStop system and on client platforms.
    - Optionally install stored procedures.
    - Verify that the entire system functions correctly.

Both first-time installation, and migration to the latest release of the NonStop ODBC Server are explained in Section 2, Installing the NonStop ODBC Server.

Configuration

Configuration of the NonStop ODBC Server is important to both installation and management of the system. Therefore, configuration is described in both Section 2, Installing the NonStop ODBC Server , and in Section 4, Managing the NonStop ODBC Server.

The NonStop ODBC Server runs without requiring much configuration. However, extensive configuration interfaces are available to meet the demands of a wide application spectrum. Configuration involves:

- the mapping of client objects to NonStop objects,
- setting process attributes on the NonStop system,
- setting attributes of the NonStop ODBC Server for function and performance.

You can use the utility program NOSUTIL, through its command interpreter, NOSCOM, to configure the ODBC Server from a terminal on the NonStop system. This interface is described under “Managing Customized Catalogs,” in the NonStop *ODBC Server Reference Manual*.

Alternatively, you can use a PC connected to a LAN to configure the system by using the NonStop ODBC Server Configuration Manager, a GUI (graphical user interface) tool. This interface is described in Section 3, Using the NonStop ODBC Server Configuration Manager.

Management

Management of the NonStop ODBC Server is a continuing process, beginning with detailed configuration of the system after it is first installed, and going on to fine-tune the operation of the system for optimum performance, based on customer needs, query traffic, and system load.

Management involves:

- system and data source configuration,
- improving performance by using techniques such as statement caching,

- diagnosing problems using Event Management Service (EMS) messages,
- using the Subsystem Control Facility (SCF) interface to check on the status of an SCS process that uses the trace feature provided by the NonStop ODBC Server,
- and developing your own applications and scripts for monitoring the system.

These management operations and techniques are described in Section 4, Managing the NonStop ODBC Server.

Resource management is also available. Section 5, Managing NonStop ODBC Server Resources, describes the resource accounting facility, which provides the ability to record resource use for analysis purposes, and the resource governing facility, which provides the ability to control database resources used by SQL statements, set a limit for resource use, and defer or stop any query that exceeds the limit.

2

Installing the NonStop ODBC Server

This section describes how to install and configure the HP NonStop ODBC Server, on both the HP NonStop system and the client PC or workstation, and includes the following:

- Overview of the installation procedure, including migration from earlier versions
- Recommended preparation before installing the NonStop ODBC Server
- Installation steps for the NonStop system
- Installation steps for PCs and UNIX workstations
- Configuration guidelines
- Troubleshooting techniques

Installation Overview

The installation of the NonStop ODBC Server includes:

1. Prepare for the installation on page 2-2

2. Install the NonStop ODBC Server on page 2-2

3. Install the client systems on page 2-3

4. Complete the configuration on page 2-3

NonStop ODBC Server Release 2 software includes features that support three possible installation types:

- New installation. NonStop ODBC Server has not been previously installed.
- Concurrent installation.   The NonStop ODBC Server Release 1 software is currently installed and is to be left untouched by installation of NonStop ODBC Release 2. The two installations can be run independently and can access the same NonStop SQL/MP data. Use this type to migrate from Release 1 to Release 2.
- Upgrade installation. The NonStop ODBC Server Release 1 software is currently installed and the NonStop ODBC Release 2 catalog tables are to be installed in the same location, replacing the Release 1 catalog tables. The Release 1 catalog tables will no longer be usable. Use this type to upgrade catalog tables.

When migrating from NonStop ODBC Release 1, HP recommends that you use the Concurrent installation type. This simplifies the migration—there is only one NonStop ODBC configuration group at a time—and provides available backup software.

Note. The Upgrade installation permanently changes NonStop ODBC metadata tables. No tool is provided to undo an Upgrade installation.

Major Steps for Installing the NonStop ODBC Server

Here is an overview of the major steps for installing the NonStop ODBC Server.

1. Prepare for the installation

- Read the NonStop ODBC Server documentation, including the software documents (softdocs) shipped with the product to understand what the NonStop ODBC Server can and cannot do. Understand the configuration issues. Learn what features the NonStop ODBC Server defaults provide for you.
- Understand your application's characteristics: what kind of SQL operations are involved, are sessions long or short, how many clients, what type of LAN is used, what are the response time requirements?
- Check conditions on your NonStop system. Check the versions of software. Make sure all necessary subsystems, such as SQL/MP, are operational.
- Check the status of the client platforms. Check hardware and software requirements. Make sure clients have LAN connectivity to the NonStop system.
- Perform any necessary backups. BACKUP can be used to save SQL table values, allowing later recovery using RESTORE.

2. Install the NonStop ODBC Server

Install the NonStop ODBC Server on the NonStop system by using the provided HP Tandem Advanced Command Language (TACL) macro NOSINST.

- Install or update the NonStop ODBC Server software.
- Create or upgrade the NonStop ODBC Server system catalog.

Note. Concurrent and Upgrade installations require more than the minimal user interaction to complete. Specific instructions are listed later in this section.

- Provide a minimum of information to establish an initial default SQL Communication Subsystem (SCS) and NonStop ODBC Server configuration.
- Start the NonStop ODBC Server using this minimal configuration:
  - a named SCS
  - a defined collection of NonStop ODBC NOS Server processes
  - a NonStop ODBC NOSUTIL Configuration Server process

3. Install the client systems

- Install the client software for connectivity. Use the NonStop ODBC installer for ODBC clients. Follow the Microsoft/Sybase documentation for DBLIB.

Note.

- If your PC already has TTE or RSC installed, or if you intend to use these products, be sure to read Checking the PC Clients on page 2-7 to see about sharing the same communication stack.
- ODBC/MP and Sybase no longer support Sybase DBLIB or DBLIB-based applications. However, ODBC/MP continues to support Sybase DBLIB-based applications if they can successfully negotiate the TDS 4.2 or TDS 4.6 message protocols on which DBLIB is based. ODBC/MP no longer releases DBLIB-based client libraries.

- Configure the client and ensure connectivity to the NonStop ODBC Server on the NonStop system.
- Install the client application and verify correct operation. For example, check that the NonStop ODBC Server is connected to the correct database with correct attributes, and so on.

4. Complete the configuration

- Complete the NonStop ODBC Server configuration on the NonStop system (and on client platforms if needed).
- Install stored procedures for execution through the NonStop ODBC Server (optional).
- Verify that the entire system works correctly, including management options such as tracing.

The NonStop ODBC Server is designed to run without requiring configuration beyond the minimal information requested by the installation macro. However, extensive configuration interfaces are available to meet the demands of a wide application spectrum. Configuration involves setting process attributes on the NonStop system, setting attributes of the NonStop ODBC Server for function and performance, and setting the attributes of SQL/MP for function and performance.

If You Use This Input Device…	**Then Use This Configuration Utility…**	**Additional Reference Information**
Terminal on a NonStop System	NOSCOM, the command interface to the NOSUTIL utility	*NonStop ODBC Server Reference Manual*
Windows(R) PC connected to a LAN	NonStop ODBC Server Configuration Manager	Section 3, Using the NonStop ODBC Server Configuration Manager

The PC interface requires that a minimal administrator configuration, such as that created during the NOSINST LOAD step, be defined and started

Configure the SCS process:

- the network service name,
- process-creation attributes,
- NonStop ODBC component location (SCSOBJ, NOS, NOSUTIL), and
- optional startup messages for SCS and NOS.

Configure a NonStop ODBC Server server class:

- usernames associated with that server class,
- CPUs to run in,
- priority,
- maximum number of processes,
- number of available (idle) processes, and
- the association between a NonStop ODBC Server server class and the SCS parent process.

Configure the NonStop ODBC servers:

- SQL statement caching,
- tracing,
- resource accounting and resource governing attributes, and
- NonStop ODBC Server session defaults.

This section provides a step-by-step procedure to install the NonStop ODBC Server, so you do not need a thorough understanding of configuration details.

After installation, a NonStop ODBC Server system must be managed to support new applications, new users, and changing workloads. Managing the system involves many of the configuration aspects of installing it, but requires a deeper understanding of determining and enhancing its operating status. Managing the system is described in Section 4, Managing the NonStop ODBC Server.

Note. Each installation step builds on the results of the previous steps, so it is important to do each installation step in sequence and to verify that the step has been completed correctly before proceeding to the next step.

The NonStop ODBC Server includes several components and configuration interfaces. In general, you could perform the installation steps in a variety of sequences. For simplicity, however, this manual specifies and recommends that *one* sequence of steps be followed.

Features Affecting Installation

Release 2 of the NonStop ODBC Server lets you install the NonStop ODBC system catalog in a location other than the subvolume of the SQL/MP system catalog. This feature allows multiple, concurrent, independent installations of the NonStop ODBC system on a single NonStop system.

This feature is controlled using a DEFINE to indicate the location of the NonStop ODBC Server system catalog. You set this DEFINE in your TACL session before starting a primary configuration.

This feature allows NonStop ODBC software to support the ODBC style definition of DATASOURCE independent of the SQL/MP data structure optimized for OLTP. The distinguishing attribute of the ODBC DATASOURCE model is that the list of objects made available to a client connecting to a named datasource is chosen to reflect the application the user is performing. The software lists only those objects needed by the application.

In an environment where both decision support and OLTP applications run concurrently, optimization of user and application access competes with optimization of data store operations for insert and update. By allowing DATASOURCEs to be defined as either coincident with the underlying data, as in the NonStop ODBC Release 1 product, or as a selective projection across all data groupings, as introduced in NonStop ODBC Release 2, the NonStop ODBC Product supports the widest possible range of access connectivity. This feature is provided without requiring change or replication to the underlying SQL/MP data.

Installation Versus Migration

Installation means bringing up a version of the NonStop ODBC Server on a system from the beginning (as with a brand new system) or in a location different from any other existing installation of the software. Migration means bringing up a version of the NonStop ODBC Server when a different version of the NonStop ODBC Server has been previously installed and the new version is to be installed into the same software and system catalog locations as the previous version.

If you remove your old version entirely, then you can do a new installation. If you are an existing customer, however, you will probably perform a concurrent installation. Then, after completing your own certification, you can remove the NonStop ODBC Release 1 software and associated metadata. This ensures availability by allowing the existing installation to persist during the installation of NonStop ODBC Release 2.

Configuration data for usernames, alias names, profiles, and trace definitions can be extracted from NonStop ODBC Release 1 catalogs and used to populate the NonStop ODBC Release 2 configuration data, whether performing concurrent installation or an upgrade migration. For more information, see the description for the utility NOSU200 in Step 8—Run NOSINST on page 2-18.

The following considerations apply to migration:

- If a previous version of the NonStop ODBC Server is already installed:

  Migration involves installing the new NonStop ODBC Server software, moving to it, and upgrading the catalogs. It is assumed that existing clients will run without change on the newer software (they can be migrated later to use the newer NonStop ODBC driver). Tools are provided to extract configuration specifications from the SCS edit files and the NonStop ODBC system catalog metadata tables (Z*).

  Note. Migration assumes that NonStop ODBC Release 1 is installed, as signified by the system catalog version 100 or 110.

- If you want to upgrade clients to new versions of DBLIB or ODBC, note that in most cases, the client software and the NonStop ODBC Server software on the NonStop system do not need to be upgraded simultaneously. In some cases, however, it may be necessary for the client and NonStop ODBC Server software upgrades to be synchronized—this is noted in the softdoc when applicable. In general, if both NonStop systems and client systems are to be upgraded, it is best to upgrade both at the same time.
- Unlimited statement caching is no longer supported in Release 2. In release 1, SQL_MAX_STATEMENT_CACHE = -1 meant no limit on number of statements cached. Release 2 does not support this unbounded cache, imposing a limit of 32767. If the -1 value is used, you must change the value it to 32767 before using NOSU200 to migrate the catalog.

Under NonStop ODBC Release 2, more than one version of NonStop ODBC Server software can run on a given node. Each catalog on a node can run a different version of NonStop ODBC Server software. Note, however, that special rules exist for managing SQL/MP data that is customized for ODBC access from more than one NonStop ODBC catalog when multiple software versions are present.

Migration for Clients

The considerations for client migration are as follows:

- If you are upgrading a DBLIB client, follow the Microsoft/Sybase instructions and verify that the new client can connect to and use the NonStop ODBC Server on the NonStop system.
- If you are upgrading an ODBC client, do one of the following:
  - To migrate to a new version of the ODBC driver manager, follow Microsoft's instructions.
  - To migrate to a new version of the NonStop ODBC driver, follow the installation instructions for Windows Setup, described in Installing the Client on the PC on page 2-27.
  - To migrate to a new version of a communications stack, follow the vendor's instructions.

Preparation

To prepare for the installation, check requirements on all systems involved and plan your installation strategy. If you currently have an old NonStop ODBC Server installed, you might want to do backups first. See When to Back Up on page 2-13.

Understand the Application

Before you begin installation, read the NonStop ODBC Server documentation and understand the applications to be supported. Questions to consider are:

- What type of application is it? What type of SQL load will there be (little queries, big queries, updating)? How many clients will be connected? How much SQL activity is expected from each client? What are the security needs (a lot, not much, many or few usernames)?
- What are the network needs? How many clients are expected? How many SCS processes should be configured?
- What does SQL/MP need? Will you use existing databases or build new ones? How many logical user names will there be on the NonStop system? How many alias names?
- What are the performance needs? Is a fast connection needed? Is caching of SQL compilations needed?

Checking the NonStop System

Verify that the following conditions are met on the NonStop system. You can use VPROC to determine the release version of most software.

- A NonStop system running the D32 or later version of the NonStop Kernel
- SQL/MP D32 or later version
- Transaction Management Facility (TMF) software
- Pathway/TS D40 or later version (if the stored procedure execution feature is used)
- For TCP/IP connections, HP TCP/IP D40 or later version

Checking the PC Clients

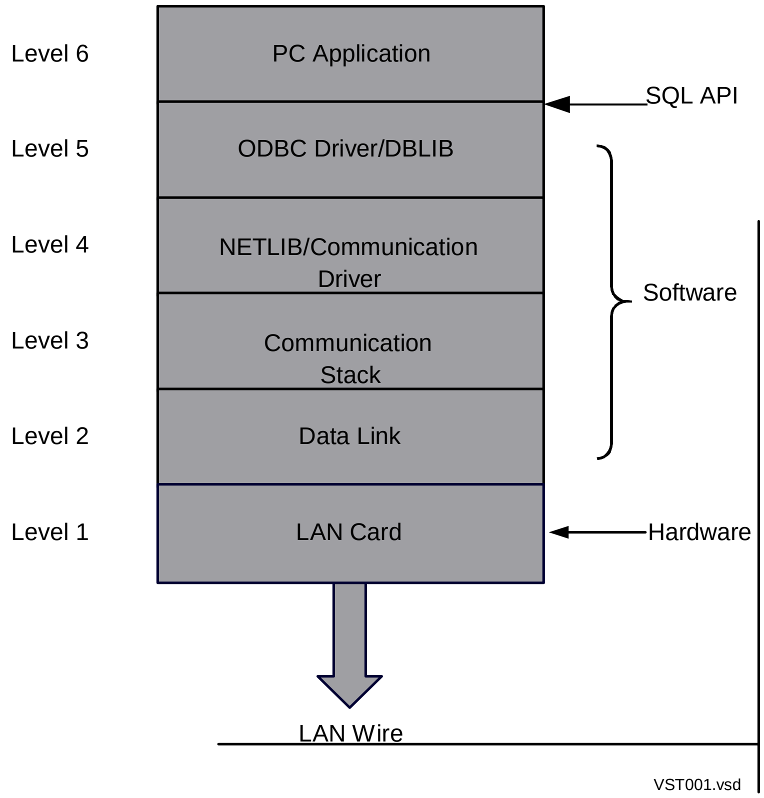
Clients on PCs can be either ODBC clients or DBLIB clients.

ODBC clients run on Microsoft Windows. DBLIB clients can run on DOS, Windows, OS/2, or UNIX platforms. For transport, any of these clients can use TCP/IP, as supplied by Microsoft Corporation, FTP, UB Networks (for PC/TCP Applications + BNS/NDIS), TCPPRO, or Novell LAN WorkPlace. The Winsock (version 1.*x*) interface is also supported.

A corresponding NETLIB, for DBLIB, or a communication driver, for the NonStop ODBC driver, is required. The communication drivers are bundled with the NonStop ODBC driver.

Ultimately, the stack of software and hardware on the PC looks something like that shown in Figure 2-1:

Figure 2-1. Software/Hardware Stack for a PC Client



This sub-section lists some of the options for each of the six levels and then discusses valid combinations. Everything described here applies to both the NonStop ODBC driver and DBLIB, unless otherwise noted.

Note. The components described here for the layers are the ones that have been tested to some extent. There are other products available—such as TCP/IP stacks from other vendors—that may also work well. If you want to use other components, be sure to install and configure them according to directions and perform the verification tests described in this section.

Level 1—LAN Card

The following options are available:

Supplier	Product Name
UB Networks	NIUpc/EOTP
Novell	NE2000
3COM	3C503

Note. UB Networks supplies an older card—NIUpc/3270NIUPC—that has worked in the past, but it is no longer recommended.

Level 2—Data Link

The following options are available:

Supplier	Product Name	Use with LAN Card
Novell	ODI	Novell
3COM	ODI or NDIS	3COM
UB Networks	NDIS	UB Networks

The data link software is provided by the LAN card vendor. Depending on the source of the 3COM LAN card, you can get ODI, or NDIS, or both.

Level 3—Communication Stack

The communication stack type can be TCP/IP. The following options are available:

Type	Supplier	Version	Product Name	Use with LAN Card	
TCP/IP	FTP	2.2 or 3.0	FTP PC/TCP	Novell	
TCP/IP	Novell	4.0	LAN Workplace for DOS	Novell, 3COM	(1)
TCP/IP	UB Networks	2.2	UB Networks PC/TCP Apps + BNS/NDIS	UB Networks	(2)
TCP/IP	Network TeleSystems	1.01	TCPPRO	Novell, 3COM, UB Networks	
TCP/IP	Wollongong	2.0	Pathway	Novell, 3COM, UB Networks	(3)
TCP/IP	NetManage	4.01	Chameleon	Novell, 3COM	(3)

Type	Supplier	Version	Product Name	Use with LAN Card
TCP/IP	Microsoft	3.11	TCP-32	Novell, 3COM, UB Networks (3, 4)
TCP/IP	Microsoft	2.1	LAN Manager	Novell, 3COM, UB Networks

(1) Ask for the LAN Workplace for DOS product. If you ask for the Novell stack, you will get IPX/SPX.

(2) This is a special version of the FTP 2.2 stack sold by UB Networks.

(3) This product is a Winsock-compliant stack.

(4) This product works with Windows for Workgroups.

Note. For Windows 95 and Windows NT, the NonStop ODBC Server also supports the Microsoft TCP/IP that comes with the operating system.

You can also run TTE or RSC on the UB Networks stack. If you already have TTE installed, you can verify connectivity to the NonStop system by starting a dynamic TACL session. If you do not have the UB Networks stack installed, you must also install LAN Manager. The major steps for doing so are summarized here, but you should refer to the product documentation.

1. Create the LAN Manager UB 16.6*x* disk by using the TCP BNS/NDIS disk that comes bundled with the UB Networks PC/TCP+Apps package. This step is dependent on the UB Networks LAN card being used.
2. Install LAN Manager version 2.1, and select the import driver option to install the UB Networks 16.6*x* driver.
3. Check that you can connect to the NonStop system using TTE by starting a dynamic TACL session.
4. Install other PC/TCP applications. They will bind with the NDIS from LAN Manager.
5. Check the TCP/IP stack by using “Ping *`NonStop IP-address`*” (the Ping utility is in the PC/TCP subdirectory).
6. Now you can install ODBC or RSC software.

Level 4—NETLIB or Communication Driver

For ODBC clients, HP supplies the communication drivers for all communication stacks with the NonStop ODBC Server product.

Level 5—ODBC Driver or DBLIB

HP supplies the ODBC driver with the NonStop ODBC Server.

The NonStop ODBC Server currently supports DBLIB versions 4.2.*x* through 4.9.*x*. Clients written to the Sybase System 10 Open Client interface can communicate with the NonStop ODBC Server provided they use only those features supported by Sybase 4.9.*x* SQL Server version.

Note. There is a large number of configuration combinations for the client. NonStop system does not validate all of the previously stated combinations or guarantee they will work under all conditions.

When you use one of the combinations listed here, you should follow all of the NonStop ODBC Server installation procedures that follow. In particular, after you have configured your LAN connections, you should verify that clients have LAN connectivity to the NonStop system before NonStop ODBC Server clients try to connect to NonStop ODBC servers.

Level 6—PC Application

Verify that the following conditions are met on each PC to be used for a client application.

ODBC Clients

For ODBC clients, check the following:

1. For Microsoft Windows, in addition to application needs, you need the following if you use the NonStop 32-bit driver:

   Windows 95, Windows 98, Windows 2000 Professional, Windows XP Professional, or NT 3.5*x* Server or Workstation

   At least 16 MB of memory

   At least 40 MB of available disk

2. For TCP/IP, verify this information:

   a. Check that the PC has the correct LAN card (for Microsoft, FTP, or Novell). Novell NE2000 and 3COM 3c503 are recommended.

   b. Follow vendor instructions for installing the communication software. (The communication drivers are supplied by HP and are installed when the NonStop ODBC driver is installed.)

3. Check that you have a file transfer capability, such as IXF or FTP, between the NonStop system and the PC.

4. Verify LAN connectivity between the NonStop system and the PC over the transport to be used. This can be done in a variety of ways, including the following:

   a. Verify that file transfer works (which also verifies the previous step).

   b. If you use TCP/IP, start a TELNET session.

   c. Start a dynamic TTE session.

Note. Verifying that the client has LAN connectivity to the NonStop system is one of the most important checks to be made before proceeding.

DBLIB Clients

For DBLIB clients, check the following:

- For Microsoft Windows, in addition to application needs, you must have:
  - Release 95 or later of Windows
  - At least 8 MB of memory
  - At least 40 MB of available disk
- For DOS, in addition to application needs, you must have:
  - Release 3.3 or later of DOS
  - At least 8 MB of memory
  - At least 40 MB of available disk
- For OS/2, in addition to application needs, you must have:
  - Release 2.1 or later of OS/2
  - At least 8 MB of memory
  - At least 70 MB of available disk
- For TCP/IP, make this verification:
  - Check that the PC has the correct LAN card (for Microsoft, FTP, or Novell).
  - Follow vendor instructions for installing the communication software.
- Verify that DBLIB is Release 4.2 or later (up to 4.9.*x*).
- Verify that the correct NETLIB is installed for DBLIB. There are separate NETLIBs for the TCP/IP variants and for Named Pipes. In all cases, follow vendor instructions for installing the NETLIB.
- Verify LAN connectivity between the NonStop system and the PC over the transport to be used. This check must be made before proceeding. Verification can be done in a variety of ways:
  - Verify that file transfer works (which also verifies the previous step).
  - If you use TCP/IP, start a TELNET session.
  - Use DBPing (supplied by Sybase or Microsoft).
  - Check the UNIX workstation clients.

Clients on UNIX workstations are DBLIB clients using TCP/IP. IPX/SPX is not supported for DBLIB clients in this release.

TCP/IP comes with each UNIX system, but in some cases you can purchase an improved version.

Verify that the following conditions are met on each UNIX workstation to be used for a client application:

- Check application and DBLIB requirements for memory and disk space. (The NonStop ODBC Server does not supply any software for UNIX systems).
- Make sure you have version 4.2 or later (up to 4.9.*x*) of DBLIB.
- Make sure you have a file transfer capability, such as IXF or FTP, between the NonStop system and the PC.
- Verify LAN connectivity between the NonStop system and the UNIX workstation. This can be done in a variety of ways:
  - Verify that file transfer works (this also verifies the previous step).
  - Use Ping to test physical IP connectivity.
  - Start a TELNET session.

Backing Up SQL/MP Data

When to Back Up

IAn initial installation of the NonStop ODBC Server can be undone, if necessary, with the Deinstall tool. Catalogs, tables, views, and indexes that were created during the install are dropped during the deinstall. In this case, backups of SQL/MP data are optional.

However, if you select the UPGRADE installation option to upgrade an existing version of the NonStop ODBC Server, the existing catalogs are changed. These changes cannot be undone by Deinstall. Therefore, you should back up the old system catalog before upgrading to the new software. See “Backing Up an Old NonStop ODBC Server System”.

Backing up an Old NonStop ODBC Server System

The procedure for backing up an older system is as follows:

1. Shut down the old NonStop ODBC Server system by stopping all associated processes. This step is recommended for prevention of locks and their delays, but is not mandatory.
2. Use the BACKUP utility to back up the old NonStop ODBC Server catalog tables and views in the system catalog. (This step backs up their indexes too.) Their names all begin with the uppercase letter Z, so you can use a command such as this example:

   ```
   BACKUP $TAPE, (*.*.ZN*,*.*.ZV*) FROM CATALOG $SYSTEM.SQL
   ```

3. Back up all the old NonStop ODBC Server SQL/MP catalog tables and views from the desired user catalogs. A BACKUP command such as the one in the preceding example works if there are no user table names starting with the letters ZN or ZV.

4. Back up all the old NonStop ODBC Server software and any associated configuration information (such as the SCS configuration files). These should be put into a single location, allowing them to be referenced by the NOSU200 utility.

   Note. It is generally not necessary to back up SQL/MP objects owned by users. The restoration process described in the next subsection resynchronizes the old catalogs with the SQL/MP objects. However, if the database is very small, it might be more convenient to back up everything.

5. If your old NonStop ODBC Server system supported stored procedures and you have stored procedures installed, there is usually nothing you need to do for a NonStop ODBC Server upgrade. If, however, you are backing up an entire NonStop ODBC Server system, including the Pathway systems used for stored procedures, there is one issue to consider: the stored procedure names registered with the NonStop ODBC Server.

   When you back up the NonStop ODBC Server catalogs, the registered names are included. The name of the Pathmon process for the Pathway system and the server class name are part of the information registered with the stored procedure name.

   If you later restore the system, you must verify this information. The utility statements VALIDATE and REFRESH do not include this information, so you must do it manually. If any of the information has changed (for example, the Pathway system is restored differently), you must use NOSUTIL to delete the stored procedure name and then reregister it.

After the backup procedure is complete, the NonStop ODBC Server can be installed according to the instructions under NonStop System Installation on page 2-15. If, after the NonStop ODBC Server is installed, you want to revert to the old NonStop ODBC Server, you must restore the older system you backed up.

Restoring an Old NonStop ODBC Server System

The restoration process is as follows:

1. Deinstall the new NonStop ODBC Server:

   a. Stop all SCS processes to prevent new connections and to stop all processes associated with the NonStop ODBC Server.

   b. Use NOSCOM to deinstall the system catalog and all user catalogs. Note that any SQL/MP objects created through the NonStop ODBC Server are also dropped.

   c. Remove all NonStop ODBC Server software.

2. Restore the software for the old NonStop ODBC Server, including any saved configuration information.
3. Restore the system catalog tables and views.
4. Restore the user catalog tables and views (if you want, you can restore only a subset of the user catalogs that were backed up).
5. If the system you are restoring supported stored procedures and you had stored procedures installed at backup time, you must ensure that the information is correct in the NonStop ODBC Server catalogs regarding what Pathway system and server classes to use (see the discussion in Step 5 of the backup procedure on page 2-14).
6. Run the old NonStop ODBC Server VALIDATE utility on the system catalog and on the restored user catalogs. It reports discrepancies such as missing user SQL/MP tables or user catalogs that are not customized. Inspect the report and adjust the SQL/MP tables and SQL/MP catalogs as necessary (do not alter any of the old NonStop ODBC Server catalog objects).
7. Run the old NonStop ODBC Server REFRESH utility on the system catalog and on user catalogs to make necessary changes. This completes the restoration.

Backups for Clients

The NonStop ODBC Server includes a NonStop ODBC installation tool that can install the new NonStop ODBC driver and configure it. If there is an existing NonStop ODBC driver of an earlier version, the Install tool erases it without prompting the user. If you want to save the old version of the NonStop ODBC driver, you should do so before running the Install tool.

NonStop System Installation

Installation on the NonStop system involves

- installing the software
- configuring the NonStop ODBC Server system
- starting the system
- verifying the installation

Configuration is done at several places in the process. Verification of each step is very important, as each step builds on the previous steps.

This subsection lists the steps for installing with a minimum of configuration; at points in the procedure where additional, optional configuration can be done, notes are given to that effect.

There are four important subvolumes with which you will be working:

- NonStop ODBC Server installation source subvolume

  Most NonStop ODBC Server files are placed here. The subvolume name is ZNOS.

Nomenclature	*NOS-Source-Disk*.ZNOS
Example	$ISV.ZNOS

- NonStop ODBC Server system object subvolume

  Most NonStop ODBC Server files are placed here. The subvolume name is ZNOS.

Nomenclature	*NOS-Target-Disk*.ZNOS
Example	$VOL2.ZNOS

- SQL/MP system catalog

  This subvolume contains the SQL/MP system catalog. The NonStop ODBC Server system catalog is also placed here.

Nomenclature	*SQL-System-Catalog*
Example	$SYSTEM.SQL

- System object subvolume

  This subvolume contains system object files.

Nomenclature	*System-Object-Subvolume*
Example	$SYSTEM.SYSTEM

Installation Steps

This subsection describes the step-by-step procedure for installing the NonStop ODBC Server on the NonStop system. Read through all the installation steps to make sure you understand the process before proceeding.

Step 1—Prepare

Read the softdocs included with the NonStop ODBC Server. Information in the softdoc supersedes information in this manual.

Step 2—Check the System

Make sure you have verified all NonStop system requirements and performed needed backups. Also, verify LAN connectivity between clients and the NonStop system.

Step 3—Back up the Old NonStop ODBC Server System

If you have a previous release of NonStop ODBC Server installed, you should back up the system at this time. For more information, see Backing up an Old NonStop ODBC Server System on page 2-13.

Note. If you have an old version of the NonStop ODBC Server installed, you do not need to deinstall it.

Step 4—Log On

Log on as a super-group user (group ID 255). Your TACL process *must* be named; if it is not, use the TACL command that includes the NAME run-option to start a named TACL process before logging on:

```
TACL /NAME/
001> LOGON SUPER.user-name
```

The installation process requires that you be logged on as the super ID. The NOSINST process checks for the super ID and reports an error if the requesting user is not the super ID.

After installation is complete, you can modify NOSUTIL for PROGID so that the NOSUTIL file owner name is a super-group user. If this is done, the Guardian user name under which you log on need not be a super-group user. For more information, see the softdoc that accompanies the NonStop ODBC Server product.

Step 5—Run INSTALL

Run INSTALL. The files are restored from their respective distributed release subvolumes to the subvolume *NOS-Target-Disk.ZNOS*.

Step 6—Verify Installation

The NOSINST macro verifies that the correct files were installed. The list is subject to change for subsequent releases, and is therefore not reported here. NOSINST contains the correct list for a specific release.

Step 7—Prepare to Run NOSINST

Use the NOSINST macro to install the NonStop ODBC Server software and make it ready to run. You must first move to the subvolume containing the NOSINST macro and the NonStop ODBC software. This subvolume was created by DSM INSTALL for software delivered by SUT, or IPUTIL for software delivered by CD.

Select a subvolume into which the NonStop SQL executables will be installed:

```
    VOLUME NOS-Source-Disk.ZNOS
```

Step 8—Run NOSINST

Run the NOSINST macro.

The TACL macro NOSINST performs these parts of the installation for you:

- Installs the NOSUTIL utility program and its interface, NOSCOM, on subvolume *`NOS-Target-Disk.ZNOS`* by copying the release file ZNOSUTIL and renaming it as NOSUTIL, copying ZNOSCOM and renaming it as NOSCOM, and then SQL-compiling NOSUTIL.
- Installs the NonStop ODBC server program on *`NOS-Target-Disk.ZNOS`* by copying the release file ZNOS and renaming as NOS, then SQL-compiling NOS and licensing it.
- Installs the SCSOBJ monitor program on *`NOS-Target-Disk.ZNOS`* by copying it and installs the ancillary SCS files by copying them.
- Installs the NOS Error file on subvolume *`NOS-Target-Disk.ZNOS`* by copying the release file ZODBCMSG and renaming as ODBCMSG.
- Builds the NonStop ODBC Server system catalog, which creates a collection of SQL/MP tables, views, and indexes, and populates the tables with NonStop ODBC metadata. If you are upgrading from a previous release of the NonStop ODBC Server, NOSINST upgrades the NonStop ODBC Server system catalog tables to the current version.

NOSINST returns messages as it completes each intermediate step. Any fatal errors result in termination of the installation script. With the exception of catalog creation and NonStop ODBC system catalog creation, no automatic backout procedure is performed.

Help is provided for the basic commands. Enter this command to get a summary of options:

```
RUN NOSINST HELP
```

The ALL and UPGRADE installation options are collections of steps that can be executed in sequence directly by the user. This strategy lets you correct a detected problem and resume the installation without repeating completed steps. For more information about NOSINST options, see NOSINST on page 2-58.

There are three NOSINST installation sequences. The sequence you choose depends on whether or not you currently have an older NonStop ODBC Server version installed, and whether or not you want to overlay the older release. Choose one of the following:

Option 8a—Run NOSINST for a New Installation

These instructions are for installing NonStop ODBC Release 2 onto a system where no prior releases of NSODBC have been installed.

1. Run NOSINST.
2. For Installation Option, specify ALL (A) to perform the compile, install, and load steps.
3. Provide the location of the SQL/MP system catalog if not $SYSTEM.SQL.
4. Provide the location of the NonStop ODBC system catalog if not the same as the SQL system catalogs.
5. Provide the location of the target NonStop ODBC system software subvolume if needed.
6. Load the default Administrator Configuration.
7. Copy the RUNME file to the client workstation and install the client software as described in Installing the Client on the PC on page 2-27.

Option 8b—Run NOSINST for a Concurrent Installation

These instructions are for installing NSODBC Release 2 onto a system where NonStop ODBC Release 1 is currently installed, and you want to leave Release 1 in place and create an additional NonStop ODBC Release 2 system. These instructions should also be used if you need to move the NonStop ODBC system catalog to a location other than the SQL/MP system catalog.

1. Run NOSINST.
2. Choose a NonStop ODBC Release 2 system catalog in a location different from the SQL/MP system catalog.You must set the DEFINE =_NSODBC_SYSTEM_CATALOG before using NOSCOM to start an NonStop ODBC system. Once install is complete, you may place this value in a TACLCSTM file or in a macro encapsulating startup.

   ```
   ADD DEFINE =_NSODBC_SYSTEM_CATALOG, CLASS CATALOG, SUBVOL
   $newvol.SQL
   ```

3. For Installation Option, specify ALL (A).
4. Change the location of the SQL/MP system catalog to the value of the define in Step 2 on page 2-19.
5. Change the location of the NonStop ODBC system catalog to the new subvolume.
6. Change the location of the target NonStop ODBC system software subvolume if needed.
7. Complete the install:
   a. Load the default Administrator Configuration.
   b. Run SYSCAT REFRESH against the Release 1 NonStop ODBC system catalog using the Release 1 NOSUTIL:

      ```
      RUN NOS-Release1-subvol.NOSUTIL SYSCAT REFRESH
      ```

c. Run the migration utility, NOSU200, to extract NonStop ODBC Release 1 configuration data and transform it into a NonStop ODBC Release 2 configuration ADD script. Use the ALL option and specify the name of one of the SCS configuration files to be used:

```
RUN NOSU200 EXTRACT ALL $system.sql addcmd scscfgt1 &
  $d.znosr2
```

d. Run the SCS option for each additional SCS configuration file. The process will append to the original file created in the first step. Check the generated NOSCOM commands and correct as needed:

```
RUN NOSU200 EXTRACT SCS $system.sql addcmd scscfgt2
$dianosr2
```

e. Install the NonStop ODBC Release 1 configuration by running NOSCOM with the file/files generated from Step 3:

```
RUN NOSCOM /in addcmd/
```

f. Install client software. If you will use the Release 2 NonStop ODBC Windows driver with an existing Release 1 system, create view ZNUDT in the Release 1 system catalog:

```
CREATE VIEW $SYSTEM.SQL.ZNUDT AS SELECT * FROM
$SYSTEM.SQL.ZNUDT
```

g. If you will use an existing NonStop ODBC Release 1 driver with this newly installed NonStop ODBC Release 2 system, create view ZNSDT in the Release 2 system catalog:

```
CREATE VIEW $R2.SQL.ZNSDT AS SELECT * FROM $R2.SQL.ZNUDT
```

Option 8c—Run NOSINST for a Upgrade Installation

These instructions are for installing NonStop ODBC Release 2 onto a system where NonStop ODBC Release 1 is currently installed, and you want to replace Release 1 with NonStop ODBC Release 2.

△ **Caution.** An upgrade installation overwrites any existing catalog contents.

1. Run SYSCAT REFRESH against the Release 1 NonStop ODBC system catalog by using the Release 1 NOSUTIL:

```
RUN NOS-Release1-subvol.NOSUTIL SYSCAT REFRESH
```

2. Run the migration utility, NOSU200, to extract Release 1 configuration data and transform it into a NonStop ODBC Release 2 configuration ADD script. Use the ALL option and specify the name of one of the SCS configuration files to be used. Run the SCS option for each additional SCS configuration file:

```
RUN NOSU200 EXTRACT ALL $system.sql addcmd scscfgt1 &
  $d.znosr2
```

Check the generated NOSCOM commands and correct as needed.

3. Run NOSINST.

    a. For Installation Option, specify UPGRADE (U) to perform the compile, install and load steps.

    b. Change the location of the SQL/MP system catalog if other than $SYSTEM.SQL.

    c. Change the location of the NonStop ODBC system catalog to the SQL/MP system catalog.

    d. Load the default Administrator Configuration.

4. Install the configuration from NonStop ODBC Release 1 by running NOSCOM with the file / files generated from the previous step.

```
RUN NOSCOM /in addcmd/
```

5. Install client software. If you will use the Release 2 NonStop ODBC Windows driver with a Release 1 system, create view ZNUDT in the Release 1 system catalog:

```
CREATE VIEW $SYSTEM.SQL.ZNUDT AS SELECT * FROM
$SYSTEM.SQL.ZNUDT
```

6. If you will use the Release 1 NonStop ODBC driver with a Release 1 system, create view ZNSDT in the Release 2 system catalog:

```
CREATE VIEW $R2.SQL.ZNSDT AS SELECT * FROM $R2.SQL.ZNUDT
```

Verification of Installation

NOSINST prints messages reporting its progress as it executes its various steps. If it encounters an error, it prints an error message and terminates. NOSINST will also advise you if it finishes without errors.

You can use the TACL command FILEINFO *`file-name`* to see the timestamp of a file and whether it is licensed. You can use the FUP command INFO *`file-name`*, DETAIL to see if a file has been SQL-compiled, if it is a SQL/MP table and, if it is, in what SQL/MP catalog it is registered.

If NOSINST does not report errors, make the following checks on the indicated subvolumes.

- Check that the following files now exist on `NOS-Target-Disk`.ZNOS:

NOSCOM	A type 100 file
NOSUTIL	A type 100 file, SQL-compiled, and registered in the SQL/MP system catalog
NOS	A type 100 file, SQL-compiled, registered in the SQL/MP system catalog
SCSOBJ	A type 100 file

- After you have run NOSINST, check that the following NonStop ODBC Server system catalog tables (SQL/MP tables and views) now exist on SQL-System-Catalog. All are registered in the SQL/MP system catalog.

  The names of all the NonStop ODBC Server catalog objects begin with the following letters:

  ZNS NonStop ODBC Server tables and indexes in the NonStop ODBC Server system catalog

  ZNU NonStop ODBC Server tables and indexes in NonStop ODBC Server user catalogs

  ZVS NonStop ODBC Server views in the NonStop ODBC Server system catalog

  ZVU NonStop ODBC Server views in NonStop ODBC Server user catalogs

  All objects in a user catalog also exist in the system catalog.

  If you have access to SQLCI, you can check all tables and views in the NonStop ODBC Server system catalog by running the following query:

```
>> SELECT TABLENAME,TABLETYPE,CONVERTTIMESTAMP(CREATETIME)
+> FROM SQL-System-Catalog.TABLES
+> WHERE TABLENAME LIKE "%.ZN%" OR
+>       TABLENAME LIKE "%.ZV%";
```

  You can check all the indexes in the NonStop ODBC Server system catalog by running the following query:

```
>>SELECT INDEXNAME,CONVERTTIMESTAMP(CREATETIME)
+>FROM $SYSTEM.SQL.INDEXES
+>WHERE INDEXNAME LIKE "%.ZN%"
+>  AND TABLENAME <> INDEXNAME;
```

  Shown following is a sample output of the SQL/MP SELECT statement on TABLES listing all the tables and views in the NonStop ODBC Server system catalog (see “NonStop ODBC Server Mapping Tables,” in the *NonStop ODBC Server Reference Manual* for definitions of the objects and their use).

```
TABLENAME                         TABLETYPE  (EXPR)
--------------------------------  ---------  --------------------------

\TESS.$SYSTEM.SQL.ZNSALT          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSCON          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSDB           TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSDEF          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSDUMMY        TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSGOV          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSMSG          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSNET          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSPROF         TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSPROT         TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSSCFG         TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSSCS          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSSER          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSSMAP         TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSTRA          TA         1996-10-01:11:01:01.123456
```

```
\TESS.$SYSTEM.SQL.ZNSUMAP          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSUS            TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNSVALUE         TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNUDT            TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNUIX            TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNUOBJ           TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNUPCOL          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZNUPROC          TA         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZVSDB            VI         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZVSMSG           VI         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZVUCOL           VI         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZVUDT            VI         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZVUIX            VI         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZVUOBJ           VI         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZVUOCOL          VI         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZVUPCOL          VI         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZVUPROT          VI         1996-10-01:11:01:01.123456
\TESS.$SYSTEM.SQL.ZVUUS            VI         1996-10-01:11:01:01.123456
```

Shown following is a sample output of the SELECT statement showing all the indexes in the NonStop ODBC Server system catalog.

```
INDEXNAME                          (EXPR)
---------------------------------  -------------------------

\TESS.$SYSTEM.SQL.ZNSALTI1         1996-10-01:21:08:34.074567
\TESS.$SYSTEM.SQL.ZNSDBI1          1996-10-01:21:08:27.931886
\TESS.$SYSTEM.SQL.ZNSDBI2          1996-10-01:00:34:28.456680
\TESS.$SYSTEM.SQL.ZNSUSI1          1996-10-01:21:08:49.972288
\TESS.$SYSTEM.SQL.ZNSUSI2          1996-10-01:21:08:52.273462
\TESS.$SYSTEM.SQL.ZNUIXI1          1996-10-01:21:09:20.028462
\TESS.$SYSTEM.SQL.ZNUOBJI1         1996-10-01:21:08:59.787055
\TESS.$SYSTEM.SQL.ZNUOBJI2         1996-10-01:21:09:02.035261
\TESS.$SYSTEM.SQL.ZNUOBJI3         1996-10-01:21:09:07.657557
\TESS.$SYSTEM.SQL.ZNUPCLI1         1996-10-01:00:35:42.386361
```

- As the final part of verification, run NOSCOM with the SYSCAT VALIDATE option to check values in the rows of the catalog tables:

  ```
  RUN NOS-Target-Disk.ZNOS.NOSCOM SYSCAT VALIDATE
  ```

At this point, the system catalog is customized, and at least one logical username, “DBO” is created. DBO is configured to use the Guardian user name super ID, the DEFAULT server class, and the DEFAULT profile. An additional alias exists if the default administration configuration, “NSOADMIN,” was loaded. This configuration is sufficient to bring up SCS and have a client connect. The Windows-based application configuration tool installed subsequently is the recommended interface for completing your configuration.

Because you install the NonStop ODBC Server while logged on as a super-group user, the SQL/MP tables created for you are owned by that super-group user. The security of the catalog tables is set to “NNNO” for the tables that can be updated and “NOOO” for the tables that do not change.

When installation is complete, you might want to alter ownership and security settings for some NonStop ODBC Server objects. You might also alter ownership and security settings for SQL/MP objects that the NonStop ODBC Server uses, such as SQL/MP catalogs. If you change SQL/MP table ownership, you should run NOSCOM SYSCAT REFRESH before proceeding.

See Section 4, Managing the NonStop ODBC Server, for a discussion of ownership and security.

Step 9—Configure the SQL Communication Subsystem (SCS)

SCS configuration is driven from data stored in NonStop ODBC Server system catalog tables. These tables contain descriptions to configure the SCS process, the network protocol and service, the NonStop ODBC Server processes, and the server classes and the associations among them.

You start an SCS process by using the following utility statement:

```
NOSCOM> START SCS scs-name
```

The process of installing the NonStop ODBC Server optionally loads a minimal configuration necessary to start an SCS, connect through the Configuration Manager, and proceed with manipulating the configuration attributes of the target system.

A configuration can be verified by using the utility statement INFO CONFIG. The following example shows the configuration as it is established during the installation process:

```
NOSCOM> info config $nsoad

INTERVAL CHECK (SECS): 60
NOS OBJECT FILE: \N1.$ODBC.ZNOSR2.NOS
NOSUTIL OBJECT FILE: \N1.$ODBC.ZNOSR2.NOSUTIL
NOSUTIL PRIORITY: -1
NOSUTIL CPU: -1
INPUT BUFFER SIZE (BYTES): 2448
OUTPUT BUFFER SIZE (BYTES): 2448
EMIT EVENTS: Y
PIPE TEST: N
MEMORY CHECK: N
SYSTEM CATALOG: \N1.$D1.ODBC2SYS
SCS LAST UPDATED: 29Apr96  8:22
SERVERS CONFIGURED: 3

NETNAME: ODBC2000
--------
PROTOCOL: TCP/IP
IOP: \TESS.$ZTC0
SO_KEEPALIVE: 1
SO_REUSEADDR: 1
NET LAST UPDATED: 25Apr96  9:11

SERVER NAME: NSOADMIN
------------
CPU LIST: -1
USE AS DEFAULT: Y
```

```
SHUTDOWN SERVER: N
PRIORITY: 140
PROFILE: NSOADMINPROF
G_USER: SUPER.ODBC
AVAILABLE_SERVERS: 2
MAX_SERVERS: 5
IDLE DELETE DELAY (SECS): 60
NOS RUN OPTIONS:
INITIAL HEAP SIZE (KBYTES): 750
MAX HEAP SIZE (KBYTES): 750
SERVER LAST UPDATED: 25Apr96 13:11
TIMEOUT SECONDS: 0
```

This INFO CONFIG statement is the same statement used by the SCS when NOSUTIL must return its configuration by name.

Step 10—Start the SQL Communication Subsystem

The procedure for starting the SCS is as follows:

1. Set the DEFINE for the location of the NonStop ODBC Server system catalog. Use the ADD DEFINE command at the TACL prompt prior to starting NOSCOM.

2. If you need start-up messages, start the RUNFILTR macro supplied with the NonStop ODBC Server. It prints system console messages generated by SCS and is useful both for verifying that SCS has started and for troubleshooting (see Troubleshooting on page 2-58 for further details).

3. You must run RUNFILTR from *NOS-Target-Disk* and on a separate terminal or in a separate window. From a separate TACL prompt, type:

   ```
   RUN RUNFILTR
   ```

4. Log on as a super-group user with sufficient authority to start a licensed process under a different Guardian user name. To ensure having authority greater than or equal to any user who may connect, you should log on as the super ID.

5. The SQL Communication Subsystem runs under this Guardian user name. NonStop ODBC server processes associated with this SCS process are started under this name, but then assume the identity of the Guardian user name indicated for the server class definition.

6. To start SCS, issue the following command from a TACL prompt:

   ```
   RUN NOSCOM START SCS scs-name
   ```

   where *scs-name* is the identifier for an SCS configuration.

Verification

You can verify a successful SCS start in various ways.

- If you are using NonStop ODBC Release 1, the simplest check is to use TDBSRV:STATUS at the TACL prompt. This command lists all of the processes started by the SCS server. The syntax for this command is:

  ```
  TDBSRV:STATUS SCS-config-file
  ```

  where *`SCS-config-file`* is the name of the SCS configuration file used to bring up the system.

  Here is sample output from the STATUS command:

  ```
  NOSCOM> status scs $nsoad

  SCS_NAME: \N1.$NSOAD
  JOB_ID: 1
  PROGRAM_FILE: \N1.$D1.ODBCSYS.SCSOBJ
  SWAPVOL: \N1.$SWAP.#0116894
  HOMETERM: \N1.$OSP
  PRIORITY: 140
  USERID: 255,255
  PARENT USERID: 255,255
  PROCESS TIME: 271808
  ```

- If you are using NonStop ODBC Release 2, use the NOSCOM STATUS command.The syntax is:

  ```
  NOSCOM STATUS SCS-name
  ```

- You can use the various SCF commands described in Section 4, Managing the NonStop ODBC Server, to check on the status of SCS and the associated NonStop ODBC servers.

- If you started RUNFILTR, check the messages it prints to ensure that SCS has started. This is a sample of RUNFILTR output when an SCS process is started and it prestarts one NonStop ODBC server (to improve readability, the output records have been reformatted slightly from what you see on the screen):

  ```
  94-11-17 08:59:21 \TESS.$ZT1       Tandem.SCS.D42
    000007 $ZDSS.1: process $ZT1 object state changed to started

  94-11-17 08:59:22 \TESS.$ZT1      Tandem.SCS.D42
    000007 $ZDSS.1: session @Listener object state changed to starting

  94-11-17 08:59:24 \TESS.$ZT1       Tandem.SCS.D42
    -32750 \TESS.$ZT1: Process started (T9245D30 26OCT94)

  94-11-17 08:59:46 \TESS.$ZT1       Tandem.SCS.D42
    000007 $ZDSS.1: server \TESS.$X716 object state changed to defined

  94-11-17 08:59:46 \TESS.$ZT1       Tandem.SCS.D42
    000007 $ZDSS.1: server \TESS.$X716 object state changed to starting
  ```

  This example shows five trace records. The SCS process is named $ZDS1 (set by process-name in the SCS section of the SCSCFG* configuration file). The first three records detail the starting of the SCS process and the last two the starting of an NonStop ODBC server process named $X716.

- Run NOSCOM and execute the utility statement STATUS SCS *`scs-name`*.

The SQL Communication Subsystem is now ready to accept connections from clients. Proceed to Installing the Client on the PC on page 2-27 to see how to configure and start your clients.

If you have any difficulty, see Troubleshooting on page 2-58.

Stopping SCS or NonStop ODBC Server Processes

To stop an SCS process, enter:

```
STOP SCS scs-name [NOWAIT]
```

The NOWAIT parameter specifies that the SCS process is to stop immediately. Without that parameter, the SCS process continues with its current sessions, but on dynamic configuration boundaries, it stops accepting new login requests and shuts down each NonStop ODBC server process as it completes its current user session.

You can also stop NonStop ODBC server processes by using the TACL command STOP. You might want to do this if the processes are not responding or are processing a very large SQL/MP statement and you do not want to wait for it to finish.

You do not need to stop any other NonStop ODBC Server processes. Private NOSUTIL processes are used by SCS and by some NonStop ODBC servers, but they are automatically managed. You never need to stop SQL/MP SQLCOMP or SQLCAT processes started by a NonStop ODBC server process. To stop the SQL operation, stop the NonStop ODBC server process.

When you stop an SCS process, messages similar to these appear in the EMS log for each NonStop ODBC server process that was running:

```
95-01-27 13:54:27 \TESS.1,70  Tandem.EMS.D42     000512 T8666Dxx NonStop ODBC
Server
95-01-27 13:54:27 \TESS.1,70  Tandem.EMS.D42     000512 Time : 1995-1-27 13:54:27
95-01-27 13:54:27 \TESS.1,70  Tandem.EMS.D42     000512 ODBC Server message
(32312 : Error
                              233 connecting to SCS.) :
95-01-27 13:54:27 \TESS.1,70  Tandem.EMS.D42     000512 T8666Dxx NonStop ODBC
Server
95-01-27 13:54:27 \TESS.1,70  Tandem.EMS.D42     000512 Server Name : $X285
                              284  1995-1-27 13:54:27
95-01-27 13:54:27 \TESS.1,70  Tandem.EMS.D42     000512 ODBC Server message
(32310 : Error 3
                              opening SCS. Server terminating.) :
```

These messages are generated by the NonStop ODBC Server because it detects that the SCS process has terminated, and attempts to reconnect.

Installing the Client on the PC

This subsection describes the procedures necessary for installing the client portion of the NonStop ODBC Server product on a PC.

ODBC Installation

This subsection describes how to install the client portion of the HP NonStop ODBC Server product on a PC running Microsoft Windows when the client will be using ODBC.

The client portion of the NonStop ODBC Server for ODBC includes these pieces:

- HP ODBC SETUP, an easy-to-use Windows application that installs all NonStop ODBC Server components for ODBC, makes numerous checks, and verifies the installation. You use SETUP for the initial installation and for installing upgrades or other components later. The NonStop ODBC SETUP application provides a stream of information and question boxes as it installs. At each step, it suggests defaults and guides you to the next step.
- HP Connectivity Tool, a Windows application that uses the HP ODBC driver to perform a number of connectivity tests with the NonStop system. You use the Connectivity Tool to verify connectivity and to measure connect and messaging times.
- HP Debug Tool, a Windows application that provides traces of events in the HP ODBC driver. You can trace the ODBC calls made by the application, the SQL statements sent to the NonStop system, and the network messages.
- HP ODBC driver, which runs under ODBC and provides connectivity to the NonStop ODBC Server components on the NonStop system.
- Communication drivers for the HP ODBC driver. All the communication drivers for supported transports are provided.
- A RUNME file from *`NOS-Target-Disk.ZNOS`* that contains all the needed PC files, including the HP ODBC SETUP application, the HP Connectivity Tool, the HP ODBC driver, and the communication drivers. RUNME is a compressed, self-extracting file. These files support installation on 16-bit platforms, Win 3.1 and Win 3.11, as well as 32-bit platforms, Win 95 and Win NT 3.51 and 4.0.
- HP Configuration Tool, in both 16-bit and 32-bit versions, which is an application used to perform configuration management for the HP ODBC Release 2 system.

In addition to the HP components, Microsoft Corporation provides the following tools with ODBC:

- Microsoft ODBC Administrator, a Windows application that helps you install and configure ODBC drivers (including the definition of datasources).
- Microsoft ODBC Test, a Windows application that provides a conversational interface to ODBC. You can issue single ODBC calls to connect and disconnect, set parameter values, execute SQL, and obtain results.

Consult the Microsoft ODBC documentation to learn more about these tools. The ODBC documentation also explains more about installing, configuring, and using ODBC drivers.

Installation Procedure

The following installation steps must be repeated for each PC client:

1. Read the softdoc included with the NonStop ODBC Server product.
2. Make sure you have checked the PC software and hardware. In addition, be sure to test connectivity to the NonStop host before continuing with activities described in this procedure.
3. If you are installing from a CD-ROM, double click on the SETUP program in the \WS_SW directory and follow the SETUP instructions to install the client software.
4. If you are installing from a NonStop system, download the RUNME file.

   The RUNME file is a self-extracting zip file. It can be run on Windows NT, Windows 95, Windows 98, Windows 2000, or Windows XP. Downloading can be done a variety of ways, such as with IXF, FTP, or COPY (if LAN Manager is installed). Regardless of the method used, create a directory (for example, C:\SETUP) and copy the RUNME file from the NonStop system to that directory as RUNME.EXE.

   Note. Make sure that the directory used for setup is different from the directory where the NonStop software is installed (usually C:\ODBC). After installation is complete, the setup directory can be deleted.

   a. Download the RUNME file from *NOS-Source-Disk.ZNOS* on the NonStop system to the PC.

   b. Rename the file RUNME.EXE, if necessary, then run the RUNME program from either a DOS prompt or from the FILE Manager RUN option. It will “unzip” itself and produce a set of disk images from which the installation will be performed. This step produces an executable file called SETUP.EXE in the subdirectory DISK1.

      Note. When SETUP is running and begins to install a file, it checks to see if the file already exists. If so, it checks the version. If the existing file has an older version, SETUP overwrites it without notice and does not save the old version. If you want to save old versions of any of the applications or of the NonStop ODBC driver, do this before you run SETUP.

   c. Run the SETUP program using the RUN command from either the Windows program or the File Manager. Only one instance of SETUP.EXE can run on the PC at a given time.

      The file README.WRI is a WRITE document describing the use of the HP ODBC SETUP program. Print this file for reference. The SETUP.EXE program does not use the README.WRI file. You can manually copy it elsewhere.

To complete the installation, follow the instructions given by the HP ODBC SETUP program as it runs. You should accept the defaults unless otherwise noted. All prompts give you a chance to exit from SETUP. If you encounter an error, you receive a message box with a description of the error. In that case, you should exit from SETUP, fix the problem, and run SETUP again.

The following steps are performed by the HP ODBC SETUP program:

1. SETUP displays a welcome screen. Click Continue.
2. SETUP displays a menu of installation options from which to choose.

   All options are selected by default. Click Continue.
3. Next, SETUP displays a box to select the directory to use. If you have a “C” disk drive on your PC, click Continue to use the default (C:\ODBC). Otherwise, set the disk name and click Continue.
4. SETUP now installs several files. During the process, it shows a “percent done” bar. If any failures occur at this point, they are typically “out of space” or “access violation” conditions.

   If SETUP finds it is installing an older version of a file than is currently installed, it prompts you for a decision about what to do (the default is not to install the older version).
5. If the HP ODBC driver was installed, SETUP prompts you to configure it now or to do it later with the ODBC Administrator (the default). Click Continue.
6. SETUP builds a Windows program group called “HP NonStop ODBC...” with program items for the following (if they were chosen to be installed):

Component	**16-bit Installation**	**32-bit Installation**
Microsoft ODBC Administrator	X	X
Microsoft ODBC Spy		X
Microsoft ODBC Test	X	X
Connectivity Tool	X	X
Configuration Manager	X	X
Debug Tool	X	X

Finally, SETUP displays a “setup completed” screen with a summary of the version installed.

HP ODBC SETUP keeps a log of the changes it makes, as well as detailed error messages. The log is stored in a file named SETUP.LOG in the setup directory. It is useful for debugging.

7. Run the Microsoft ODBC Administrator (you can select this tool's icon from the HP NonStop SQL Program Group). Select the option to add a datasource and select the SQL driver. You will see a screen prompting for the following input:

   Data Source Name — The name the client will specify in the SQLConnect statement. "NonStop ODBC" or the same name as the Network Service name is suggested.

   Description — An optional comment: for example "NonStop ODBC Server database on *node*.$*vol*.*subvol*."

   Protocol — The network protocol to use. There is only one choice, NonStop WinSockets.

   Login Timeout — The number of seconds that the HP ODBC Driver waits for a connection to be completed. 0 tells the driver to wait indefinitely.

   Host Name — Input depends on the choice of protocol. For TCP/IP, specify the IP address corresponding to the installed NonStop system.

   Port — Chose the port number configured for SCS datasource. In this example, it is 2314 for TCP/IP. For TCP/IP, other site specific conventions can influence the choice.

   Database — The name of the database to be used for object qualification and object abstractions.

   Access Mode — Acceptable values are: SQL_MODE_READ_WRITE, SQL_MODE_READ_ONLY or SQL_MODE_NULL. See Section 4, Managing the NonStop ODBC Server for more details.

   Cursor Default Mode — Acceptable values are: FOR_UPDATE, READ_ONLY or READ_ONLY_STREAMING.

   Transaction Isolation — Acceptable values are:
   SQL_TXN_READ_UNCOMMITTED,
   SQL_TXN_READ_COMMITTED,
   SQL_TXN_REPEATABLE_READ or
   SQL_TXN_SERIALIZABLE

You can use the Microsoft ODBC Administrator tool to add new datasources, modify existing datasources, or delete existing datasources at any time.

1. Run the HP Connectivity Tool; you can select its icon from the SQL Program Group. Select the System Information Window to view versions of many components and available system resources. Click in this window at any time to cause the information to be updated.

   You may also view a Connection Window. You can create additional Connection Windows by choosing File | Connect from the Connection Menu.

A Connection Window is associated with one datasource; you should use it as follows:

a. If the Connection Window is not current, click within its top bar. Specify the datasource type as ODBC by selecting the appropriate radio button at the top right. Use the pull-down menu to choose among the datasource names that you configured with the ODBC Administrator tool in Step 7 on page 2-31. Choose a datasource name corresponding to the SCS you started on the NonStop system.

b. After you have chosen a datasource, click the CONNECT button to the right. A pop-up window prompts you for your username and password. Supply them and click OK.

   The Connectivity Tool then initiates a connection to the NonStop system and waits up to 15 seconds for a response. An information bar just below provides status and a hand shake icon appears to the left if the connection is successful.

   Note. If you get a successful connection, you know the HP ODBC driver is installed, the network transport is working, SCS and the NonStop ODBC server on the NonStop system are running, and you have configured the connection correctly.

   Whenever you are having trouble with an application connecting, repeat Steps 2.a and b as a test of basic connectivity.

2. After you are connected, you can use FILE | DISCONNECT to break the connection. There is a “Statistics” area. In this area are three buttons marked Connect, Echo, and Query, which correspond to experiments that you can do.

   You can specify that an experiment is to be run *n* times by setting the number of iterations (*n* defaults to 10). Upon completion, the average response time is shown and you will see a graph of the *n* samples. The graph is useful to get an idea of the variability of the response time.

The experiments you can do are as follows:

CONNECT The tool repeatedly connects to and disconnects from the datasource you have specified (using the same username and password each time). This experiment tells you how much time is needed just to connect to a NonStop ODBC server. If you have configured SCS server classes so that NonStop ODBC servers are prestarted, the connect time experiment tells you what the network latency is.

ECHO Using one connection, the tool repeatedly sends an “echo” command to the NonStop ODBC Server. The NonStop ODBC server immediately replies to the echo (it does not issue any SQL/MP activity nor does it return any data). This experiment tells you the minimum time needed for a round trip message from the client to the NonStop ODBC server and back. Therefore, it helps you understand how much of your wait time is for network latency.

QUERY Using one connection, the tool repeatedly executes the SQL statement you enter in the area at the bottom. If you execute a SQL SELECT statement, the tool does not fetch any of the result rows but merely issues an ODBC Execute Direct call and then closes the cursor by executing an ODBC Free Statement call. The difference between the query time and the echo time is the amount of time required for the NonStop ODBC server to translate and execute the SQL statement and to return the results.

When you have finished running experiments, use FILE | DISCONNECT and exit from the Connectivity Tool.

Microsoft ODBC Test allows you to set up and issue single ODBC calls. For example, you can connect to a datasource, execute SQL statements, and read the results (and see them on the screen). You could use ODBC Test as another way to verify connectivity to the NonStop system. In whatever way you verify connectivity, you should also be sure to run some SQL statements either from the Connectivity Tool or from ODBC Test.

3. To run Microsoft ODBC Test, select its icon from the SQL Program Group and follow directions for connecting and for executing SQL statements.

4. When the preceding steps have been completed successfully, you can install and run the client applications. You can run the Connectivity Tool or Microsoft ODBC Test at any time to verify connectivity to NonStop, test SQL execution, and measure response times.

UNIX Workstation Installation

This subsection describes how to install a UNIX workstation client that uses DBLIB to communicate with the NonStop ODBC Server.

Installation Procedure

The installation procedure is as follows:

1. Read the softdoc included with the NonStop ODBC Server.
2. Make sure all checks of UNIX software and hardware have been done.
3. Configure the network connection according to vendor instructions. For TCP/IP, you must set the IP address for the NonStop system and the port number to connect to (2314 in this example).
4. Verify that TCP/IP is active and a LAN connection exists between the client workstation and the NonStop system. As discussed under DBLIB Clients on page 2-12 you can perform this step in a number of ways:

   - Use Ping to test physical IP connectivity. Enter the following:

     ```
     /usr/etc/ping Tandem-host
     ```

     You should see the response:

     ```
     Tandem-host is alive
     ```

   - Verify that file transfer works.
   - Start a TELNET session.

5. Run the Sybase or DBLIB embedded application script to configure the DBLIB environment. For Sybase script, type:

   ```
   SYBCONFG
   ```

   The script checks to make sure all required files have been installed correctly. If an error is detected, the user is notified of possible causes and a suggested course of action to correct the problem.

   The script requests the following information:

   ```
   Enter Server Name?
   Enter Query Port Number?
   ```

   For example:

   ```
   Enter Server Name?       ODBCTCP1
   Enter Query Port Number?  2314
   Installation complete.
   ```

   Answer the first request with the logical name (*network-name*) that represents the host NonStop system on which the NonStop ODBC server has been installed.

Answer the second request with the port number (*port*) configured in the TCP/IP SERVICES file on the NonStop system.

The SYBCONFG program creates or adds to the $SYBASE/INTERFACES file entries for this NonStop ODBC server. The format of the file is:

```
server
     query tcp sun-ether server port-number
     master tcp sun-ether server port-number
     console tcp sun-ether server port-number
```

One multiple-line entry is displayed for the NonStop ODBC server. For example:

```
ODBCTCP1
     query tcp sun-ether ODBCTCP1 2314
     master tcp sun-ether ODBCTCP1 2314
     console tcp sun-ether ODBCTCP1 2314
```

The server name must be a name found in the TCP/IP /ETC/HOSTS file, associating this name with an actual Ethernet address. The name in the TCP/IP /ETC/HOSTS file is case sensitive, so it is important to use the same case when answering the prompt in the SUN.INSTALL script.

Note. UNIX systems with Directory Network Services (DNS) installed should update the master set of maps (name service databases) on the designated DNS master server instead of updating the local /ETC/HOSTS file. After the DNS master database is updated, DNS can be used to propagate this change to all client systems, or the underlying system software providing network name lookup can be configured to examine the remote name service master server. (Consult the *Sun 4.x System and Network Administration Manual* for additional details in setting up the Directory Network Services.)

6. After installation is complete, edit the user environment of each intended workstation user of ODBCTCP1 to reflect the default directory. There are two user environment files. The user selects the correct one based on whether the shell is a Bourne-shell or C-shell.

   Different client tools specify different shells. The default shell for the user can be found by looking at the /ETC/PASSWD file. If the last entry on the line for the user is “/bin/sh,” the user default is the Bourne-shell. If it is “/bin/csh,” the default is the C-shell.

   Bourne-Shell: “.profile” looks like:

```
PATH=.:/usr/5bin:/bin:/usr/bin:/usr/ucb:/etc:/usr/etc:/home/sybase/bin
export PATH TERM
SYBASE=/home/sybase
export SYBASE TERM
```

C-Shell: “.cshrc” looks like:

```
setenv SYBASE /home/sybase
set path=(. /usr/ucb /bin /usr/bin /etc /usr/etc /home/sybase/bin)
```

Note. Although additional entries can be placed in the PATH definition, these two entries represent the minimum for running the NonStop ODBC Server client gateway. Examples for these two shell definitions are in the files .PROFILE.EXAMPLE and.CSHRC.EXAMPLE in the extracted Sun installation.

7. Verify that you can connect to the NonStop ODBC Server and execute SQL statements by running the Sybase interactive SQL tool called ISQL.
8. When the preceding steps have been completed successfully, you can install and run the client applications. You can rerun ISQL with the SAMPNEW script at any time to verify connectivity to the NonStop system and test SQL execution.

Examples of Installation

Here are some scenarios of how you might set up your system.

Create a Default Configuration With Load

In this example, ODBC is already installed, but you want to create a new ODBC system catalog and user catalog using the Load option in NOSINST to create a default configuration. If you do this you will be able to use the PC based HP ODBC Configuration Manager to finish the configuration for your system.

On the NonStop side, using TCP/IP communication, follow these steps:

1. Log on as a super-group user (group ID 255).
2. At a TACL prompt, add a =_NSODBC_SYSTEM_CATALOG define to point to where you want the ODBC system catalog to be created:

   ```
   DELETE DEFINE =_NSODBC_SYSTEM_CATALOG
   ```

   ```
   ADD DEFINE =_NSODBC_SYSTEM_CATALOG, CLASS CATALOG, SUBVOL
   $vol.subvol
   ```
3. Change the volume to the location where the ODBC objects are located:

   ```
   VOLUME ODBC-object-file-subvolume
   ```
4. Run NOSINST from a TACL prompt to create a new system catalog, as follows:
   a. Respond with an I at the prompt:

      ```
      ?? ENTER KEY LETTER (shown in uppercase) OF OPTION : I
      ```
   b. You now have the option to change the default subvolumes for various items. If you need to change the NonStop ODBC System Catalog to the volume and

subvolume in which you want the new system catalog to be created, do this now by responding with N when the system prompts you with:

```
?? ENTER KEY LETTER (shown in uppercase) OF OPTION : N
```

c. NOSINST will prompt you for the volume and subvolume with:

```
?? ENTER NEW NSODBC System Catalog Name : $vol.subvol
```

d. Next NOSINST will ask whether the CURRENT DEFAULT SETTINGS are correct. If they are, press Return:

```
?? ENTER KEY LETTER (shown in uppercase) OF OPTION :
return
```

e. NOSINST will ask if you want to scan the entire system for previous versions of the software. If you don't want to do that, press Return:

```
?? ENTER KEY LETTER (shown in uppercase) OF OPTION :
return
```

f. Finally, NOSINST will ask if you want to continue or exit. If you want to continue, press Return:

```
?? ENTER KEY LETTER (shown in uppercase) OF OPTION :
return
```

g. NOSINST completes the system catalog installation and exits.

5. Run NOSINST from a TACL prompt, to create a user catalog and fill the ODBC tables with default configuration information so that the ODBC Configuration Manager can be used for the rest of the configuration:

a. Respond with the letter L at the prompt:

```
?? ENTER KEY LETTER (shown in uppercase) OF OPTION : L
```

b. You now have the option to change the default subvolumes for various items. If you changed the NonStop ODBC System Catalog in Step 3 on page 2-36, you should do this again by responding with N when the system prompts you with:

```
?? ENTER KEY LETTER (shown in uppercase) OF OPTION : N
```

c. NOSINST will prompt you for the volume and subvolume with:

```
?? ENTER NEW NSODBC System Catalog Name : $vol.subvol
```

d. Next NOSINST will ask whether the CURRENT DEFAULT SETTINGS are correct. If they are, press Return:

```
?? ENTER KEY LETTER (shown in uppercase) OF OPTION :
return
```

e. NOSINST will ask if you want to scan the entire system for previous versions of the software. If you do not want to do that, press Return:

```
?? ENTER KEY LETTER (shown in uppercase) OF OPTION :
return
```

f. Finally, NOSINST will ask if you want to continue or exit. If you want to continue, press Return:

```
?? ENTER KEY LETTER (shown in uppercase) OF OPTION :
return
```

g. NOSINST will ask if you want Add Administrator Default Config. Respond with Y:

```
?? Do you wish to Add Administrator Default Config (Y/N) :
Y
```

h. You must now provide communication values. You normally use the default values provided by NOSINST.

```
?? Network Protocol, default "TCP/IP" : return
```

```
?? IOP Name, default "$ZTC0" : return
```

i. NOSINST will ask you to confirm the defaults. If they are correct, respond with y:

```
?? Proceed with Add Defaults (Y/N) : y
```

j. NOSINST completes the user catalog installation configuration, and exits.

6. If you do not have a network service name and port number entry in the $SYSTEM.ZTCPIP.SERVICES file, you need to add a line. For example:

```
ODBCTEST        2700/TCP
```

7. To start SCS, issue the following command from a TACL prompt:

```
RUN NOSCOM START SCS scs-name
```

where *scs-name* is the identifier for an SCS configuration.

On the PC client side, using TCP/IP communication, follow these steps:

1. Run the Microsoft ODBC Administrator (select this tool’s icon from the SQL Program Group).
2. Select the option to add a datasource and select the SQL driver. Click OK. You will see the HP NonStop SQL ODBC Setup screen. Enter the following:

Data Source Name	nsoadmin (any name is allowed)
Description	Optional
Protocol	WinSockets
Login Timeout	Use the default, 0 (zero)

Host Name	The IP address corresponding to the installed NonStop system
Port	The port number you entered into $SYSTEM.ZTCPIP.SERVICES in Step 6 on page 2-38.
Database	Master
Access Mode	Use default SQL_MODE_READ_WRITE
Cursor Default Mode	Use default FOR_UPDATE
Transaction Isolation	Use default SQL_TXN_READ_UNCOMMITTED

3. Click OK. You will return to the Data Sources window.
4. In the User Data Sources list, click on the new data source name (nsoadmin) that you just created and click Close.

You are now set up on both the NonStop server side and the PC client side. You can now use the Connectivity Tool on the PC to test your connection.

1. To run the Connectivity Tool select its icon from the NonStop SQL Program Group. It displays the Connecting to Server screen.
2. Fill in the information on the screen:

Data Srcs	nsoadmin
User ID	nsoadmin or dbo
Password	The SUPER.* password you entered in Step 1 on page 2-36

3. Click OK.

After you are satisfied that you have a working ODBC system you may use the ODBC Configuration Manager to further configure your ODBC system.

1. Run the ODBC Configuration Manager. It displays the Session Logon screen.
2. Fill in the information on the screen:

Data Source	nsoadmin
User ID	nsoadmin or dbo
Password	The SUPER.* password you entered in Step 1 on page 2-36

3. Click OK.

Create a New System Catalog With NOSCOM

In the next example, ODBC is already installed, but you want to create a new system catalog and user catalog by using NOSCOM.

On the NonStop server side using TCP/IP communication, follow these steps:

1. Log on as a super-group user (group ID 255).
2. If you do not have a network service name and port number entry in the $SYSTEM.ZTCPIP.SERVICES file, you need to add a line. For example:

   ```
   ODBCTEST       2700/TCP
   ```

3. At a TACL prompt, add a =_NSODBC_SYSTEM_CATALOG define to point to where you want the ODBC system catalog to be created:

   ```
   DELETE DEFINE =_NSODBC_SYSTEM_CATALOG
   ```

   ```
   ADD DEFINE =_NSODBC_SYSTEM_CATALOG, CLASS CATALOG, SUBVOL
   $vol.subvol
   ```

4. Run NOSCOM from a TACL prompt:

   ```
   NOSCOM
   ```

5. Create the ODBC system catalog:

   ```
   NOSCOM> SYSCAT INSTALL
   ```

6. Create a user catalog:

   ```
   NOSCOM> USERCAT INSTALL Guardian-subvolume
   ```

7. Add an SCS configuration:

   ```
   NOSCOM> ADD SCS SCS-name SCS_object $vol.subvol.SCSobj &
   >nos_object $vol.subvol.NOS &
   >nosutil_object $vol.subvol.NOSUTIL swapvol $swapvol-volume
   ```

   Example:

   ```
   NOSCOM> ADD SCS $HRSCS scs_object $SQLCON.ZNOSD31.SCSOBJ &
   >nos_object $SQLCON.ZNOSD31.NOS &
   >nosutil_object $SQLCON.ZNOSD31.NOSUTIL SWAPVOL $ODBC
   ```

8. Add the network service information. For the net_name, use the same name that you used in Step 2 on page 2-40:

   ```
   NOSCOM> ADD net_service SCS-name  net_protocol tcp/ip &
    >net_name network-name iop_name $ztc0
   ```
   (for conventional/classic TCP/IP)

   or

   ```
    NOSCOM> ADD net_service SCS-name  net_protocol tcp/ip &
    >net_name network-name iop_name <parallel tcp/ip process
   name>
   ```

9. Add the serverclass definition:

   NOSCOM> ADD serverclass *server-class-name* &

   >max_servers *max-#-of-servers* & >

   available_servers *#-of-initially-started-servers* &

   >g_username *Guardian-user-name-of-server-class-owner* &

   >profile *profile-for-this-server-class*

   Example:

   ```
   NOSCOM> ADD serverclass hrsc max_servers 3 available_servers
   1 &
   ```

   ```
   >g_username sql.henryr profile hrprof
   ```

10. Add the profile:

    NOSCOM> ADD profile *profile-for-this-server-class*

    Example:

    ```
    NOSCOM> ADD profile hrprof
    ```

11. Add a user:

    NOSCOM> ADD user *logical-user-name Guardian-user-name*

    Example:

    ```
    NOSCOM> ADD user henry sql.henryr
    ```

12. Add server class mapping:

    NOSCOM> ADD smap *SCS-name server-class-name*

    Example:

    ```
    NOSCOM> ADD smap $hrscs hrsc
    ```

13. Add user mapping:

    NOSCOM> ADD umap *SCS-name* &

    alias name *logical-user-name* &

    user_name *server-class-name*

    Example:

    ```
    NOSCOM> ADD umap $hrscs henry ser_name hrsc
    ```

14. Start the SCS process:

    ```
    NOSCOM> START SCS $hrscs
    ```

On the PC client side using TCP/IP communication, follow these steps:

1. Run the Microsoft ODBC Administrator (you can select this tool’s icon from the SQL Program Group).

2. Select the option to add a datasource and select the SQL driver. Click OK. You will see the SQL ODBC Setup screen. Enter the following:

Data Source Name	nsoadmin (any name is allowed)
Description	Optional
Protocol	WinSockets
Login Timeout	Use the default, 0 (zero)
Host Name	The IP address corresponding to the installed NonStop system.
Port	Enter the port number you entered into $SYSTEM.ZTCPIP.SERVICES in Step 2 on page 2-40
Database	Master
Access Mode	Use default SQL_MODE_READ_WRITE
Cursor Default Mode	Use default FOR_UPDATE
Transaction Isolation	Use default SQL_TXN_READ_UNCOMMITTED

3. Click OK. You will return to the Data Sources window.

4. In the User Data Sources list, click on the new data source name (nsoadmin) that you just created and click Close.

You are now set up on both the NonStop server side and the PC client side. You may now use the NonStop Connect Tool on the PC to test your connection.

1. Run the NonStop Connectivity Tool; you can select its icon from the SQL Program Group. It displays the Connecting to Server screen.

2. Fill in the information on the screen:

Data Srcs	nsoadmin
User ID	nsoadmin or dbo
Password	The SUPER.* password you entered in Step 1 on page 2-40

3. Click OK.

After you are satisfied that you have a working ODBC system you may use the ODBC Configuration Manager to further configure your ODBC system.

1. Run the ODBC Configuration Manager. It will display the Session Logon screen.

2. Fill in the information on the screen:

Data Source	nsoadmin
User ID	nsoadmin or dbo
Password	The SUPER.* password you entered in Step 2 on page 2-40

3. Click OK.

Configuration

Configuring the NonStop ODBC Server involves managing NonStop ODBC Server catalogs, usernames, alias names, their associated attributes, and stored procedures. It also involves server classes and process attributes. This subsection describes the basic configuration actions you can take and how you perform them. See Section 4, Managing the NonStop ODBC Server, in this manual and “Managing Customized Catalogs,” in the *NonStop ODBC Server Reference Manual* for complete descriptions of the configuration commands.

Many of the configuration actions involve running the NOSCOM program supplied with the NonStop ODBC Server. You can execute almost all utility statements in one of four ways:

- Using the HP Configuration Manager Tool from a NonStop ODBC installed PC client.
- Using the NOSCOM interface interactively:

```
RUN NOSCOM
NOSCOM> utility-statement  parameters ...
```

- Using the NOSCOM interface from a TACL prompt, using an input file:

```
RUN NOSCOM / IN file-of-utility-statements-and-parameters /
```

- Using pass-through mode from a client application:

```
SELECT "tdm: util utility-statement  parameters ..."
```

If you are using a client application that supports scripts of SQL statements, you can essentially build NOSCOM scripts as batches of pass-through SQL statements.

NOSCOM uses SQL/MP to access the NonStop ODBC Server catalogs, so you must have the required Guardian authority for the accesses.

The NonStop ODBC Server catalogs are expected to satisfy certain integrity constraints. The NonStop ODBC Server does not strictly enforce this integrity, however, because it cannot prevent SQL/MP access by others. The NonStop ODBC Server’s own actions maintain catalog integrity, and NOSUTIL has options to check (VALIDATE) and restore (REFRESH) that integrity.

Not every utility statement causes catalog integrity to be checked, however, which allows you more flexibility in the sequencing and allows you to effect an UPDATE by performing a REMOVE followed by an INSERT.

When configuring the NonStop ODBC Server involves making changes to the NonStop ODBC Server catalogs—and it usually does—do the following:

1. Make the desired configuration changes.
2. Use the VALIDATE utility statement to check NonStop ODBC Server catalog integrity.
3. If Step 2 produces a problem report, decide how to solve the problem or problems.

   In some cases, you can use the ADD TABLE statement to affect the needed changes. An example is a case in which a SQL/MP user creates a table T in catalog C after a NonStop ODBC Server catalog has been built for C (T does not appear in the NonStop ODBC Server catalog).

   After you have fixed the problem or problems, repeat Step 2.

Dynamic Configuration

A powerful feature of the NonStop ODBC Release 2 Server is the ability to change the configuration of a running NonStop ODBC Server system—and to have the changes take effect—without needing to stop and restart the system. The method used for handling configuration changes is based on the system components checking for configuration changes at a system-defined interval. The checking interval is set by the system administrator in the CHECK_INTERVAL_SECONDS value in the system configuration values table. See the description of ZNSSCFG (system configuration values) under “NonStop ODBC Server Mapping Tables,” in the *NonStop ODBC Server Reference Manual* for more details.

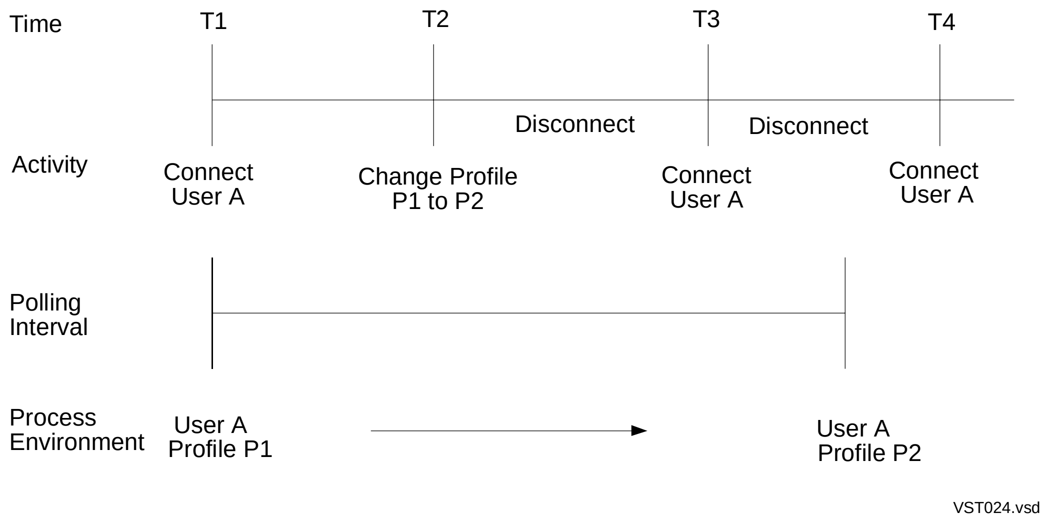
Two important concepts that relate to dynamic configuration are polling and the Configuration Update option:

Polling

Dynamic reconfiguration is based on the concept of system components polling for system configuration changes and then taking action as a result of any change that might affect them. Having a fixed polling interval reduces the effect on the system of having components continually checking for configuration changes, but it also means that a configuration change does not take effect until a component polls for a change.

As an example, consider a NonStop ODBC server process to which a user, User A, has been connected at time T1 with profile P1, as illustrated in Figure 2-2. At some point during the session, time T2, the profile is changed (profile P2). The server does not check to see if the profile has changed for the next connection until the specified polling interval has expired. If User A reconnects at time T3 to the same server, the server continues to use the P1 profile definition, but if User A reconnects after the polling interval has expired, at time T4, the new P2 profile definition is in effect.

Figure 2-2. Timing of Dynamic Configuration



If a new server had been started to service the connection at time T3, or if another server currently running under a different username and profile had been selected, the new profile would have been brought into effect immediately when User A connected.

Configuration Update Option

Another system attribute that controls when some configuration changes take effect is the LAST_UPDATE_SYSTEM_CONFIG attribute, also stored in the system configuration defaults table, ZNSSCFG. This value is a timestamp that indicates when the administrator last made a global change to configuration data. This mechanism lets running processes ignore partial configuration data changes without requiring the administrator to specify update locks. This avoidance of locks allows logins to continue unimpeded while updates are occurring.

When a component checks for a configuration change, it checks to see if the timestamp value of LAST_UPDATE_SYSTEM_CONFIG is later than when the last check was made. If so, the component reviews each configuration entity and modifies its current run environment to reflect any changes. If the timestamp is unchanged, the component can resume processing without further checks on configuration data, as it is either unchanged or within a cycle of change not yet completed. This technique ensures that a coordinated set of configuration changes can all take effect at the same time.

One consideration exists for this mechanism. Newly started servers read all configuration attributes, regardless of whether or not a coordinated multiple attribute configuration is completed—as in the case where the administrator has not completed updates to each of the configuration elements. This activity occurs regardless of the value for LAST_UPDATE_SYSTEM_CONFIG.

If, in Figure 2-2, User A had decided that the profile change from P1 to P2 should not be propagated to all running servers, User A would not have chosen to update the system configuration timestamp; when User A reconnected at time T4, the server would have continued to use the old profile. If a new server were started after the profile change from P1 to P2, however, the new server would have adopted the new profile definition immediately, because it had not been running under the old profile.

Configuring NonStop ODBC Server Catalogs

SQL/MP data is accessible to NonStop ODBC Server users as collections of objects at a named datasource. A datasource is defined by the Network Service Name to which a user connects and by the collection of data groupings available to a user connected to that service. These groupings of data are known as databases. A user connecting to a datasource can specify a specific database or use the default database for that datasource.

An ODBC database contains a collection of accessible objects, referred to in ODBC as ACCESSIBLE_TABLES. In NonStop ODBC Release 1, and by default in Release 2, an ODBC database and a SQL/MP catalog are equivalent collections of accessible objects. The INSTALL process registers each SQL/MP object into the mapping tables.

NonStop ODBC Release 2 Features

NonStop ODBC Release 2 includes a set of features that can be used to customize the ACCESSIBLE_TABLES list to match client application needs. First, you can create a NonStop ODBC catalog independent of the SQL/MP data that will be made available for access by using the ADD {TABLE | VIEW | INDEX} commands. You can modify NonStop ODBC catalog configurations for the access rights and application needs of clients connecting and using SQL/MP data, without redesigning the SQL/MP catalog. SQL/MP data can be customized for access in more than one ODBC system.

A major benefit of creating these virtual databases is the performance gain at connect time, when most applications download all of the metadata in that catalog to the PC platform. SQL/MP catalogs easily grow to a size where the download of metadata exceeds both the space available on the PC and a reasonable time frame for completing the connect operation.

Note. Applications often have the restriction of accessing data from only one database. This can be solved by using the virtual database mechanism to reference objects across multiple SQL/MP catalogs.

Another NonStop ODBC Release 2 feature is the ability to install more than one NonStop ODBC system catalog on a system. Each ODBC system catalog is a complete, independent collection of ODBC configuration data and ODBC-customized SQL/MP data grouped into reference groups, or databases. A user that can connect to a database at a datasource can specify as default any database within that datasource.

Note. N SQL/MP and its underlying file system perform the actual access authorization. The registry list of SQL/MP objects mapped for ODBC access provides a mechanism for customizing access to an application's specific needs for performance and usability. Even if a SQL/MP object is registered in an ODBC database, the user might not have access rights on that object.

As the database administrator you must choose which SQL/MP data objects should be exposed to ODBC clients. The administrator must also determine whether complete SQL/MP catalogs should be customized or whether virtual databases should be constructed, exposing only a selected subset of SQL/MP data.

Listed below are ways in which you might customize your data:

- For each independent collection of configuration data desired, create a NonStop ODBC system catalog. For each collection that is not based on the SQL/MP system catalog, set the DEFINE =_NSODBC_SYSTEM_CATALOG and perform NOSCOM SYSCAT INSTALL. This step is usually performed for the primary ODBC configuration within NOSINST and is indicated for each additional collection of desired configuration data. It is also possible to use NOSINST to perform this step and populate the catalog with default administrator configuration values.
- For each complete SQL/MP catalog that is to be exposed, perform NOSCOM USERCAT INSTALL on the SQL/MP catalog subvolume. This function must be done for each NonStop ODBC system catalog for which access to SQL/MP data is desired.

  Note. The metadata created and stored in the NonStop ODBC catalog tables is loaded only at the time the catalog is first created and populated by USERCAT INSTALL. USERCAT INSTALL of a SQL/MP catalog that already has a completed set of ODBC catalog tables requires only registry in the designated ZNSDB system table.

- For each virtual database, create an independent ODBC user catalog, again using NOSCOM USERCAT INSTALL on an empty subvolume. Then perform the NOSCOM functions ADD TABLE, ADD INDEX, and ADD VIEW for each virtual object reference. Even though an index is not directly accessible by a user, you need to add a mapping entry so that applications can better choose access path and join actions. As with the preceding step, this virtual database must be added to each NonStop ODBC system catalog for which access to this collection of SQL/MP data is desired.
- As the database administrator, you can customize (or de customize) other SQL/MP catalogs at any time, as follows:

  ```
  NOSCOM> USERCAT INSTALL Guardian-subvolume
  ```

  Guardian-subvolume is a file-system subvolume name such as $DATA.MINE. This subvolume can be an existing or new SQL/MP catalog. If the catalog does not yet exist, then a SQL/MP catalog is created before the NonStop ODBC system tables are created. The logical database name is generated as a formatted name, node_volume_subvolume. Components are taken from the fully-qualified Guardian subvolume name provided.

Alternatively, you can customize a catalog implicitly using the TSQL CREATE DATABASE command within a client application, as follows:

```
CREATE DATABASE database-name [ON disk ]
```

Database-name is the logical database name visible to clients. If *disk* is specified, it must be a Guardian subvolume name. Substitutions for missing subvolume name are made from the current user environment. If *disk* is not provided, the database is created in the user's current subvolume.

Similarly, as the database administrator you can issue the NOSCOM command, specifying the logical name:

```
USERCAT INSTALL Guardian-subvolume AS database-name
```

The AS clause lets the user set the logical database name to an identifier other than the previously described formatted database name.

The preceding two commands have equivalent actions. Both build an association between a logical database name and a SQL/MP catalog, creating the NonStop ODBC Server configuration and mapping tables and populating these meta-data tables.

To customize an SQL/MP catalog on \NODE.$VOL1.SUB2 and give it the logical name NODE_VOL1_SUB2, use any of the following commands:

```
CREATE DATABASE NODE_VOL1_SUB2 ON \NODE.$VOL1.SUB2

USERCAT INSTALL \NODE.$VOL1.SUB2 AS NODE_VOL1_SUB2

USERCAT INSTALL \NODE.$VOL1.SUB2
```

Using NOSCOM to Install and Deinstall Catalogs

Corresponding commands to de customize (deinstall) an SQL/MP catalog are also available.

NOSCOM can be used to install and deinstall the system catalog. Typically, NOSINST installs the system catalog so you do not need to do so. Deinstalling the system catalog essentially removes the NonStop ODBC Server from the NonStop system.

Note. NonStop ODBC Server applications and SQL/MP applications (such as Pathway programs) can share SQL/MP catalogs and the objects they contain. If SQL data definition is done from NonStop ODBC Server clients, the NonStop ODBC Server catalogs are kept current. If SQL data definition is done from the SQL/MP applications, NonStop ODBC Server catalogs are *not* kept current—they must be periodically refreshed by using the utility statement REFRESH.

If you have an earlier version of the NonStop ODBC Server installed and you run NOSINST with the UPGRADE option, the NOSINST macro upgrades the NonStop ODBC Server system catalog, but it does not upgrade any of the NonStop ODBC

Server user catalogs. You must upgrade manually for each NonStop ODBC Server user catalog by running

```
NOSCOM> USERCAT UPGRADE NonStop-SQL/MP-catalog-name
```

where *NonStop SQL/MP-catalog-name* corresponds to the NonStop ODBC Server user catalog. You do not need to upgrade all NonStop ODBC Server user catalogs immediately, but it is suggested that you upgrade them before referencing any objects registered in them.

To determine which NonStop ODBC Server catalogs have been upgraded, check the version number of the catalog. You can do so either by issuing the TSQL SELECT statement on SYSOBJECTS (look at the VERSION column) or by issuing the SQL/MP SELECT statement on *SQL-System-Catalog*.ZVSDB (look at the T_VERSION column). The version number of the current release is 110; any lower version number indicates a NonStop ODBC Server catalog that must be upgraded.

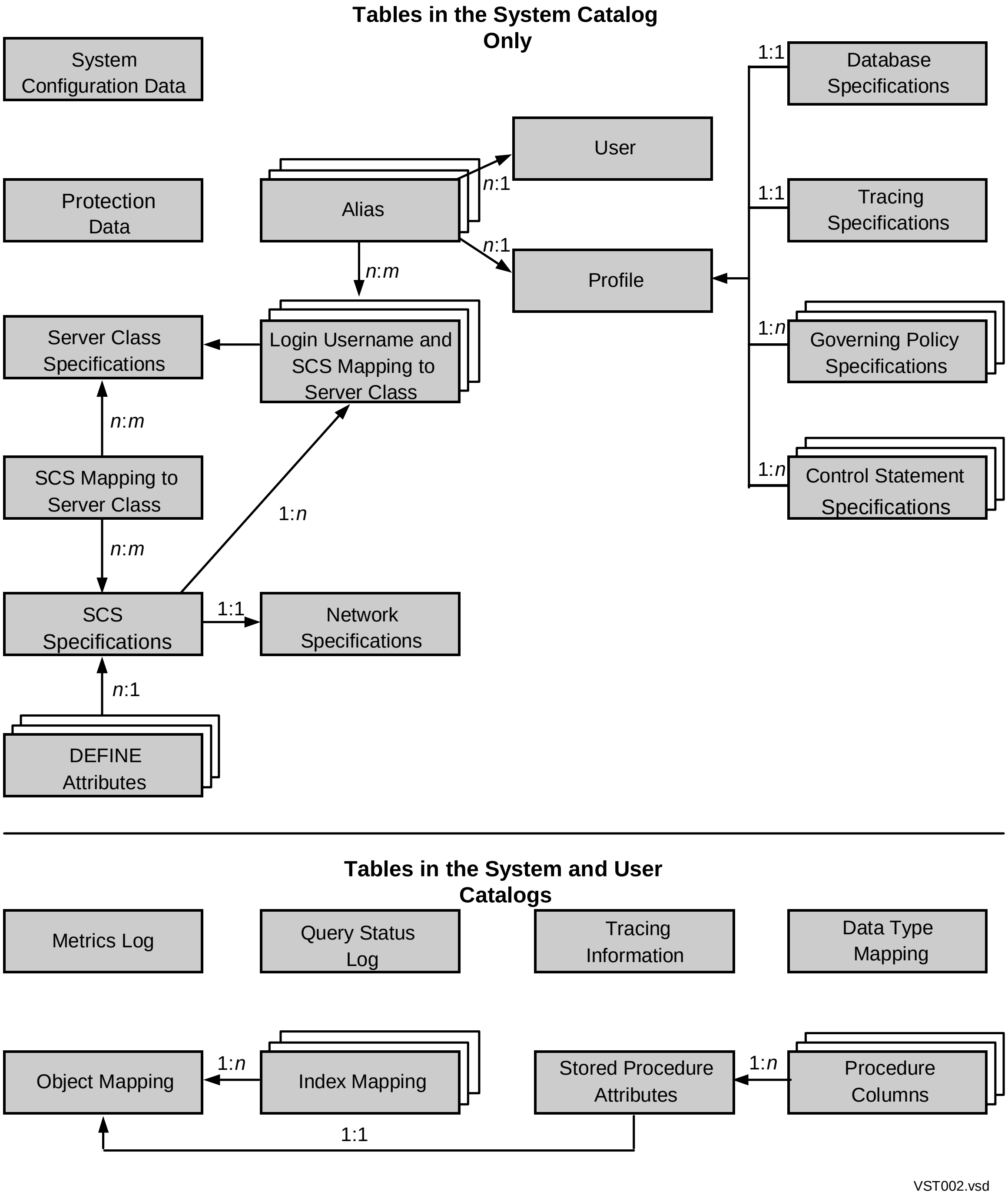
Configuring Usernames and Attributes

NonStop ODBC Server clients connect to servers and access SQL objects by using a logical username. Most NonStop ODBC Server attributes are also associated with usernames. usernames and their attributes must be configured on the NonStop system by using the NonStop ODBC Server Configuration Manager or NOSCOM.

The attribute settings determine the logical, performance, and tracing characteristics of client connections, so their management is a major part of configuration.

Figure 2-3 shows the relationships among the objects and attributes of the NonStop ODBC Server:

Figure 2-3. NonStop ODBC Server Object and Attribute Relationships

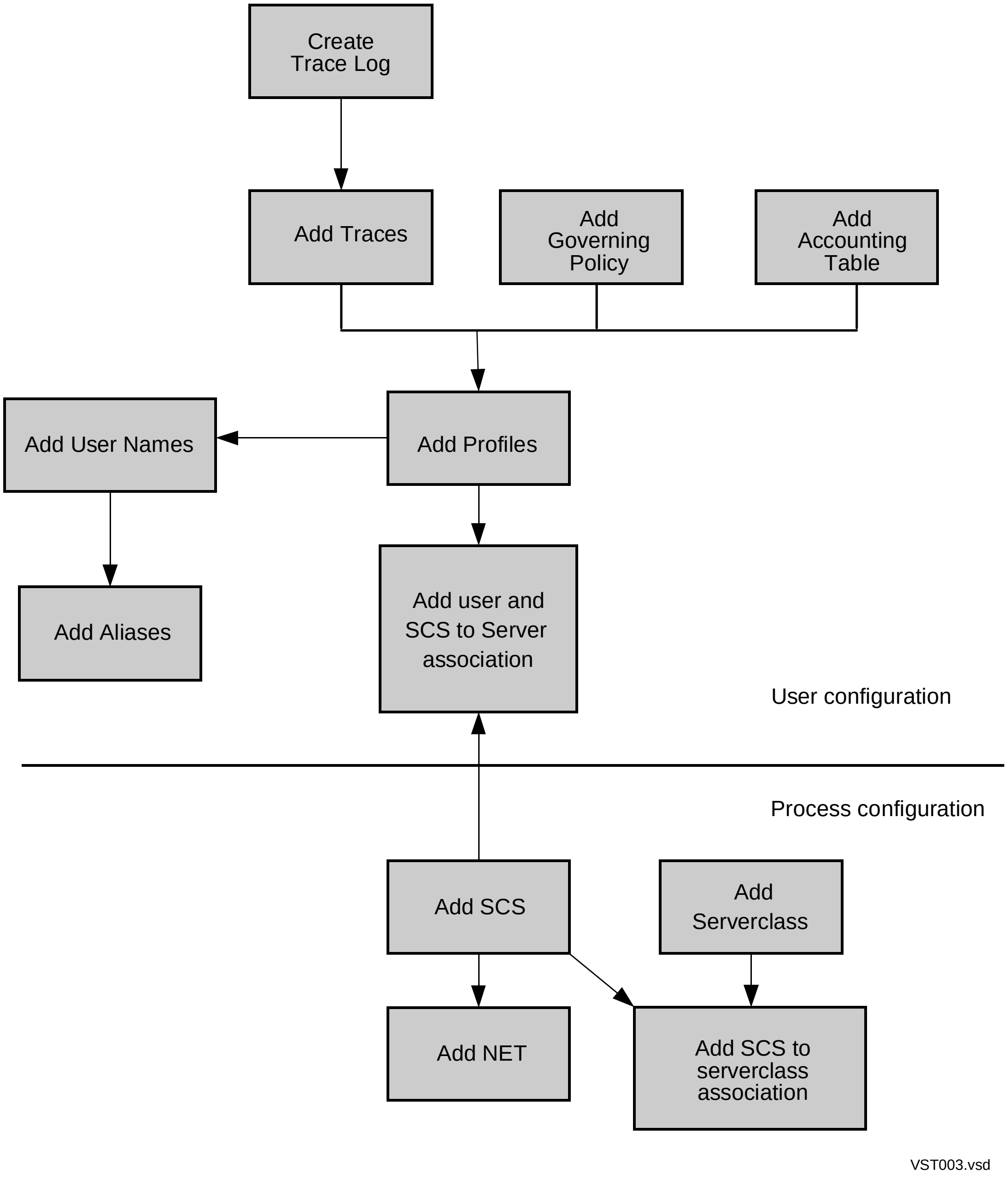


You should first determine the set of usernames and alias names you will use based on your needs and priorities.

Note. If you configure and use alias names, you should remember that alias names are not to be used in qualifying table and other names. Instead, you must use the username associated with the alias name. The effect is as if you connected by using the associated username. A principal use of alias names is when you want different clients that use the same username to have different profiles. If you do not need this feature, it is suggested you do not use alias names.

Follow the basic scheme shown in Figure 2-4 for configuring usernames and their attributes:

Figure 2-4. Configuring NonStop ODBC Server Attributes



When configuring NonStop ODBC, the following ordering will avoid warnings. You first add a Trace Log, then add Trace Configurations, Governing Policies and Accounting Logs. Next, add profiles as pointers to items created in the previous steps, representing different behaviors. You then add usernames and associate them with profiles, and add username aliases. Finally, you add SCS and username associations to named server class configurations. Configuring items in other orders is allowed but will produce warning messages to indicate missing referenced configuration utilities.

The following describes the basic steps for configuring the NonStop ODBC Server attributes:

1. Create profiles representing different behaviors. If these profiles refer to tracing or governing entities, those entities must also be set up.
2. Create the usernames and associate them with one of the profiles set up in Step 1.
3. Create any alias names needed and set up profiles for those alias names.
4. Register the user and alias names with one or more SCS entities.

You can repeat this sequence many times. You can build everything for one or two usernames first—you are not required to create all the profiles and usernames before creating an alias name, for example.

One requirement for the NonStop ODBC Server is that it can be installed and brought up with a minimum of configuration. The NonStop ODBC Server achieves this by using defaults for a number of items:

- DEFAULT profile—the profile is named DEFAULT; NOSINST inserts a row for it in ZNSPROF during the NonStop ODBC Server installation (the column values are all set to default values).
- Default server class—the user explicitly identifies one named server class specification as the default server class for a specific SCS configuration. The same server class can be used as the default in one SCS configuration and not the default in another configuration.
- Default Guardian user name—when a client connects with username *xxx* and no username or alias name *xxx* is configured, the NonStop ODBC Server tries to decode *xxx* to a Guardian user name as follows: if the username is of the form *ggg_uuu*, the NonStop ODBC Server tries to use the Guardian user name *ggg*.*uuu* (*ggg* represents a group name, and *uuu* represents a username within that group).

In other words, if SQL.A and SQL.B are valid Guardian user names on the NonStop system, then usernames SQL_A and SQL_B are configured by default. This feature means that they can connect to and use the NonStop ODBC Server without having been explicitly configured. The NonStop ODBC Server always assigns a DEFAULT profile and the server class designated as the default to such users.

If you do not want the DEFAULT profile or default server class, you must configure a username for SQL.A or SQL.B. You can use any logical name, including SQL_A or SQL_B.

Utility statements such as ADD USER are used to add and delete usernames and their attributes. These statements are described in the *NonStop ODBC Server Reference Manual*.

Configuring ODBC Clients

As described under ODBC Installation on page 2-28, you can use the ODBC administrator tool to add datasources. For each datasource, you can set the database name and data access options.

When a NonStop ODBC server first starts, it reads all user attributes, including default database and user profile from various NonStop ODBC Server catalog tables and establishes the session defaults. When an ODBC client connects to a NonStop ODBC server, it passes the ODBC client options that override session defaults. When the ODBC client is disconnected, the NonStop ODBC server resets to the session defaults.

The following are the ODBC client options that can overwrite the session defaults:

```
SQL_ACCESS_MODE
SQL_CURSOR_MODE
SQL_DIALECT
SQL_TXN_ISOLATION
```

See Performance Tuning on page 4-7 for a full description.

Note. When configuring the ODBC datasource, the datasource settings on the PC *override* the profile settings set up on the NonStop system for that user. It is very important to configure the PC settings for ODBC datasources carefully to ensure the best performance.

Configuring SCS and Server Classes

Following are some SCS configuration considerations that can help improve performance of the NonStop ODBC Server.

Additional Server Classes for Fast Connection

Under Installation Steps on page 2-16, Step 8—Run NOSINST on page 2-18 showed NOSINST being used to configure SCS with only a DEFAULT server class. NonStop ODBC servers in the DEFAULT server class do not provide for the fastest connection by a client, as NonStop processes might need to be started. To achieve fast connections, you will want to configure named server classes for specific usernames.

To use named server classes, use ADD USER and ADD UMAP statements to define a username and specify its server class name. For example:

```
NOSCOM> ADD USER theuser SQL.ODBC PROFILE theprof
NOSCOM> ADD UMAP $thscs theuser SER_NAME theserverclass
```

Statement Caching

You can often obtain greatly improved response time for SQL execution by invoking statement caching, in which SQL compiles are cached: the statement is compiled using the SQL/MP statement PREPARE, and the prepared statement is maintained by the NonStop ODBC Server for reuse.

Statement caching is configured in a user’s profile record by setting the SQL_MAX_STATEMENT_CACHE field to a number greater than zero (cache no more than *n* statements). The default is zero, which means no statement caching.

The NonStop ODBC Server requires additional heap space when statement caching is active. Typically, a simple SELECT (100 characters long) with two input and two output parameters requires about 350 bytes of heap space to cache. You configure additional heap space in the SCS configuration file for the necessary server class. The recommended minimum heap size is 750 KB.)

There are no firm guidelines for when to use statement caching, how many statements to cache, or how much heap space to specify, because all these factors depend on the behavior of the application or the end user. In general, it is best to use as little statement caching as you need, as caching consumes resources.

Use these steps to invoke caching for up to 75 statements for the user SQL.ODBC:

1. Add a profile entry that enables statement caching:

   ```
   NOSCOM> ADD PROFILE cache SQL_MAX_STATEMENT_CACHE 75
   ```

2. Add user SQL.ODBC, specifying the profile you just added, CACHE, and server class CLASS1.

   ```
   NOSCOM> ADD USER sql_odbc sql.odbc PROFILE cache
   ```

3. Caching 75 statements might require 50 KB of memory, so the suggested heap size of 750 KB is probably enough. However, as an example, we will modify this profile to increase the heap size to 800 KB.

   Modify the SCS configuration, setting the heap size for the server class named CLASS1:

   ```
   NOSCOM> MODIFY SERVERCLASS class1 INIT_HEAP_SIZE_KB 800
   ```

4. Start the SCS; your NonStop ODBC server for SQL.ODBC will use statement caching.

Note. If you configure statement caching, make sure that you also set MINSERVERS > 0 for the related server classes so that the NonStop ODBC servers—and their caches of compiled SQL statements—are reused for subsequent connections.

Configuring Stored Procedures

Stored procedures are installed by writing Pathway server programs, configuring and starting the Pathway systems, and registering the stored procedures in the NonStop ODBC Server catalogs. For a complete description of the use of stored procedures, see “Stored Procedures,” in the *NonStop ODBC Server Reference Manual*.

The last step, registration of the stored procedures in the NonStop ODBC Server catalogs, is done by using NOSCOM. For example, to register a stored procedure called MYPROC implemented in server class SPROCS on a Pathway system with Pathmon $PMON1, issue the following command:

```
NOSCOM> ADD PROCEDURE myproc $pmon1 scprocs
```

See “Managing Customized Catalogs,” in the *NonStop ODBC Server Reference Manual* for an explanation of the full set of parameters of ADD PROCEDURE. A stored procedure name can be registered in any number of NonStop ODBC Server catalogs. Stored procedure entries in several NonStop ODBC Server catalogs can all refer to the same service in the same server class of the same Pathway system. Stored procedures registered in NonStop ODBC Server catalogs can refer to several different Pathway systems.

Registration of the stored procedure in the NonStop ODBC Server catalog should be the last step in the configuration procedure, because after it has been completed, clients can start executing stored procedures.

A NonStop ODBC Server catalog from a previous version of the NonStop ODBC Server must first be upgraded before a stored procedure can be registered in it.

Configuration of the Pathway system (server class definition, for example) is independent of the NonStop ODBC Server. If the Pathway system configuration is changed in any way, necessary changes in the NonStop ODBC Server catalogs must be done manually. For example, if the server class name in the preceding example were changed from SPROCS to NEWSPROCS in the Pathway system, the stored procedure name would have to be dropped from the NonStop ODBC Server catalog or catalogs by using REMOVE PROCEDURE and registered again with the proper server class name.

The NonStop ODBC Server system does not monitor the state of the Pathway system. When a client executes a stored procedure, the NonStop ODBC server process simply builds and issues a ServerClass_Send. If the Pathway system is down or misconfigured, the NonStop ODBC server receives an error, which it returns to the client.

Configuring SQL/MP

Some SQL/MP configuration is done by executing special statements from the application. For example, you can issue a CONTROL statement to turn parallel execution on or off. See the *NonStop SQL/MP Reference Manual* for a full list of configuration statements.

The NonStop ODBC Server client application can do the same configuration by issuing the necessary SQL/MP statements in the pass-through mode. This technique is useful only for client applications that can be extended to issue the extra statements. It is not useful for self-contained client applications that build the SQL statements for the user.

To extend the ability of the NonStop ODBC Server user to configure SQL/MP, some of the configuration can now be done by setting DEFINEs on the NonStop system; the client application is not changed at all. Any DEFINEs available to SQL/MP can be used.

The DEFINEs are set in the ZNSDEF table during configuration, either by the NOSCOM interface to NOSUTIL or by the NonStop ODBC Server Configuration Manager. When the SCS process is started, it inherits the DEFINEs. When SCS creates NonStop ODBC servers, they inherit the DEFINEs, and when a NonStop ODBC server starts a SQL/MP compiler process, it inherits the DEFINEs. Thus, the DEFINEs apply to all of the NonStop ODBC servers run under the given SCS.

Two DEFINEs commonly used with the NonStop ODBC Server are as follows:

- =_SQL_CMP_PARALLEL_ON enables SQL/MP parallel execution.

  The SQL/MP compiler recognizes the presence of this DEFINE. If it exists, parallel execution is enabled. Enabling parallel execution causes the compiler to consider, but not necessarily choose, a parallel plan.

  NOSUTIL sets this DEFINE automatically at the time the SCS is started.

- =_SQL_EXE_STOP_SORTPROGS terminates sort processes.

  The SQL/MP executor recognizes the presence of this DEFINE. If it exists, the executor stops all sort processes started on behalf of a cursor whenever the cursor is closed.

Troubleshooting

If you encounter problems with any of the installation steps, do the following first:

- Use the appropriate INFO statements in NOSCOM to gain information about the components involved.
- Check the step you just attempted to make sure you followed directions. Also verify that the previous step was completed successfully.
- Make sure you have read the softdoc to be aware of any late changes.
- Reread the appropriate topics in this section. See also Section 4, Managing the NonStop ODBC Server.

This subsection describes some tools that can assist you in minimizing errors when installing the NonStop ODBC Server, some areas to inspect for possible failures in installation and operation, and some considerations for system performance.

NOSINST

NOSINST is a TACL macro used to install many components of the NonStop ODBC Server on a NonStop system. It has the following functions:

- Installs the NOSUTIL and NOSCOM utility programs on *NOS-Target-Disk*.ZNOS and SQL compiles the program.
- Installs the NonStop ODBC server program on *NOS-Target-Disk*.ZNOS, SQL compiles it, and licenses it (its file name is NOS).
- Installs the SCS program on *NOS-Target-Disk*.ZNOS, and licenses it (its file name is SCSOBJ).
- Builds the NonStop ODBC Server system catalog by customizing the SQL/MP system catalog; this function creates several SQL/MP tables, views, and indexes, and populates the tables with data.
- When upgrading from a previous release of the NonStop ODBC Server, NOSINST automatically upgrades the NonStop ODBC Server system catalog to the current version.

NOSINST proceeds until it terminates successfully or until it finds an error. In the latter case, it reports the error and sometimes suggests a corrective action. It also suggests the NOSINST option to use when you run NOSINST again to resume the installation after you have fixed the problem.

If you currently have NOSUTIL and NOSCOM installed, the ALL and INSTALL options will fail. The SYSCAT UPGRADE option is the only one you may use.

The NOSINST options are as follows:

ALL — Performs the COMPILE, INSTALL/UPGRADE, and LOAD steps in order.

COMPILE — Installs and SQL compiles NOSUTIL (the utility program that manages the NonStop ODBC Server catalog tables) and NOS (the NonStop ODBC server program) and registers them in the SQL/MP system catalog (usually $SYSTEM.SQL). Also licenses NOS and SCSOBJ.

INSTALL — Installs the NonStop ODBC Server system catalog tables in the SQL/MP system catalog.

SYSCAT INSTALL — Installs a new NonStop ODBC System catalog.

SYSCAT UPGRADE — Upgrades the existing NonStop ODBC Server system catalog to the current NonStop ODBC Server version.

LOAD — Loads default SCS configuration values into the system catalog tables.

HELP — Provides on screen help information.

EXIT — Terminates NOSINST without completion.

Using Tracing Facilities

The NonStop ODBC Server contains a number of tracing facilities that can be used to determine the status of the system and to aid in debugging.

Following are descriptions of some of the tracing facilities available. See Section 4, Managing the NonStop ODBC Server, for greater detail.

RUNFILTR

RUNFILTR is a macro supplied with the NonStop ODBC Server. It prints all EMS event messages generated by the NonStop ODBC Server. These messages pertain to such events as starting and stopping NonStop ODBC servers. You can start RUNFILTR before or after you start the SCS; starting it before helps you to troubleshoot problems that occur during startup.

You must run RUNFILTR on a separate terminal or in a separate window. Position on *NOS-Target-Disk*.ZNOS and, from a separate TACL prompt, type the following:

```
RUN RUNFILTR
```

NonStop ODBC Server Tracing

The NonStop ODBC server processes can be configured to trace on a user-name basis. Tracing is enabled by setting the TRA_MODE_ON attribute to “Y” in the user profile. The level of tracing to be used (INPUT STREAM, for example) is specified by the trace definition referenced in the user profile.

You use the ADD TRACE utility statement to set up a trace definition, then use the ADD PROFILE or MODIFY PROFILE statement to set the TRA_MODE_ON and TRACE_NAME options.

△ **Caution.** If you have enabled OUTPUT_STREAM tracing for a session, do not issue a SELECT of a log table during that session; this action causes the trace to enter an infinite loop writing to the log table. Disable the trace by issuing a pass-through statement of the form

```
SELECT "tdm:set tra_mode_on n"
```

then select from the log table, or examine the log table using a different username, or use SQLCI on the NonStop system.

To disable tracing for a given profile, just set TRA_MODE_ON to “N.”

The SQL Communication Subsystem

If the START SCS statement fails to bring up the SCS, check the following:

- Check that your configuration is valid by using the INFO CONFIG statement.
- Start RUNFILTR, then reissue the START SCS statement. Inspect the RUNFILTR output for error indications. If other SCS processes are running, you can identify records for your SCS by matching the *`process-name`* you set in the SCS section.

Two of the most common errors reported by the SCS during startup are:

34005 This error indicates that the SCS had a problem finding the network name in the TCP/IP SERVICES file ($SYSTEM.ZTCPIP.SERVICES). Check to make sure that the network name in the SERVICES file is all uppercase and that it matches the name specified in the SCS configuration file. Also check to make sure that the SERVICES file is located in the subvolume $SYSTEM.ZTCPIP and that it has the proper file security to permit the SCS process to read it.

34009 This error indicates that you tried to start SCS while running under an improper user ID. You must start the SCS under another username.

See the *NonStop ODBC Server Messages Manual* for detailed information on these and other event messages generated by SCS.

Checking Connections

If your SCS process starts, but clients cannot connect to an NonStop ODBC server, do the following:

- Check that there is network connectivity between the client and the NonStop system, as described in the installation procedure. For example, if you are in a Windows environment, you can use the HP Connectivity Tool.
- Check that the client application is using the proper username or alias name and has the proper password to connect.

- Check error messages you receive from the client application when a connection is attempted.
- Check your NonStop configuration by using the appropriate INFO statements.
- If you have configured the NonStop ODBC Server to prelaunch NonStop ODBC servers, check that they are running and are using the correct Guardian user name.
- Use the SCF interface to check on the status of the SCS and the NonStop ODBC servers. For more information, see Section 4, Managing the NonStop ODBC Server.
- Start RUNFILTR, then try to establish a client connection. Observe the RUNFILTR trace output and look for error indications. If other SCS processes are running, you can identify records for your SCS by matching the *process-name* you set in the SCS section.

  Following is an example of RUNFILTR output when a client connects to a NonStop ODBC server named $X716:

  ```
  95-01-27 09:01:15 \TESS.$ZT1       Tandem.SCS.Dxx
   000007 $ZDSS.1: session GZORN object state changed to started
  95-01-27 09:01:17 \TESS.$ZT1       Tandem.SCS.Dxx
   000007 $ZDSS.1: server \TESS.$X716 object state changed to started
  ```

- Check the EMS messages for any information logged by the NonStop ODBC server.

Checking Sessions

After a client has connected to a NonStop ODBC server successfully, these problems could occur with the session:

- The client encounters errors whenever it tries to execute SQL statements.

  If the client application does not display error messages, you can invoke NonStop ODBC server tracing to display the errors. You should set up a trace definition with TRA_ERROR set to “Y.” You can then turn on NonStop ODBC server tracing by setting TRA_MODE_ON to “Y” in your profile.

  Check that the SQL language dialect in your ZNSPROF record is set correctly.

- The session appears to be hung. The client has issued a request, but has not received a reply.

  Verify that the client is indeed waiting for a reply. If the client user is entering SQL statements directly, as with ISQL, client status is usually easy to determine. If the client user is running a self-contained application, however, status may be more difficult to determine.

  There can be a variety of causes for a hung session:

  - A state where some set of processes is mutually waiting (this usually happens in the communications code)

- The components are in fact working, but more slowly than you expect—SQL/MP execution is taking a long time, the NonStop system is heavily loaded, or the LAN has heavy traffic. This situation can happen either because you did not configure correctly (for example, you intended to prestart NonStop ODBC servers and use statement caching, but you made a mistake and this scenario did not occur) or because the system load is heavy.

To troubleshoot a hung session, perform the following checks:

- Observe the RUNFILTR trace to be sure that the client has a session and that there are no unexpected errors.
- Use the SCF to look for active sessions and to check that there is an active session for this user.
- If the client hangs repeatedly, turn on tracing for that user and check the trace log. You should trace the input and output streams. Alternatively, for ODBC clients, you can turn on ODBC driver tracing by using the HP Debug Tool. This trace should show the last ODBC API call issued.
- Stop the NonStop ODBC server associated with the session. If the client returns an error, this indicates the client was hung waiting for the NonStop ODBC server; if the client is still hung, the problem is with the client.

Uninstalling the ODBC Driver

To uninstall the 2.10 Driver of HP's NonStop ODBC, double-click the Uninstall icon in the HP NonStop ODBC program group. The program automatically uninstalls the current release of HP NonStop ODBC and all associated files. Uninstall deletes the HP NonStop ODBC driver. It does not delete the Driver Manager or existing data sources. However, these data sources are not usable until the HP NonStop ODBC driver is reinstalled.

Only drivers that have been installed since the most recent release can be uninstalled using this method.

3

Using the NonStop ODBC Server Configuration Manager

This section describes how to use the NonStop ODBC Server Configuration Manager to set up the components for your server.

The NonStop ODBC Server Configuration Manager is a Windows application that provides a graphical user interface to a NonStop ODBC Server configuration setup process. You can use this tool to configure the mapping tables for a particular NonStop ODBC Server user. The Configuration Manager provides these capabilities through a multiple-document interface (MDI) that allows multiple copies of configuration windows to be open simultaneously.

Operating Systems

The NonStop ODBC Server Configuration Manager can run under any of the following operating systems:

- Windows 95
- Windows 98
- Windows 2000
- Windows XP
- Windows NT

If you are using a 486 PC with 8 MB of RAM, you should not open more than four configuration windows simultaneously. Opening more windows degrades performance, and could cause an out-of-memory error, resulting in abnormal termination of the NonStop ODBC Server Configuration Manager.

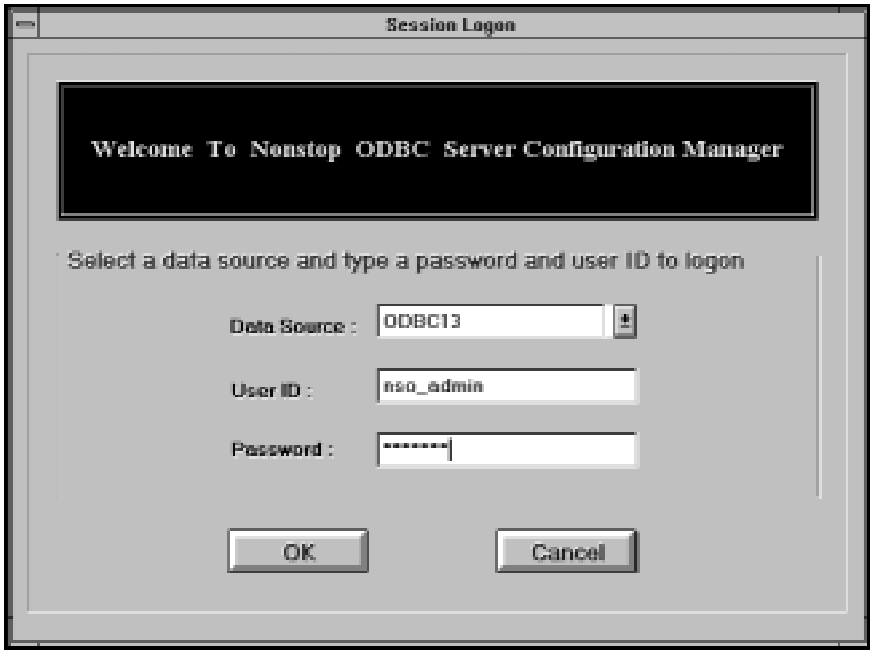
If you are using a Pentium PC with 16 MB or more of RAM and no other applications are running, you can open all 13 configuration windows. However in general, the more windows open, the slower the performance of the NonStop ODBC Server Configuration Manager.

Session Logon Window

When you start the NonStop ODBC Server Configuration Manager, a Session Logon window appears, as illustrated in Figure 3-1.

Note. You must have a running NonStop ODBC Server system before you can use the NonStop ODBC Server Configuration Manager.

Figure 3-1. Session Logon Window



Session Logon

Welcome To Nonstop ODBC Server Configuration Manager

Select a data source and type a password and user ID to logon

Data Source : ODBC13

User ID : nso_admin

Password : ********

VST004.vsd

You can select the data source (an entry in the ODBC.INI file) for which you want to establish a connection either by choosing its name from the Data Source menu or by typing in the name. Clicking OK initiates a separate connection to a particular server (the OK button is not enabled until you have entered a user ID and password). To terminate the connection attempt, click Cancel.

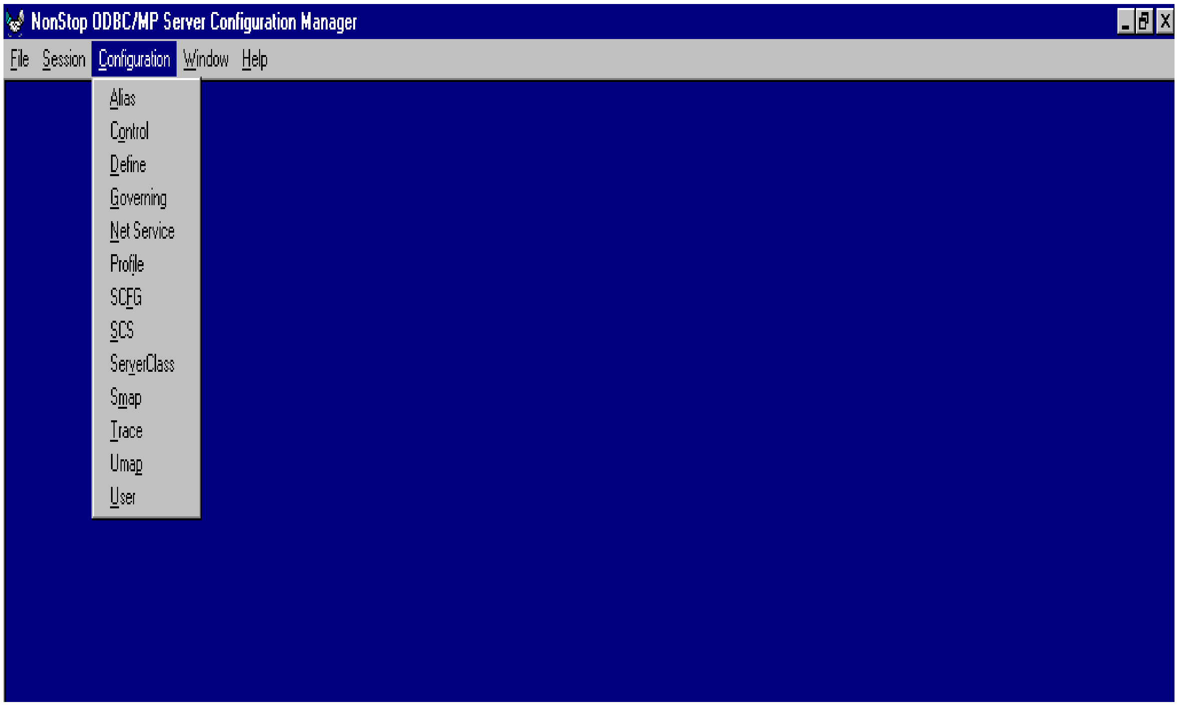
If the connection is successful, a “connection established” dialog box is displayed. Should an error occur (SQLConnect returns SQL ERROR, for example), a dialog box appears describing the error. The possible error messages are described in the appropriate section of the *NonStop ODBC Server Messages Manual*.

Main Window

After you have logged on to the configuration process, the main window appears. It contains the pull-down menus you use to control the configuration session. The main window remains in place while the NonStop ODBC Server Configuration Manager is running.

At the top of the main window is the menu bar containing pull-down menus from which you can choose controlling options. The menu bar is illustrated in Figure 3-2.

Figure 3-2. Pull-Down Menus



VST005.vsd

You can choose a menu item by using the mouse pointer or by using a sequence of key strokes. In all keyboard equivalents of mouse selections, the sequence is initiated by the ALT key followed by the key that corresponds to the underlined letter in the menu or button title.

The following table shows the effects of choosing items under each menu.

Menu Choice	Keyboard Equivalent	Action
File Menu		
Exit	ALT F X	Exit from the NonStop ODBC Server Configuration Manager.
Session Menu		
Connect	Alt S C	Establish a new session with the server.
Disconnect	Alt S FD	Close the current session.
Configuration Menu		
Alias	Alt C A	Open an alias configuration window.
Control	Alt C O	Open a control configuration window.
Define	Alt C D	Open a define configuration window.
Governing	Alt C G	Open a governing policy configuration window.
Net Service	Alt C N	Open a network service configuration window.
Profile	Alt C I	Open a profile configuration window.
SCFG	Alt C F	Open a system configuration window.
SCS	Alt C S	Open an SCS configuration window.
ServerClass	Alt C V	Open a server class configuration window.
Smap	Alt C M	Open an Smap configuration window.
Trace	Alt C T	Open a trace configuration window.
Umap	Alt C P	Open a Umap configuration window.
User	Alt C U	Open a user configuration window.
Window Menu		
Cascade	Alt W C	Lay out multiple-document interface in cascade display mode.
Arrange Icons	Alt W A	Arrange all the miniaturized icons in the work space.
Help Menu		
About Configuration Manager	Alt H A	Display the “About” box for the NonStop ODBC Server Configuration Manager and its version.
Help on Topic	Alt H H	Display onscreen help text for a selected topic.

Configuration Windows

The NonStop ODBC Server Configuration Manager provides 13 windows for configuring NonStop ODBC Server mapping table entries. Each window contains fields for entering specifications and buttons for initiating activities.

The configuration windows are as follows:

Alias	Configures alias names
Control	Configures CONTROL statements
Define	Configures DEFINEs
Governing	Configures governing policies
Net Service	Configures network service specifications
Profile	Configures user profiles
SCFG	Configures system configuration values
SCS	Configures SCS specifications
ServerClass	Configures server class definitions
Smap	Configures server class-to-SCS mapping
Trace	Configures trace specifications
Umap	Configures username-to-server class mapping
User	Configures usernames

Configuration window names are arranged in alphabetical sequence in the pull-down menu, although the Alias, Net Service, Profile, SCS, ServerClass, Smap, and User windows are more frequently used than the others.

Control Windows

Common to all configuration windows is a tab control graphic that resembles the divider tabs used in a filing cabinet. Clicking a tab brings the related control window to the front so it can be used. Up to four control windows—used to add, copy, modify, or remove configuration entries—can be present in a configuration window. If an activity is not appropriate to a given table configuration, that tab is omitted from the configuration window. When a configuration window first opens, the default tab, Add, is visible in front of the other tabs.

The functions of the control windows are as follows:

Add Adds an entry to the selected table.

Copy Adds an entry to the selected table, speeding up the operation by allowing you to search for and copy an existing entry and then make minor changes in the copy, rather than creating the entire entry from the beginning.

Modify Makes changes to an existing table entry (this option is available only for certain fields; it is disabled where not appropriate).

Remove Removes an entry from the selected table.

The Add and Remove tabs are present in all configuration windows. If there is no Copy tab, it is because there is nothing to copy, or else all fields are required. If the Modify tab is missing, the system does not support modification of the selected table.

Action Buttons

Also common to all configuration windows are three buttons: Apply, Cancel, and Close.

Apply Performs the configuration operation you specify.

Cancel Discontinues a configuration operation, leaving tables unchanged, and returns control to the default tab (Add).

Close Closes the control window.

In any window, field titles can appear on the screen in blue, indicating required fields. Failure to enter a required field can result in an error, or else the Apply button can be disabled until all required fields contain entries.

After you click the Apply button, the system displays a message informing you if the configuration operation was successful or if an error occurred.

Window Divisions

A number of configuration windows are divided into three areas: the search box, the list box, and the display box. The procedure for using these areas is as follows:

1. Enter a search criterion in the search box and click the Search button to find all entries in the table that match the criterion. A Clear button is also available to erase the search fields if you make a typing error. Some fields are “comboboxes”: an adjacent button with a downward-pointing arrow displays a pull-down menu from which you can choose a search item; alternatively, you can type the entry in the field.

   You can use wild-card characters in search criteria. Entering an asterisk (*) or a percent symbol (%) causes the search criterion to match any number of characters in that area of the searched field.

When the search is done, items that match the search criteria are shown in the list box. If you clicked the Search button without entering any search criteria, the list box shows all entries in the table.

2. Select an item from the list box by double-clicking on it.
3. If appropriate to the current operation, make any desired changes to fields from the selected items that appear in the display box. When the configuration is to your satisfaction, click the Apply button.

The labels of the search box, the list box, and the display box are numbered in the same sequence as the preceding description.

Subwindows

Five configuration windows—Network Service, Profile, SCS, ServerClass, and Trace—have subwindows with these features:

- In all but the Network Service Configuration window, clicking the Advanced button opens the subwindow, which contains optional entries.
- In the Network Service Configuration window, clicking Protocol opens the subwindow; the subwindow contains entries specific to the network protocol to be used.

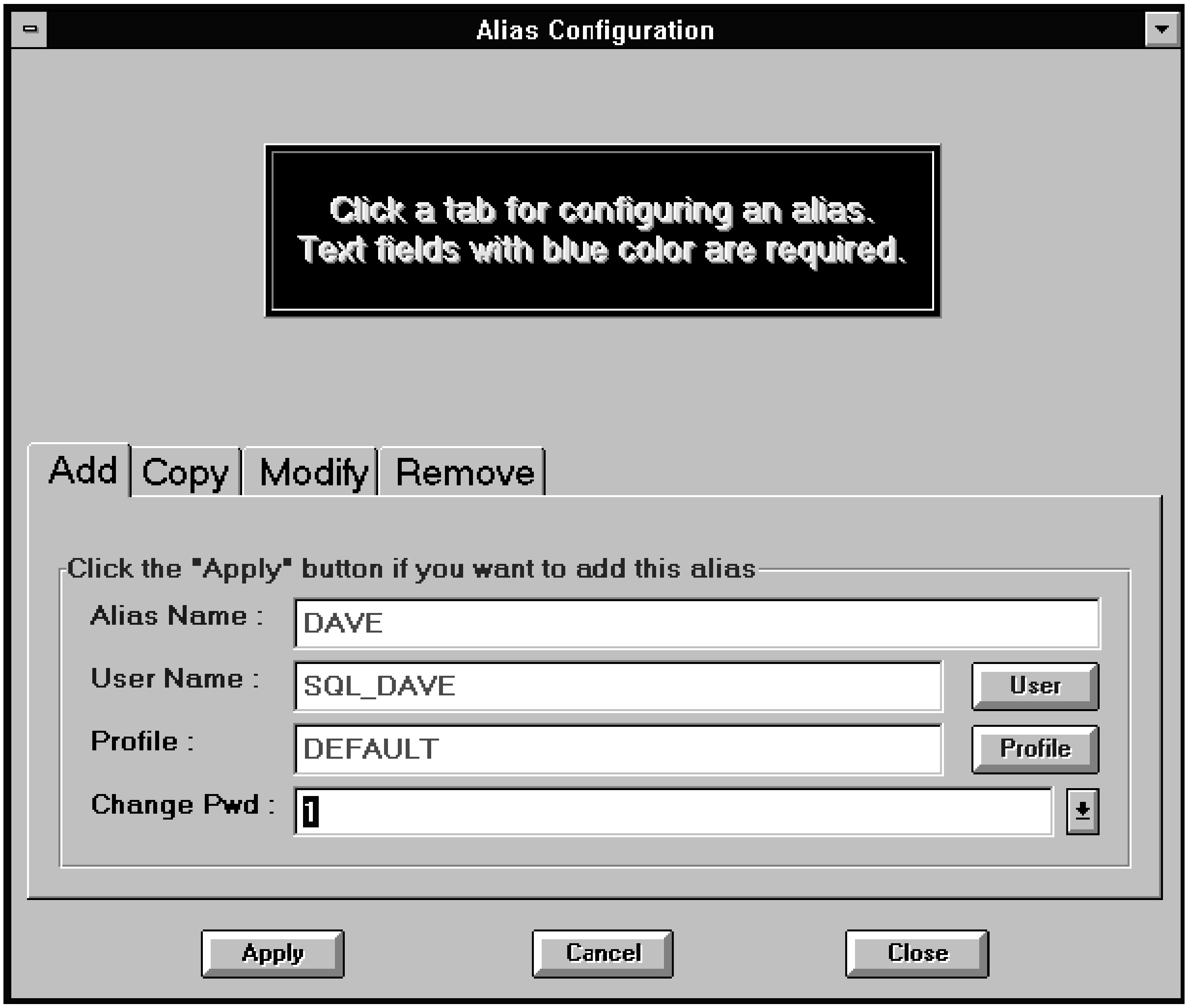
Each subwindow contains a Hide button that closes the subwindow. Clicking the Close button in the parent window also closes the subwindow.

Alias Configuration Window

Use the Alias Configuration window to configure alias names.

Open the Alias Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C A. The Alias Configuration window is shown in Figure 3-3.

Figure 3-3. Alias Configuration Window



Click a tab for configuring an alias.
Text fields with blue color are required.

Click the "Apply" button if you want to add this alias

Alias Name : DAVE

User Name : SQL_DAVE

Profile : DEFAULT

Change Pwd : 1

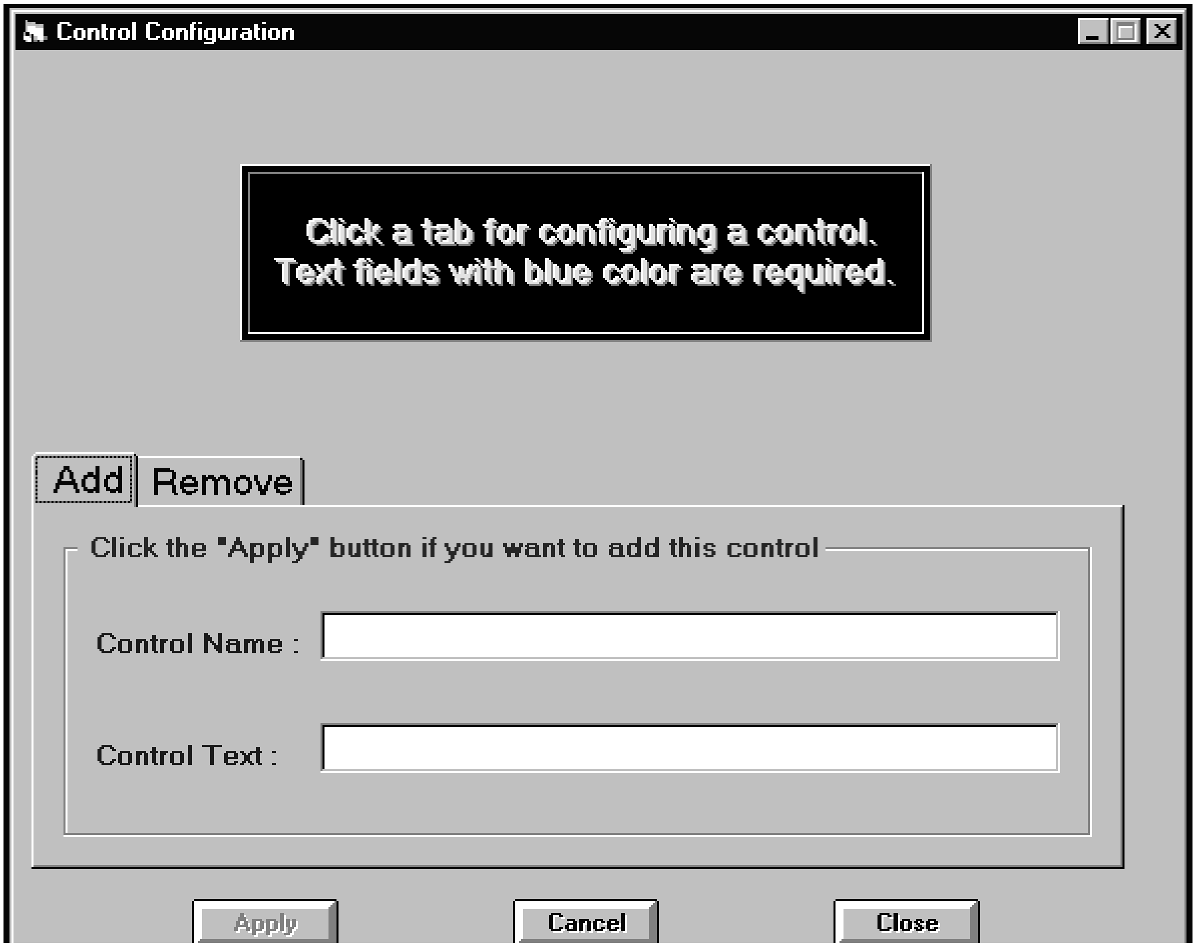
VST006.vsd

Control Configuration Window

Use the Control Configuration window to configure SQL/MP CONTROL statements that are to be executed during process initialization or reinitialization.

Open the Control Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C C. The Control Configuration window is shown in Figure 3-3.

Figure 3-4. Control Configuration Window



Control Configuration
Click a tab for configuring a control.
Text fields with blue color are required.
Add
Remove
Click the "Apply" button if you want to add this control
Control Name :
Control Text :
Apply
Cancel
Close

VST007.vsd

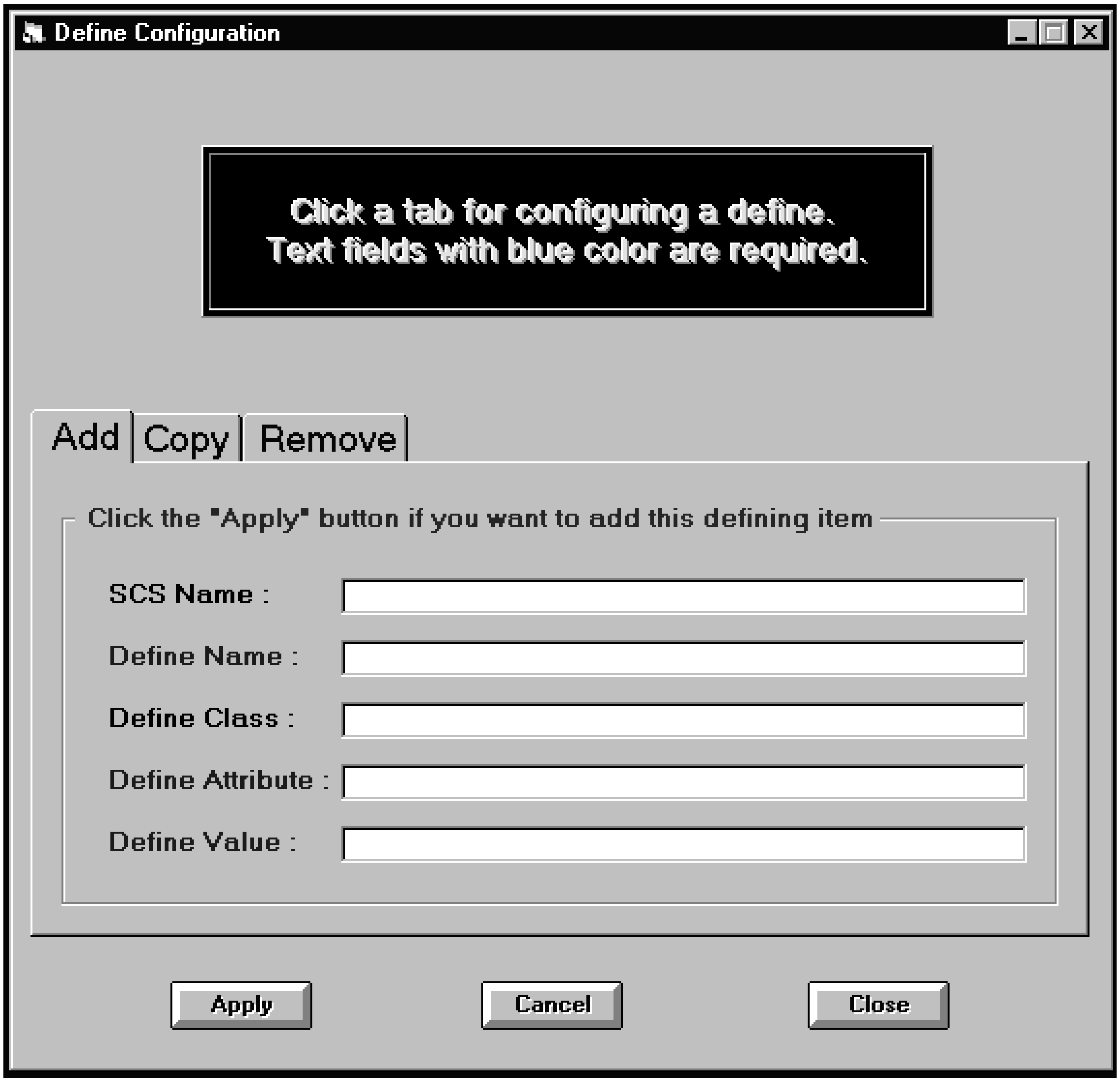
The Control Configuration window supports only adding or removing configuration entries for control statements. The Copy and Modify tabs are not included this window.

Define Configuration Window

Use the Define Configuration window to configure DEFINEs that are to be set prior to starting a given SQL Communication Subsystem and its associated NOSODBC servers.

Open the Define Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C D. The Define Configuration window is shown in Figure 3-3.

Figure 3-5. Define Configuration Window



Click a tab for configuring a define.
Text fields with blue color are required.

Click the "Apply" button if you want to add this defining item

SCS Name :

Define Name :

Define Class :

Define Attribute :

Define Value :

VST008.vsd

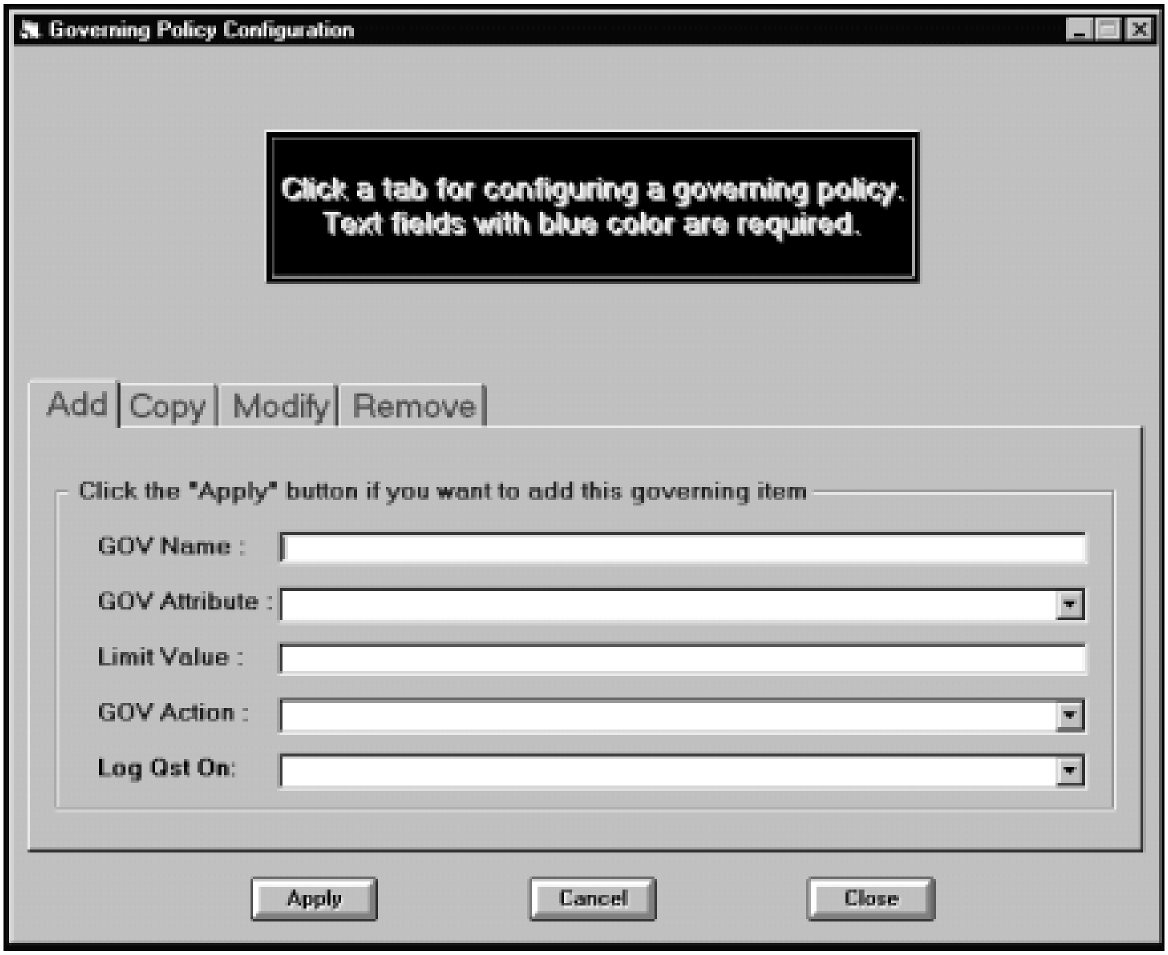
The Define Configuration window allows you to add, add by copying, or remove a configuration entry for DEFINEs. The Modify tab is not included in this window.

Governing Policy Configuration Window

Use the Governing Policy Configuration window to configure governing policies for the resource governing facility.

Open the Governing Policy Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C G. The Governing Policy Configuration window is shown in Figure 3-6.

Figure 3-6. Governing Policy Configuration Window



Governing Policy Configuration

Click a tab for configuring a governing policy.
Text fields with blue color are required.

Click the "Apply" button if you want to add this governing item

GOV Name :

GOV Attribute :

Limit Value :

GOV Action :

Log Qst On:

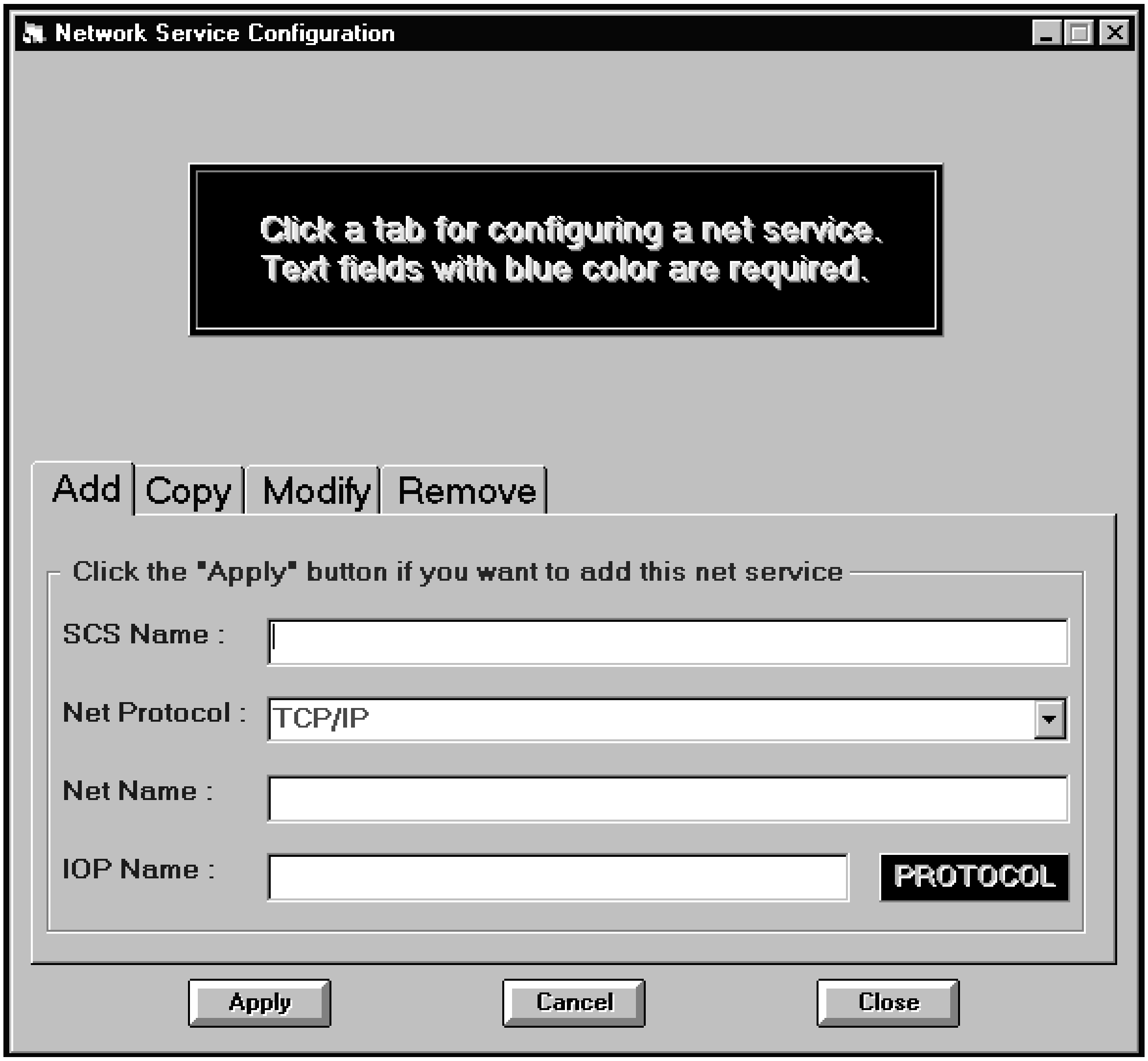
VST009.vsd

Network Service Configuration Window

Use the Network Service Configuration window to configure network definitions, which hold the data needed by the SQL Communication Subsystem to initiate a network service.

Open the Network Service Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C N. The Network Service Configuration window is shown in Figure 3-7.

Figure 3-7. Network Service Configuration Window



Click a tab for configuring a net service.
Text fields with blue color are required.

Click the "Apply" button if you want to add this net service

Net Protocol : TCP/IP

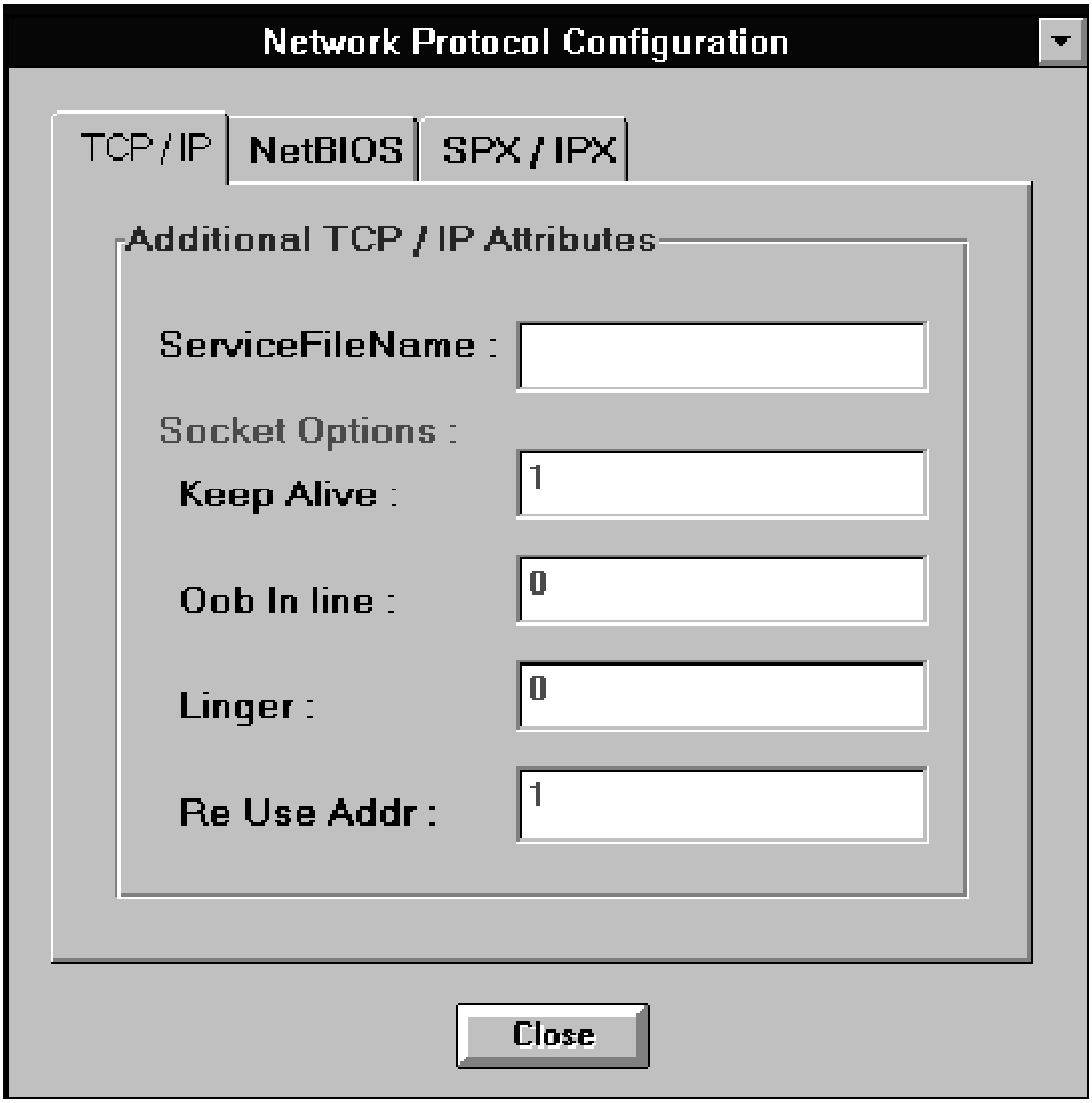
VST010.vsd

The Net Service, Net Protocol, Net Name, and IOP Name entries are required. TCP/IP is the default net protocol

Network Protocol Configuration Subwindow

The Network Service Configuration window provides a subwindow that allows you to make configuration entries specific to the network protocol being used. Clicking the Protocol button opens that subwindow. The Network Protocol Configuration subwindow is illustrated in Figure 3-8.

Figure 3-8. Network Protocol Configuration Subwindow


Network Protocol Configuration
TCP / IP
NetBIOS
SPX / IPX
Additional TCP / IP Attributes
Close

ServiceFileName :

Socket Options :

Keep Alive : 1

Oob In line : 0

Linger : 0

Re Use Addr : 1

VST026.vsd

Specifying a protocol name in the parent window also opens the corresponding Network Protocol Configuration subwindow. Conversely, clicking the Protocol button and selecting a tab in the subwindow causes the chosen protocol name to appear in the parent window.

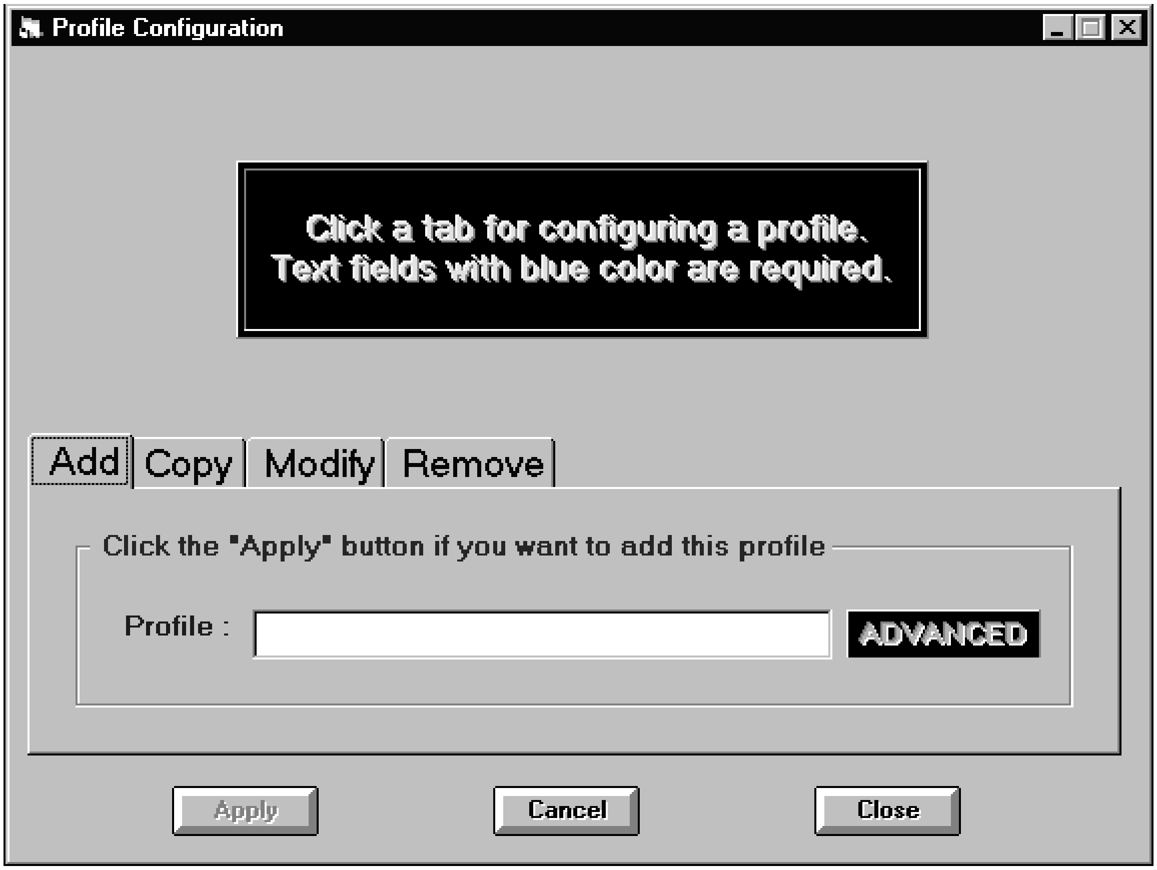
The number of required fields in the Network Protocol Configuration subwindow differs according to the protocol chosen.

Profile Configuration Window

Use the Profile Configuration window to configure user profile information used by the NonStop ODBC server to establish session-based defaults.

Open the Profile Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C I. The Profile Configuration window is shown in Figure 3-9.

Figure 3-9. Profile Configuration Window



Profile Configuration

Click a tab for configuring a profile.
Text fields with blue color are required.

Click the "Apply" button if you want to add this profile

Profile :

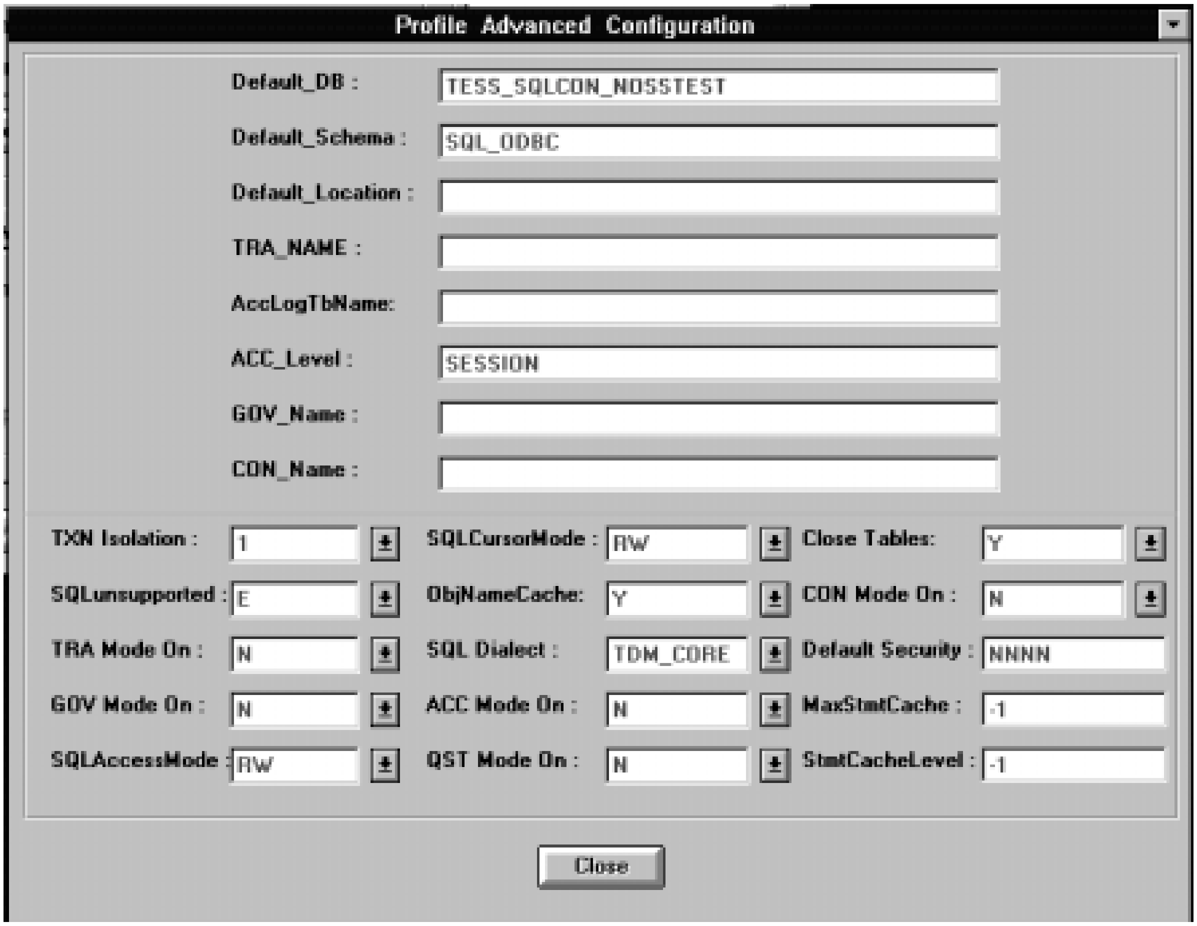
VST012.vsd

Only the profile name entry is required. Other entries are available in a subwindow.

Profile Advanced Configuration Subwindow

The Profile Configuration window provides a subwindow that allows you to make additional, optional configuration entries. Clicking the Advanced button opens that subwindow. The Profile Advanced Configuration subwindow is illustrated in Figure 3-10.

Figure 3-10. Profile Advanced Configuration Subwindow



Default_DB : TESS_SQLCON_NOSSTEST

Default_Schema : SQL_ODBC

Default_Location :

TRA_NAME :

AccLogTbName:

ACC_Level : SESSION

GOV_Name :

CON_Name :

TXN Isolation :	1	SQLCursorMode :	RW	Close Tables:	Y
SQLunsupported :	E	ObjNameCache:	Y	CON Mode On :	N
TRA Mode On :	N	SQL Dialect :	TDM_CORE	Default Security :	NNNN
GOV Mode On :	N	ACC Mode On :	N	MaxStmtCache :	-1
SQLAccessMode :	RW	QST Mode On :	N	StmtCacheLevel :	-1

VST013.vsd

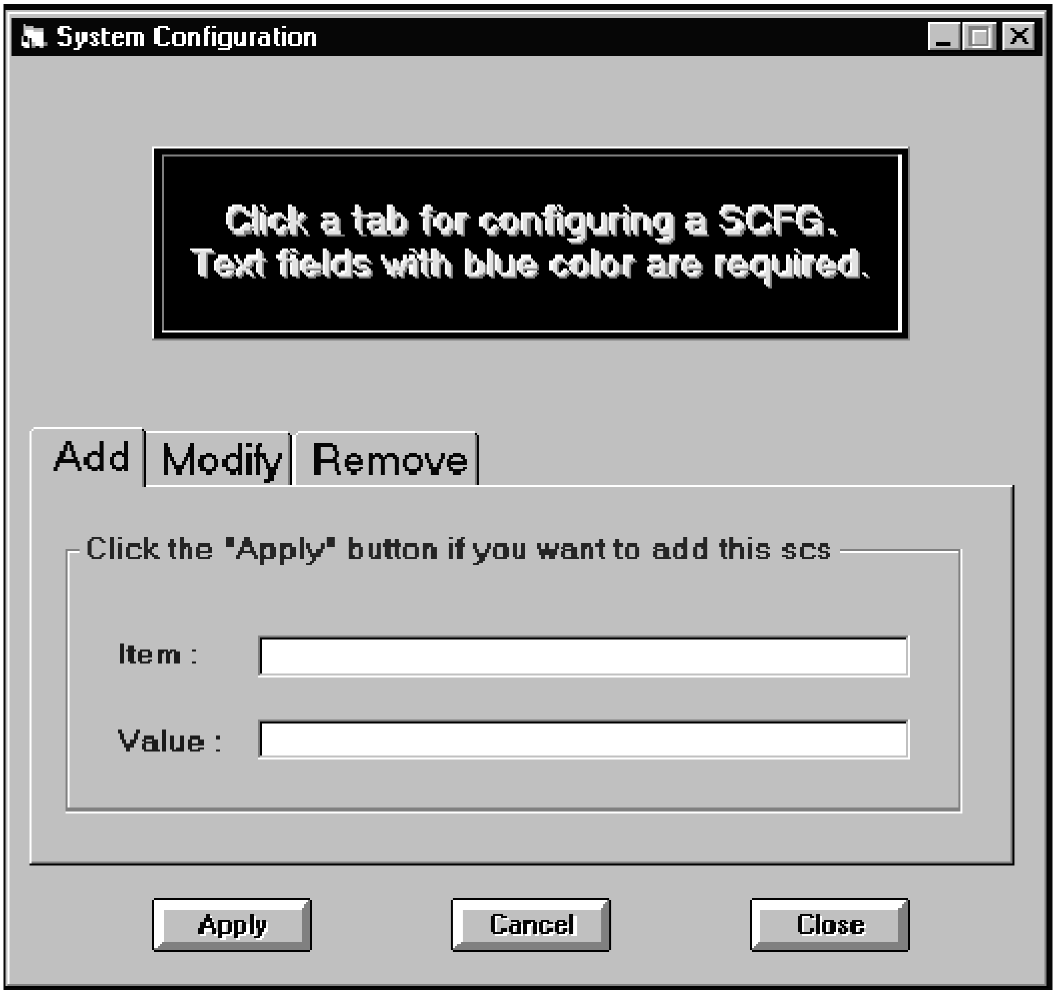
All entries in the subwindow are optional. If a system default exists for an entry, the default appears in the window when you first open it. Some entries have comboboxes to allow selection from a menu; alternatively, you can type in the entries.

System Configuration Window

Use the System Configuration window to configure the system configuration default values used by the NonStop ODBC server and the NOSUTIL process (and, indirectly through NOSUTIL, the SCS process).

Open the System Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C F. The System Configuration window is shown in Figure 3-11.

Figure 3-11. System Configuration Window



Click a tab for configuring a SCFG.
Text fields with blue color are required.

Click the "Apply" button if you want to add this scs

Item :

Value :

VST014.vsd

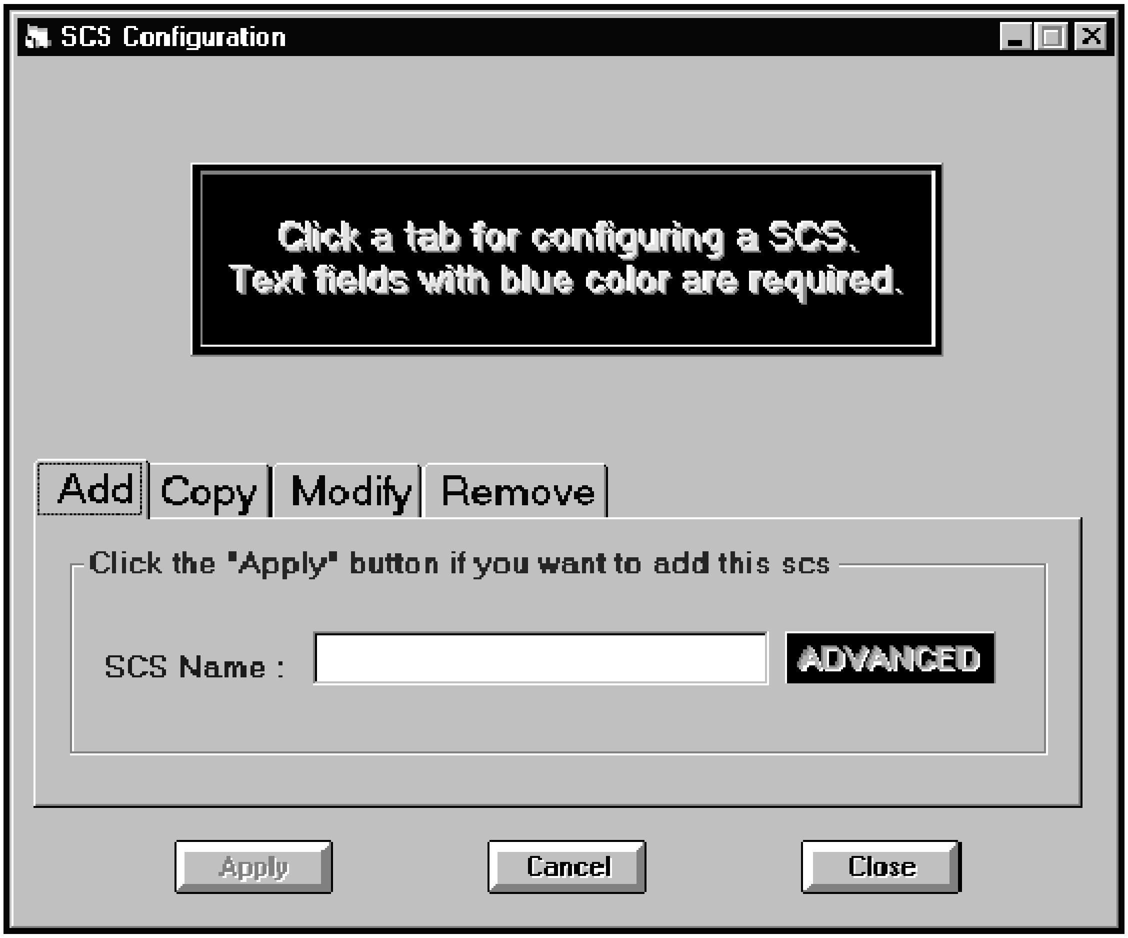
The System Configuration window supports adding, modifying, and removing configuration entries, but the copy feature is not available with this window.

SCS Configuration Window

Use the SCS Configuration window to configure the configuration definitions for the SQL Communication Subsystem.

Open the SCS Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C S. The SCS Configuration window is shown in Figure 3-12.

Figure 3-12. SCS Configuration Window



Click a tab for configuring a SCS.
Text fields with blue color are required.

Click the "Apply" button if you want to add this scs

SCS Name :

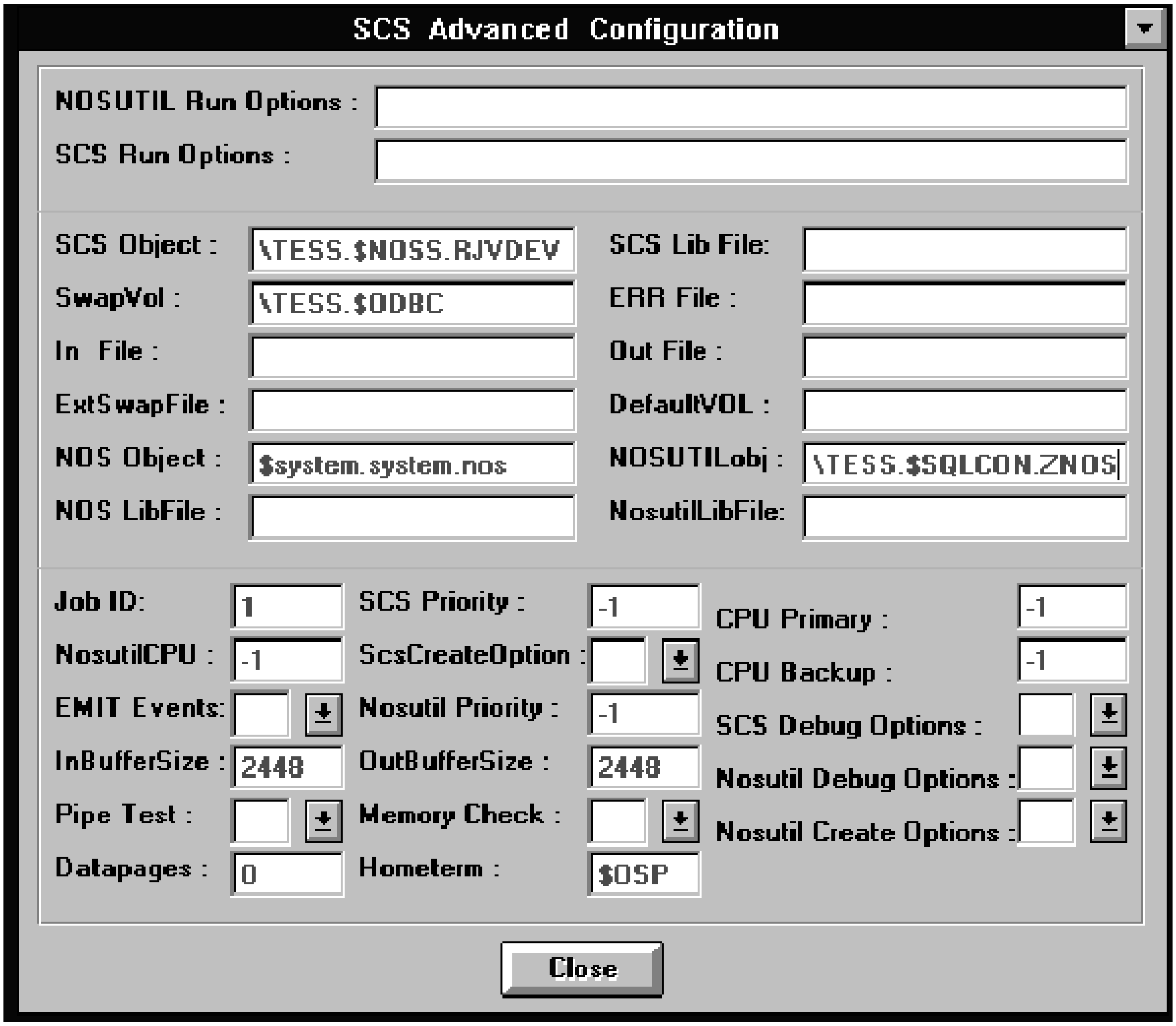
VST015.vsd

Only the name of the SQL Communication Subsystem to be configured is required. Other fields can be entered in a subwindow.

SCS Advanced Configuration Subwindow

The SCS Configuration window provides a subwindow that allows you to make additional, optional configuration entries. Clicking the Advanced button opens that subwindow. The SCS Advanced Configuration subwindow is illustrated in Figure 3-13.

Figure 3-13. SCS Advanced Configuration Subwindow



SCS Advanced Configuration

NOSUTIL Run Options :
SCS Run Options :

SCS Object :	\TESS.$NOSS.RJVDEV	SCS Lib File:	
SwapVol :	\TESS.$ODBC	ERR File :	
In File :		Out File :	
ExtSwapFile :		DefaultVOL :	
NOS Object :	$system.system.nos	NOSUTILobj :	\TESS.$SQLCON.ZNOS
NOS LibFile :		NosutilLibFile:	

Job ID:	1	SCS Priority :	-1	CPU Primary :	-1
NosutilCPU :	-1	ScsCreateOption :		CPU Backup :	-1
EMIT Events:		Nosutil Priority :	-1	SCS Debug Options :	
InBufferSize :	2448	OutBufferSize :	2448	Nosutil Debug Options :	
Pipe Test :		Memory Check :		Nosutil Create Options :	
Datapages :	0	Hometerm :	$OSP		

VST016.vsd

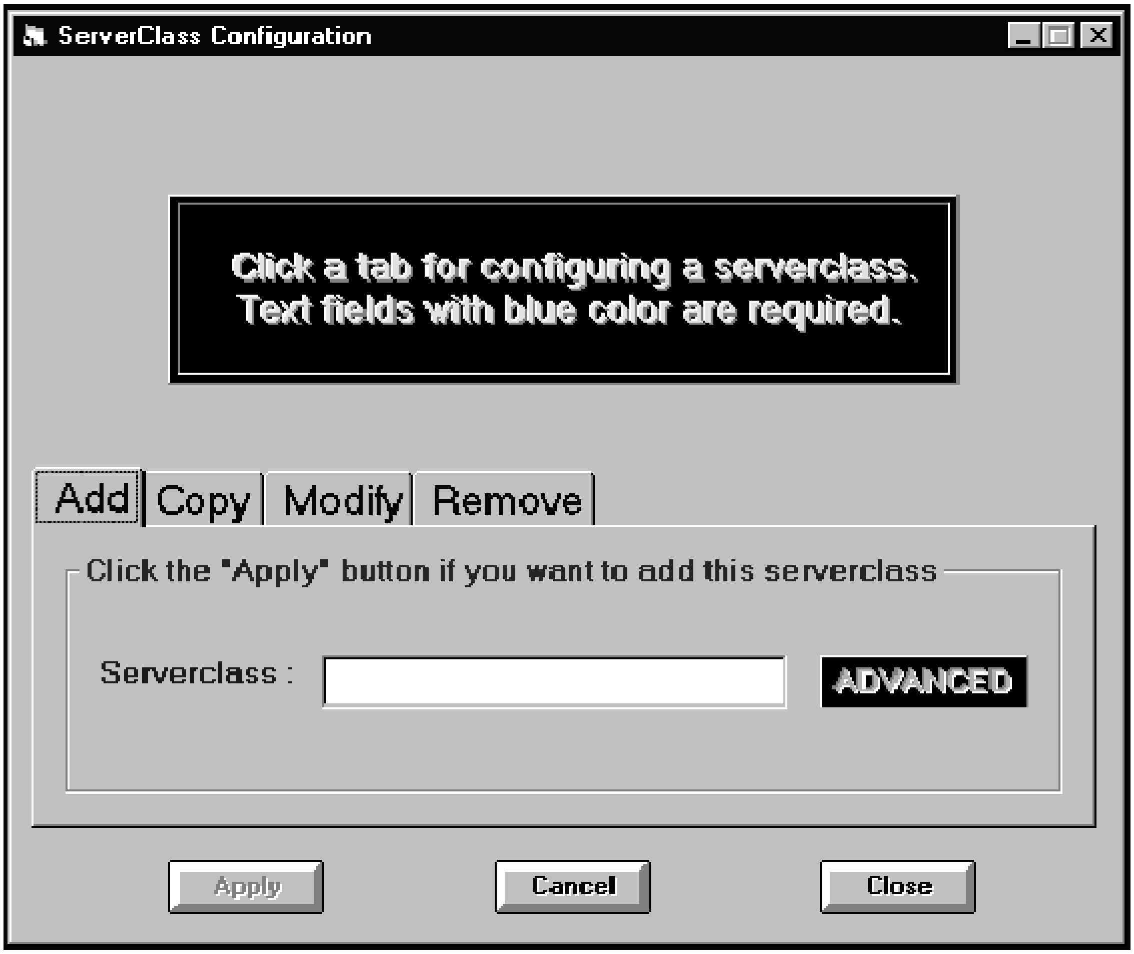
All entries in the subwindow are optional. If a system default exists for an entry, the default appears in the window when you first open it.

ServerClass Configuration Window

Use the ServerClass Configuration window to configure server class definitions.

Open the ServerClass Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C V. The ServerClass Configuration window is shown in Figure 3-14.

Figure 3-14. ServerClass Configuration Window



ServerClass Configuration

Click a tab for configuring a serverclass.
Text fields with blue color are required.

Click the "Apply" button if you want to add this serverclass

Serverclass :

VST017.vsd

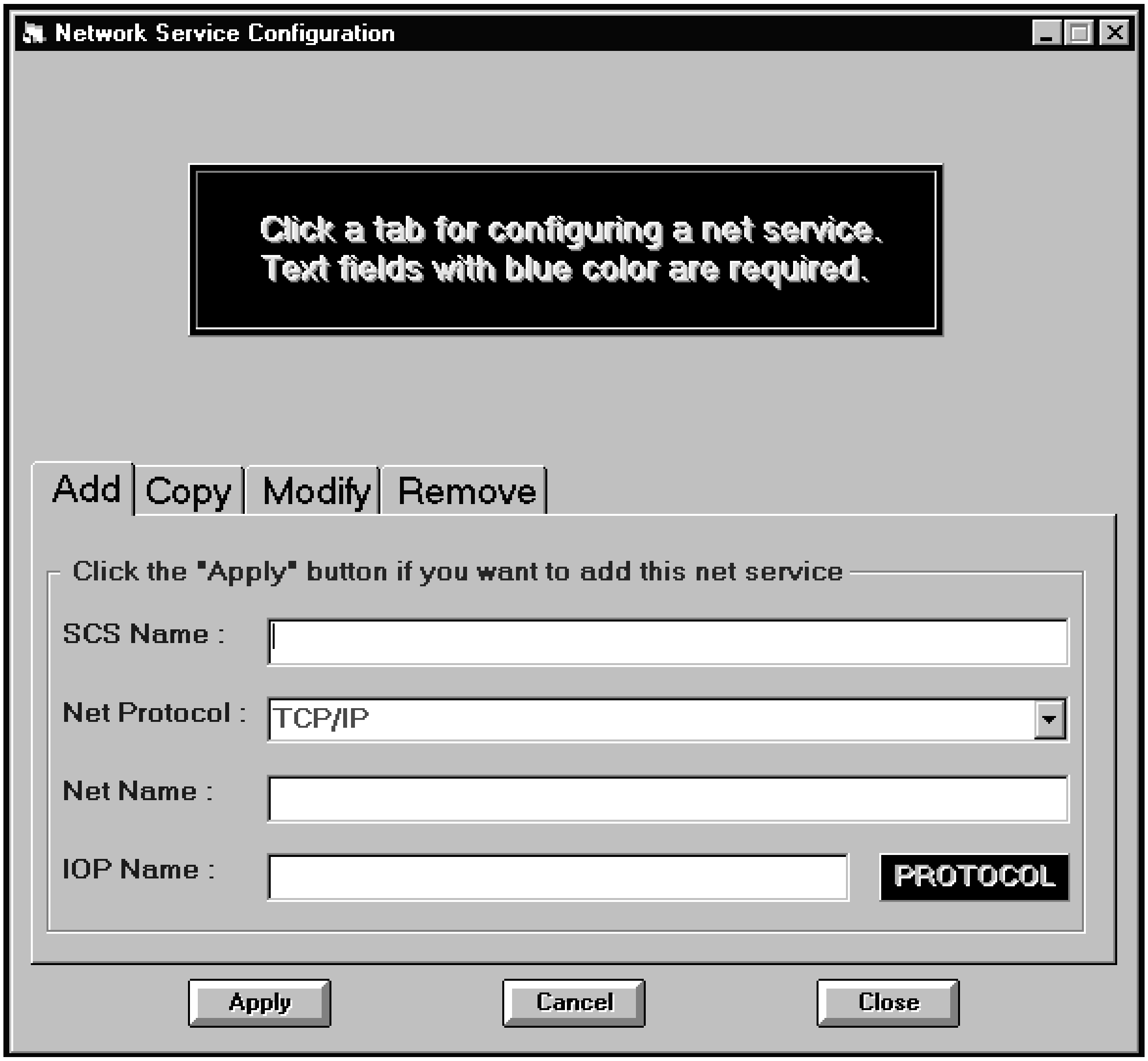
Only the name of the server class to be configured is required. Other fields can be entered in a subwindow.

Network Service Configuration Window

Use the Network Service Configuration window to configure network definitions, which hold the data needed by the SQL Communication Subsystem to initiate a network service.

Open the Network Service Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C N. The Network Service Configuration window is shown in Figure 3-14.

Figure 3-15. Network Service Configuration Window



Network Service Configuration

Click a tab for configuring a net service.
Text fields with blue color are required.

Click the "Apply" button if you want to add this net service

SCS Name :

Net Protocol : TCP/IP

Net Name :

IOP Name :

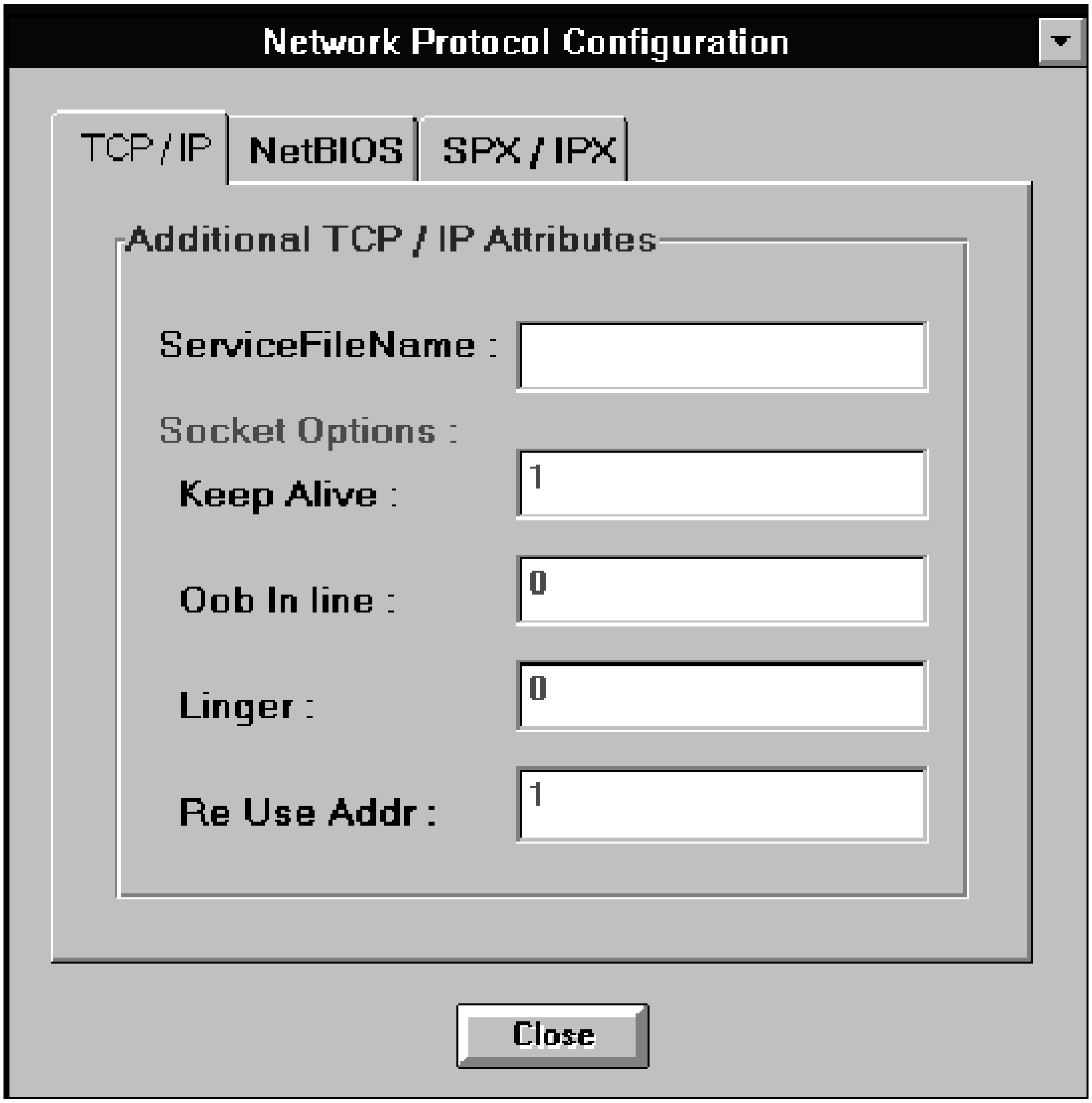
VST010.vsd

The Net Service, Net Protocol, Net Name, and IOP Name entries are required. TCP/IP is the default net protocol

Network Protocol Configuration Subwindow

The Network Service Configuration window provides a subwindow that allows you to make configuration entries specific to the network protocol being used. Clicking the Protocol button opens that subwindow. The Network Protocol Configuration subwindow is illustrated in Figure 3-16.

Figure 3-16. Network Protocol Configuration Subwindow


Network Protocol Configuration
TCP / IP
NetBIOS
SPX / IPX
Additional TCP / IP Attributes
Close

ServiceFileName :

Socket Options :

Keep Alive : 1

Oob In line : 0

Linger : 0

Re Use Addr : 1

VST026.vsd

Specifying a protocol name in the parent window also opens the corresponding Network Protocol Configuration subwindow. Conversely, clicking the Protocol button and selecting a tab in the subwindow causes the chosen protocol name to appear in the parent window.

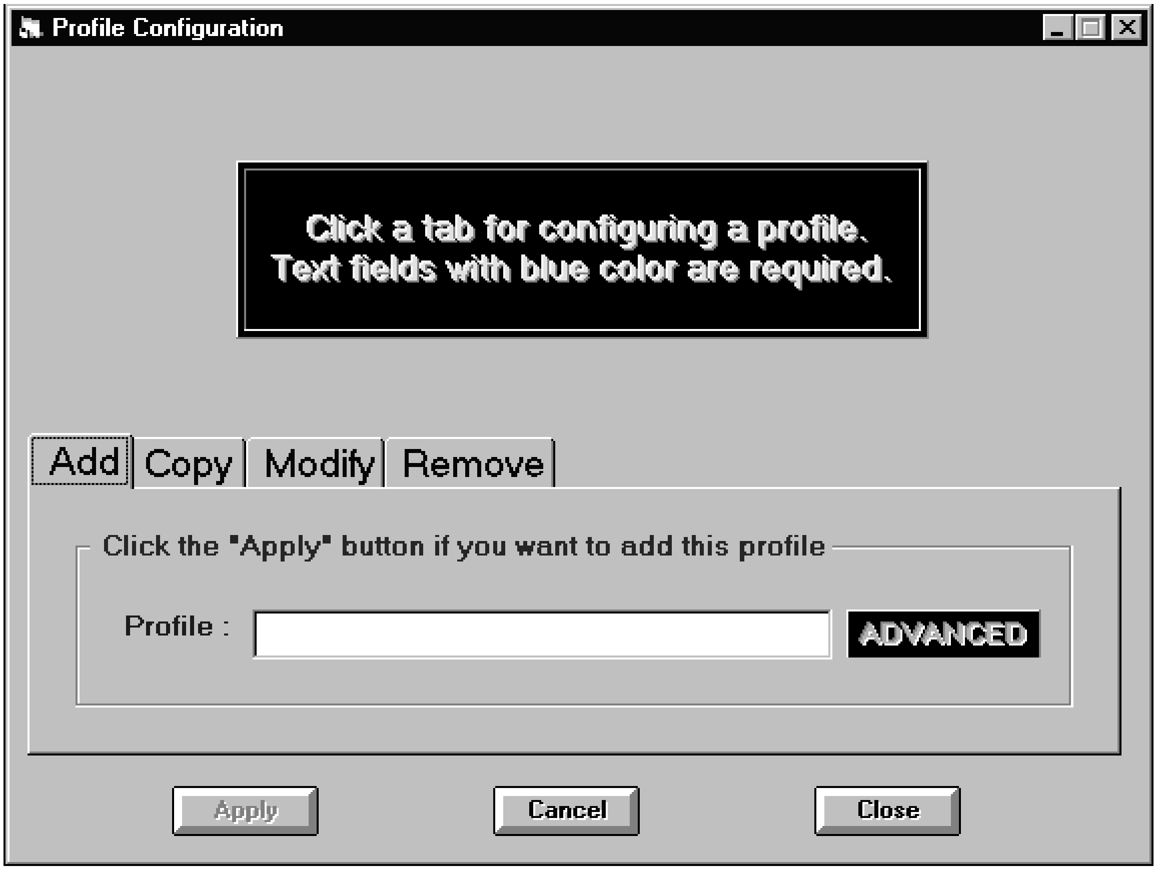
The number of required fields in the Network Protocol Configuration subwindow differs according to the protocol chosen.

Profile Configuration Window

Use the Profile Configuration window to configure user profile information used by the NonStop ODBC server to establish session-based defaults.

Open the Profile Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C I. The Profile Configuration window is shown in Figure 3-17.

Figure 3-17. Profile Configuration Window



Click a tab for configuring a profile.
Text fields with blue color are required.

Click the "Apply" button if you want to add this profile

Profile :

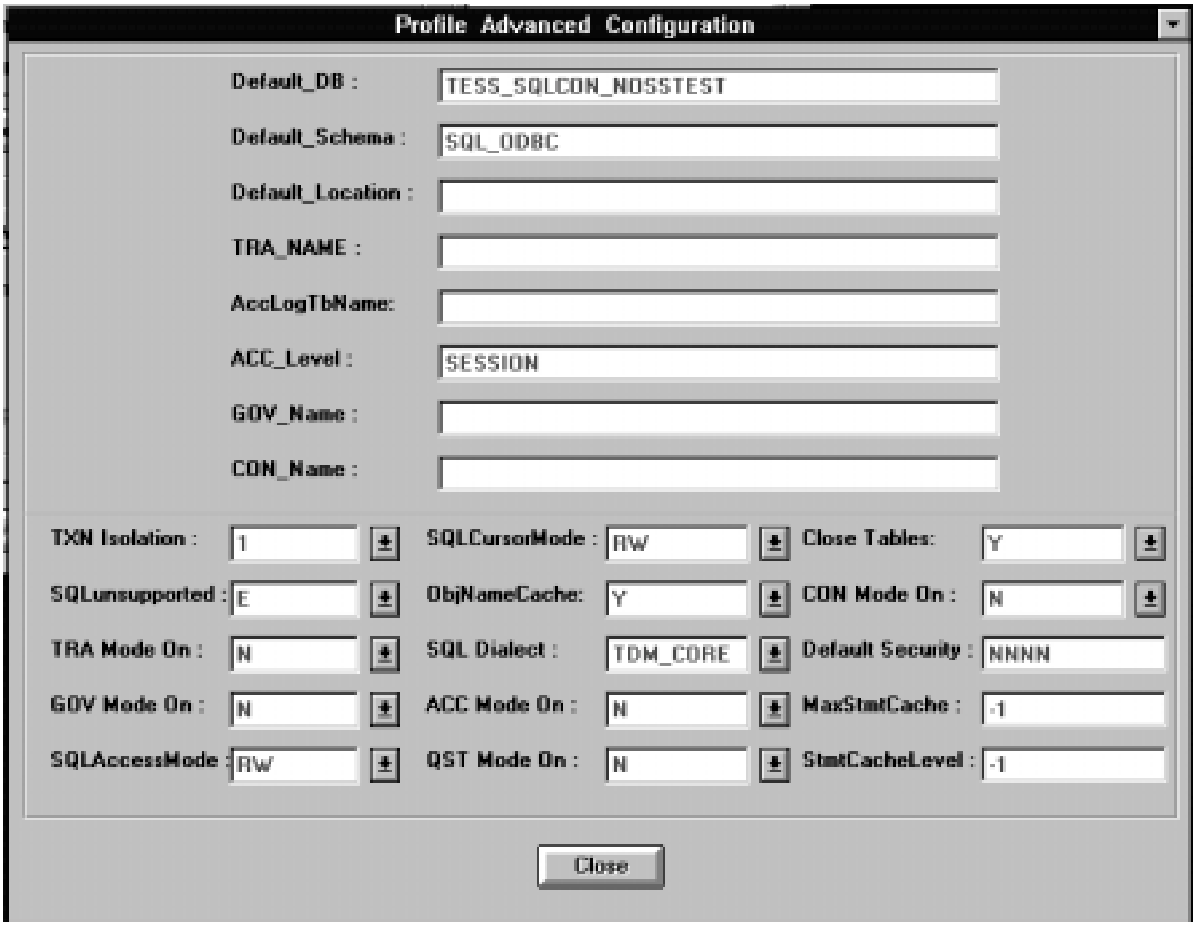
VST012.vsd

Only the profile name entry is required. Other entries are available in a subwindow.

Profile Advanced Configuration Subwindow

The Profile Configuration window provides a subwindow that allows you to make additional, optional configuration entries. Clicking the Advanced button opens that subwindow. The Profile Advanced Configuration subwindow is illustrated in Figure 3-18.

Figure 3-18. Profile Advanced Configuration Subwindow



Default_DB : TESS_SQLCON_NOSSTEST
Default_Schema : SQL_ODBC
Default_Location :
TRA_NAME :
AccLogTbName:
ACC_Level : SESSION
GOV_Name :
CON_Name :

TXN Isolation : 1
SQLunsupported : E
TRA Mode On : N
GOV Mode On : N
SQLAccessMode : RW

SQLCursorMode : RW
ObjNameCache: Y
SQL Dialect : TDM_CORE
ACC Mode On : N
QST Mode On : N

Close Tables: Y
CON Mode On : N
Default Security : NNNN
MaxStmtCache : -1
StmtCacheLevel : -1

VST013.vsd

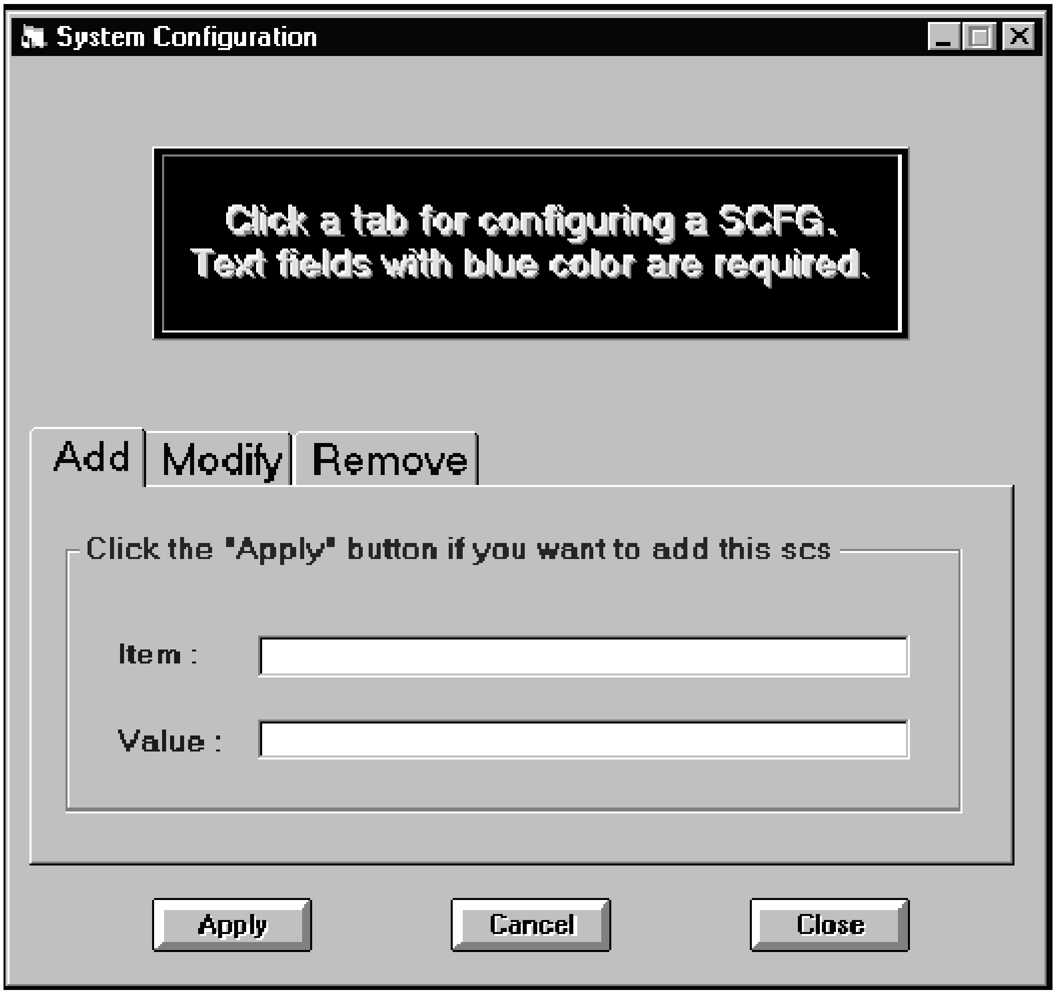
All entries in the subwindow are optional. If a system default exists for an entry, the default appears in the window when you first open it. Some entries have comboboxes to allow selection from a menu; alternatively, you can type in the entries.

System Configuration Window

Use the System Configuration window to configure the system configuration default values used by the NonStop ODBC server and the NOSUTIL process (and, indirectly through NOSUTIL, the SCS process).

Open the System Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C F. The System Configuration window is shown in Figure 3-19.

Figure 3-19. System Configuration Window



Click a tab for configuring a SCFG.
Text fields with blue color are required.

Click the "Apply" button if you want to add this scs

Item :

Value :

VST014.vsd

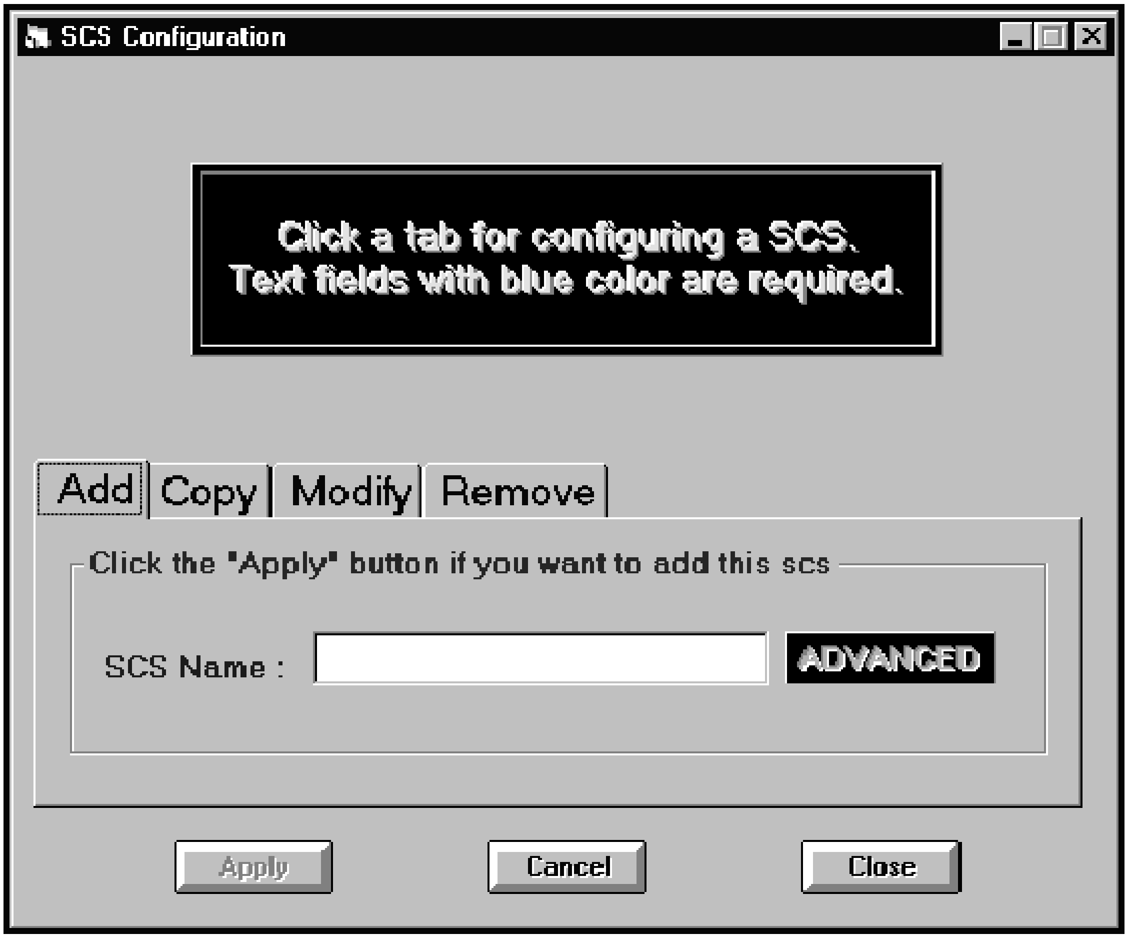
The System Configuration window supports adding, modifying, and removing configuration entries, but the copy feature is not available with this window.

SCS Configuration Window

Use the SCS Configuration window to configure the configuration definitions for the SQL Communication Subsystem.

Open the SCS Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C S. The SCS Configuration window is shown in Figure 3-20.

Figure 3-20. SCS Configuration Window



Click a tab for configuring a SCS.
Text fields with blue color are required.

Click the "Apply" button if you want to add this scs

SCS Name :

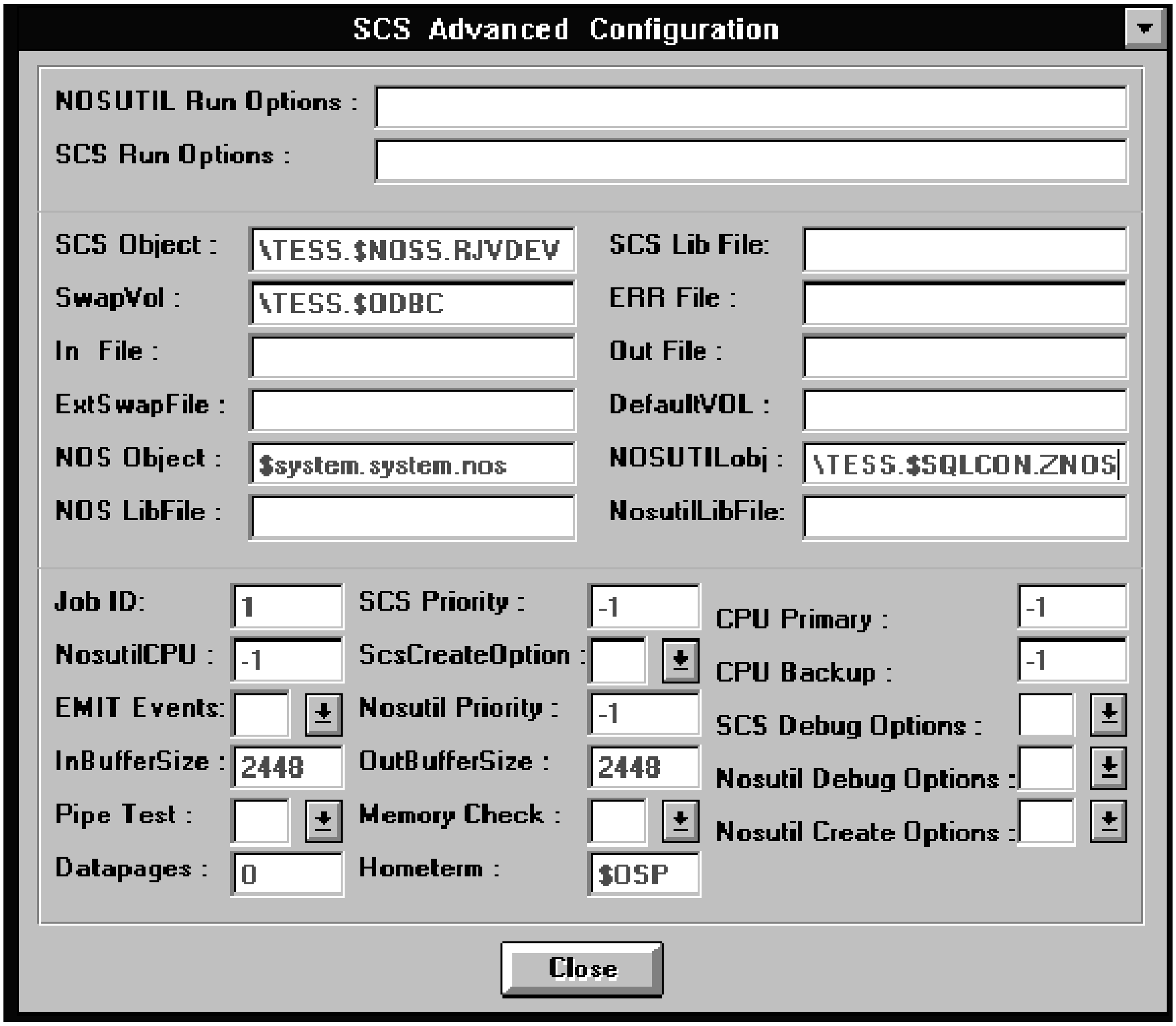
VST015.vsd

Only the name of the SQL Communication Subsystem to be configured is required. Other fields can be entered in a subwindow.

SCS Advanced Configuration Subwindow

The SCS Configuration window provides a subwindow that allows you to make additional, optional configuration entries. Clicking the Advanced button opens that subwindow. The SCS Advanced Configuration subwindow is illustrated in Figure 3-21.

Figure 3-21. SCS Advanced Configuration Subwindow



SCS Advanced Configuration

NOSUTIL Run Options :
SCS Run Options :

SCS Object : \TESS.$NOSS.RJVDEV
SCS Lib File:
SwapVol : \TESS.$ODBC
ERR File :
In File :
Out File :
ExtSwapFile :
DefaultVOL :
NOS Object : $system.system.nos
NOSUTILobj : \TESS.$SQLCON.ZNOS
NOS LibFile :
NosutilLibFile:

Job ID: 1
SCS Priority : -1
CPU Primary : -1
NosutilCPU : -1
ScsCreateOption :
CPU Backup : -1
EMIT Events:
Nosutil Priority : -1
SCS Debug Options :
InBufferSize : 2448
OutBufferSize : 2448
Nosutil Debug Options :
Pipe Test :
Memory Check :
Nosutil Create Options :
Datapages : 0
Hometerm : $OSP

VST016.vsd

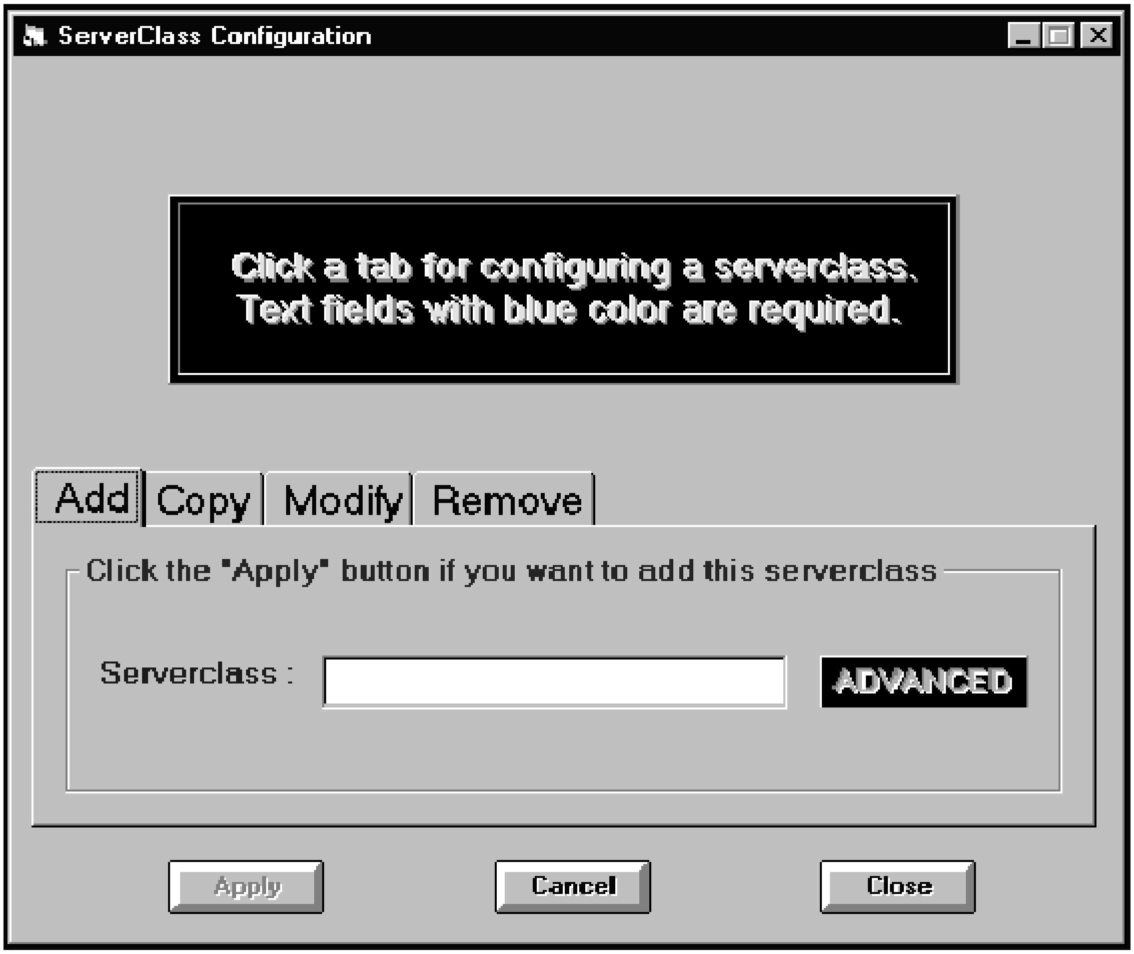
All entries in the subwindow are optional. If a system default exists for an entry, the default appears in the window when you first open it.

ServerClass Configuration Window

Use the ServerClass Configuration window to configure server class definitions.

Open the ServerClass Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C V. The ServerClass Configuration window is shown in Figure 3-22.

Figure 3-22. ServerClass Configuration Window



ServerClass Configuration

Click a tab for configuring a serverclass.
Text fields with blue color are required.

Click the "Apply" button if you want to add this serverclass

Serverclass :

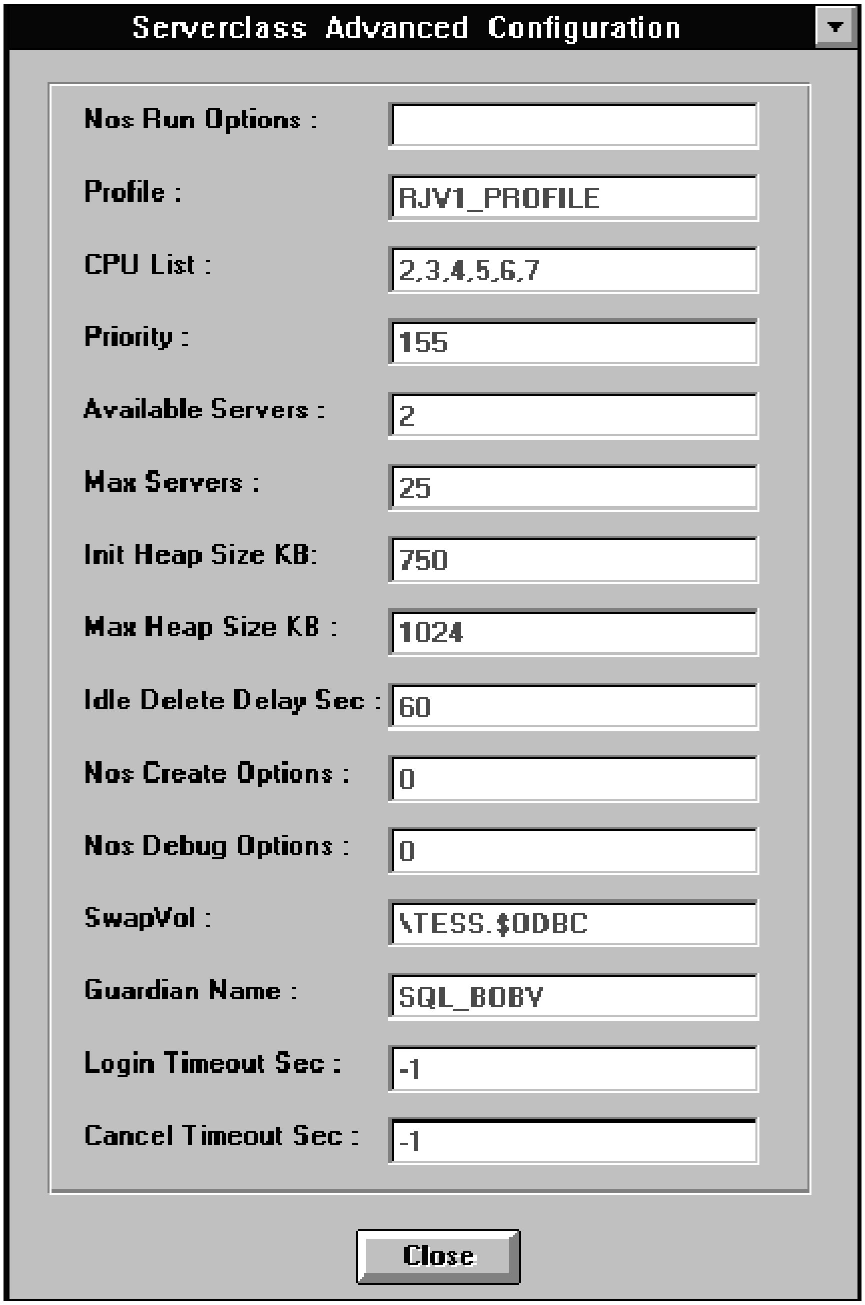
VST017.vsd

Only the name of the server class to be configured is required. Other fields can be entered in a subwindow.

ServerClass Advanced Configuration Subwindow

The ServerClass Configuration window provides a subwindow that allows you to make additional, optional configuration entries. Clicking the Advanced button opens that subwindow. The ServerClass Advanced Configuration subwindow is illustrated in Figure 3-18.

Figure 3-23. ServerClass Advanced Configuration Subwindow



Serverclass Advanced Configuration

Nos Run Options :	
Profile :	RJV1_PROFILE
CPU List :	2,3,4,5,6,7
Priority :	155
Available Servers :	2
Max Servers :	25
Init Heap Size KB:	750
Max Heap Size KB :	1024
Idle Delete Delay Sec :	60
Nos Create Options :	0
Nos Debug Options :	0
SwapVol :	\TESS.$ODBC
Guardian Name :	SQL_BOBY
Login Timeout Sec :	-1
Cancel Timeout Sec :	-1

VST018.vsd

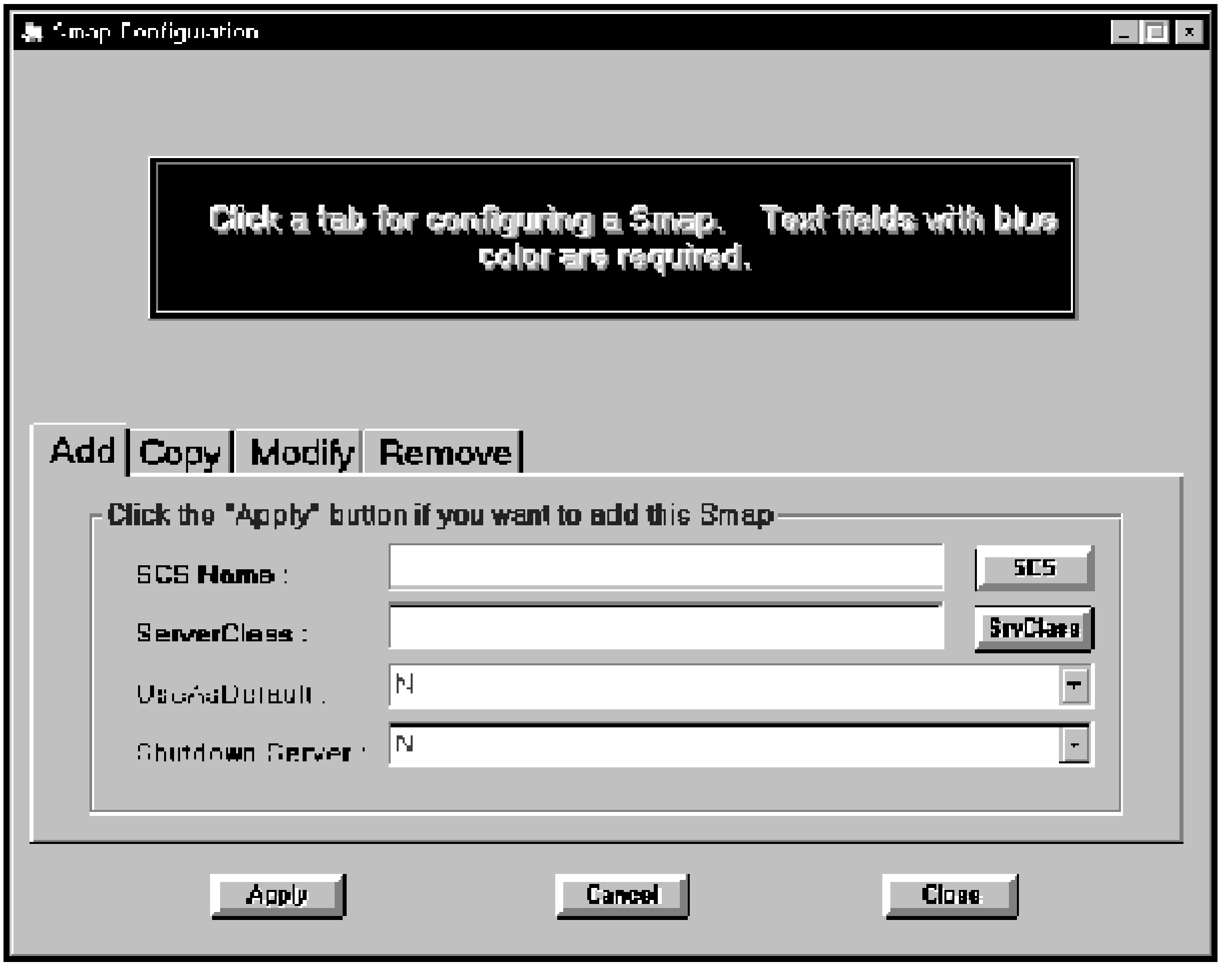
All entries in the subwindow are optional. If a system default exists for an entry, the default appears in the window when you first open it.

Smap Configuration Window

Use the Smap Configuration window to set up the association between a server class description and an SCS description.

Open the Smap Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C M. The Smap Configuration window is shown in Figure 3-24.

Figure 3-24. Smap Configuration Window



Click a tab for configuring a Smap. Text fields with blue color are required.

Click the "Apply" button if you want to add this Smap

SCS Name :

ServerClass :

UseAsDefault : N

Shutdown Server : N

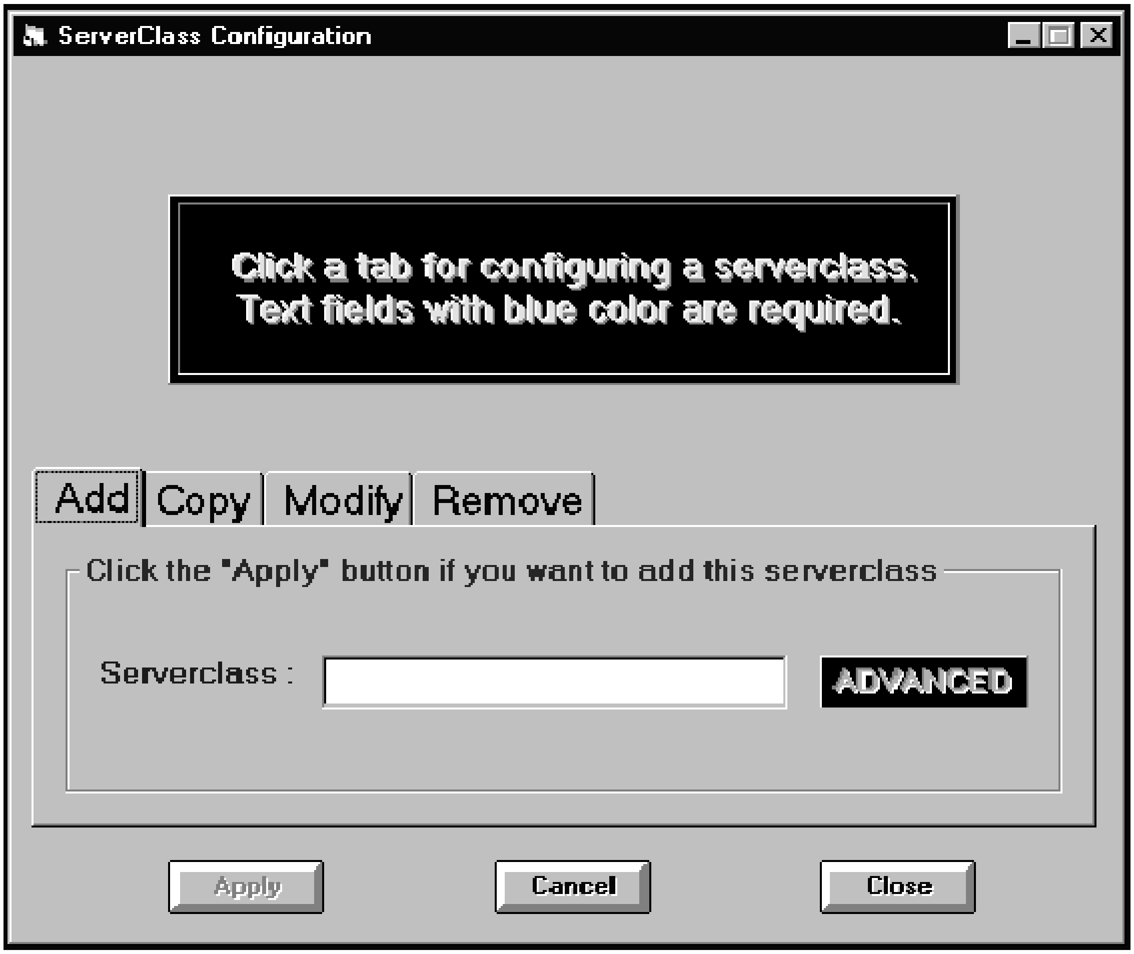
Because the Smap Configuration window is associated with a SCS/server class relationship, you can do a compound-key search on both entities when using the search feature.

ServerClass Configuration Window

Use the ServerClass Configuration window to configure server class definitions.

Open the ServerClass Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C V. The ServerClass Configuration window is shown in Figure 3-25.

Figure 3-25. ServerClass Configuration Window



Click a tab for configuring a serverclass.
Text fields with blue color are required.

Click the "Apply" button if you want to add this serverclass

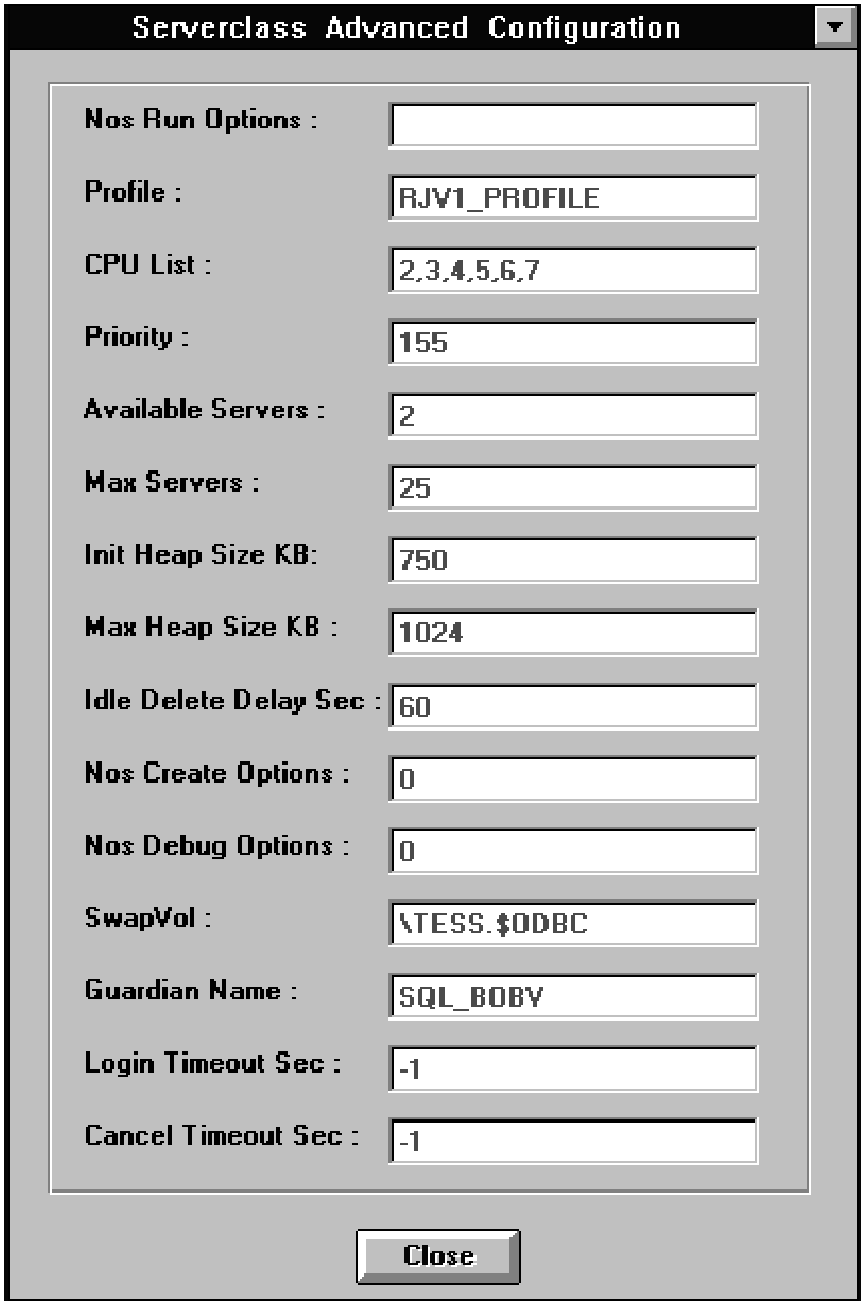
VST017.vsd

Only the name of the server class to be configured is required. Other fields can be entered in a subwindow.

ServerClass Advanced Configuration Subwindow

The ServerClass Configuration window provides a subwindow that allows you to make additional, optional configuration entries. Clicking the Advanced button opens that subwindow. The ServerClass Advanced Configuration subwindow is illustrated in Figure 3-26.

Figure 3-26. ServerClass Advanced Configuration Subwindow



Serverclass Advanced Configuration

Nos Run Options :

Profile : RJV1_PROFILE

CPU List : 2,3,4,5,6,7

Priority : 155

Available Servers : 2

Max Servers : 25

Init Heap Size KB: 750

Max Heap Size KB : 1024

Idle Delete Delay Sec : 60

Nos Create Options : 0

Nos Debug Options : 0

SwapVol : \TESS.$ODBC

Guardian Name : SQL_BOBY

Login Timeout Sec : -1

Cancel Timeout Sec : -1

VST018.vsd

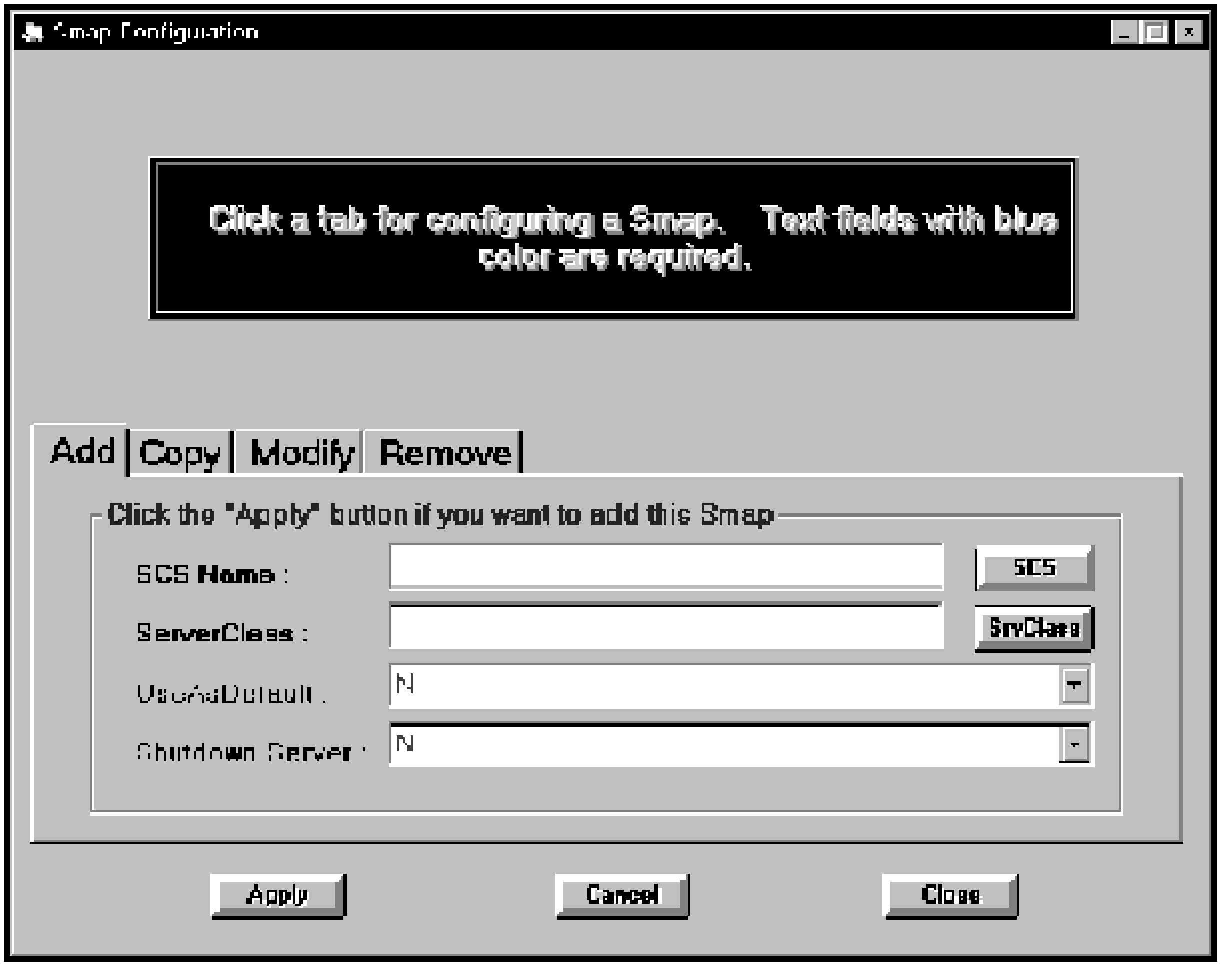
All entries in the subwindow are optional. If a system default exists for an entry, the default appears in the window when you first open it.

Smap Configuration Window

Use the Smap Configuration window to set up the association between a server class description and an SCS description.

Open the Smap Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C M. The Smap Configuration window is shown in Figure 3-27.

Figure 3-27. Smap Configuration Window



Click a tab for configuring a Smap. Text fields with blue color are required.

Click the "Apply" button if you want to add this Smap

SCS Name :

ServerClass :

UseAsDefault : N

Shutdown Server : N

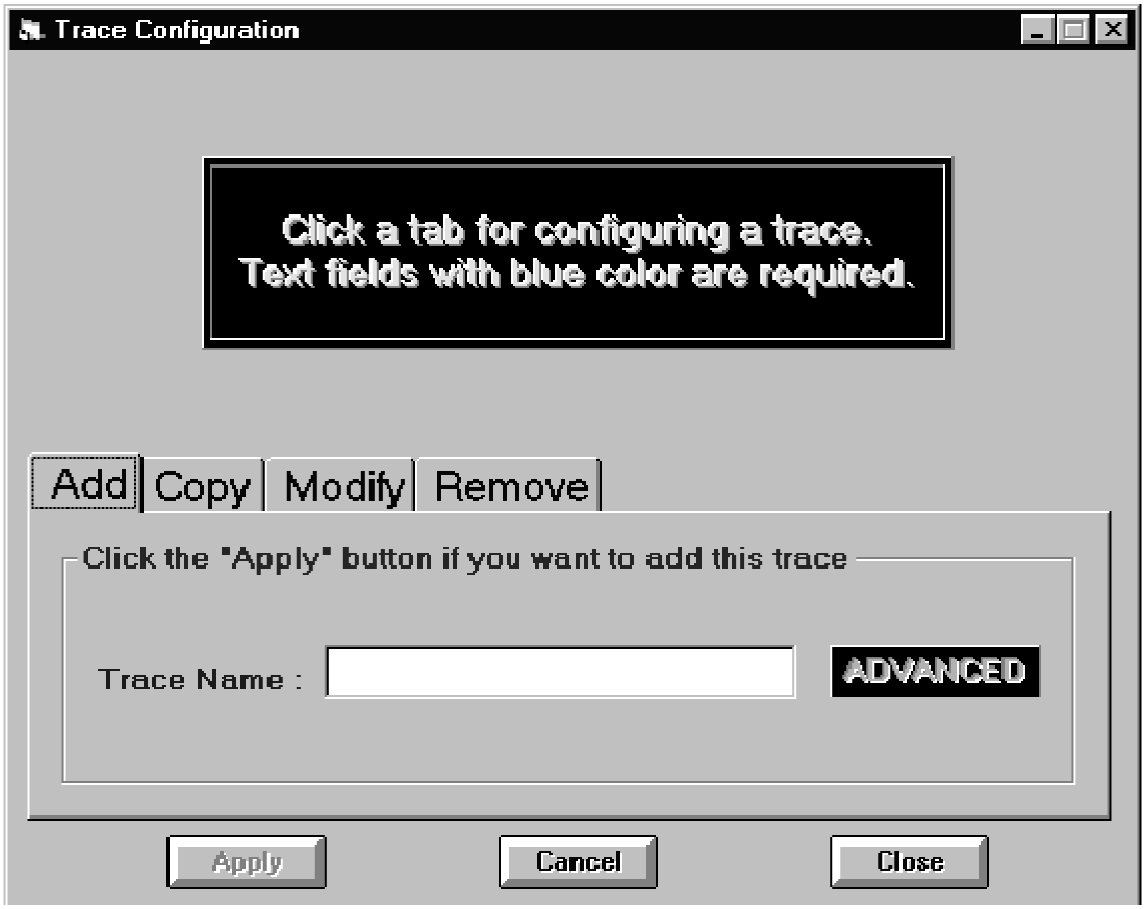
Because the Smap Configuration window is associated with a SCS/server class relationship, you can do a compound-key search on both entities when using the search feature.

Trace Configuration Window

Use the Trace Configuration window to set up trace definitions, which contain information needed by the NonStop ODBC Server trace facility.

Open the Trace Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C T. The Trace Configuration window is shown in Figure 3-28.

Figure 3-28. Trace Configuration Window



Trace Configuration

Click a tab for configuring a trace.
Text fields with blue color are required.

Click the "Apply" button if you want to add this trace

Trace Name :

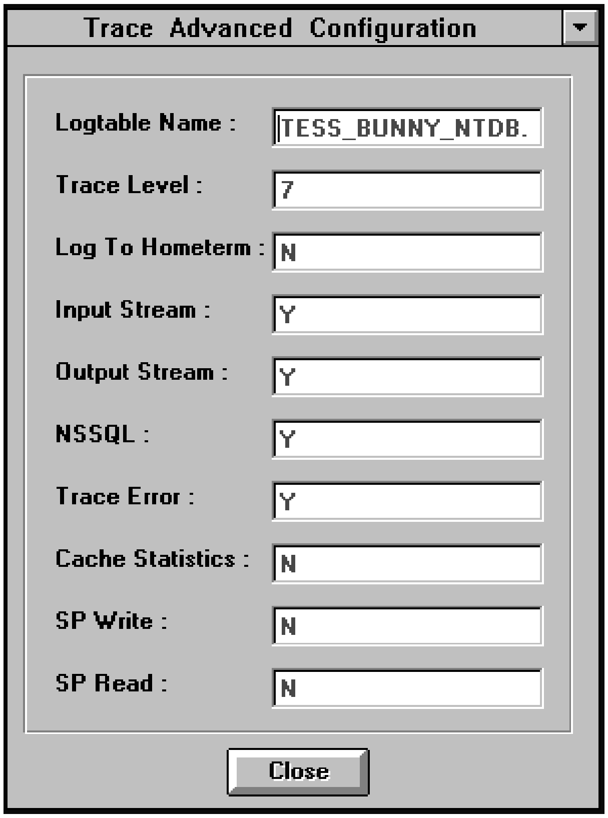
VST020.vsd

Only the trace name, a unique key, is required. Other fields can be entered in a subwindow.

Trace Advanced Configuration Subwindow

The Trace Configuration window provides a subwindow that allows you to make additional, optional configuration entries. Clicking the Advanced button opens that subwindow. The Trace Advanced Configuration subwindow is illustrated in Figure 3-29.

Figure 3-29. Trace Advanced Configuration Subwindow



Trace Advanced Configuration

Logtable Name : TESS_BUNNY_NTDB.
Trace Level : 7
Log To Hometerm : N
Input Stream : Y
Output Stream : Y
NSSQL : Y
Trace Error : Y
Cache Statistics : N
SP Write : N
SP Read : N

VST021.vsd

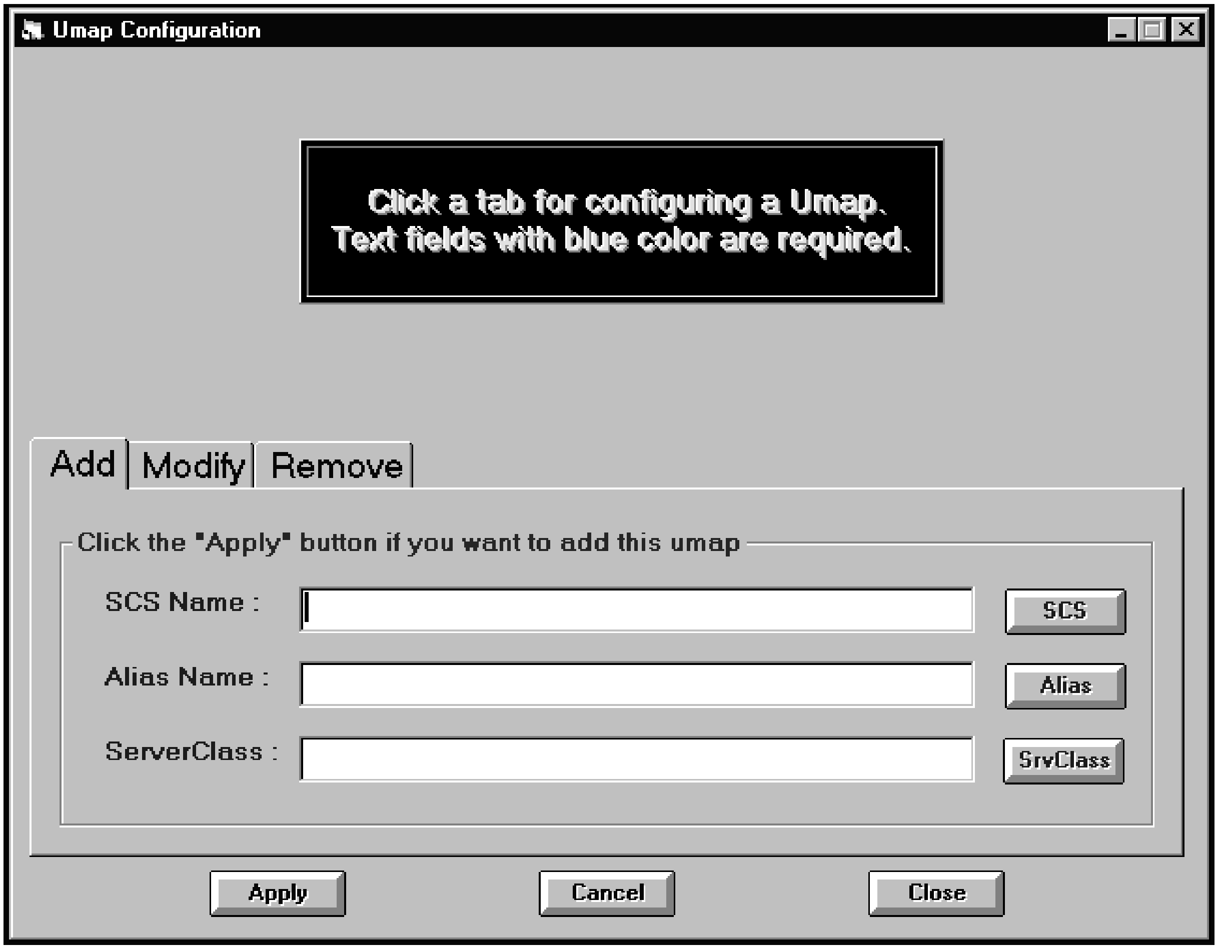
All entries in the subwindow are optional. If a system default exists for an entry, the default appears in the window when you first open it.

Umap Configuration Window

Use the Umap Configuration window to set up the mapping between user/alias names and server classes for a given SQL Communication Subsystem.

Open the Umap Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C P. The Umap Configuration window is shown in Figure 3-30.

Figure 3-30. Umap Configuration Window



Umap Configuration

Click a tab for configuring a Umap.
Text fields with blue color are required.

Click the "Apply" button if you want to add this umap

SCS Name :

Alias Name :

ServerClass :

VST022.vsd

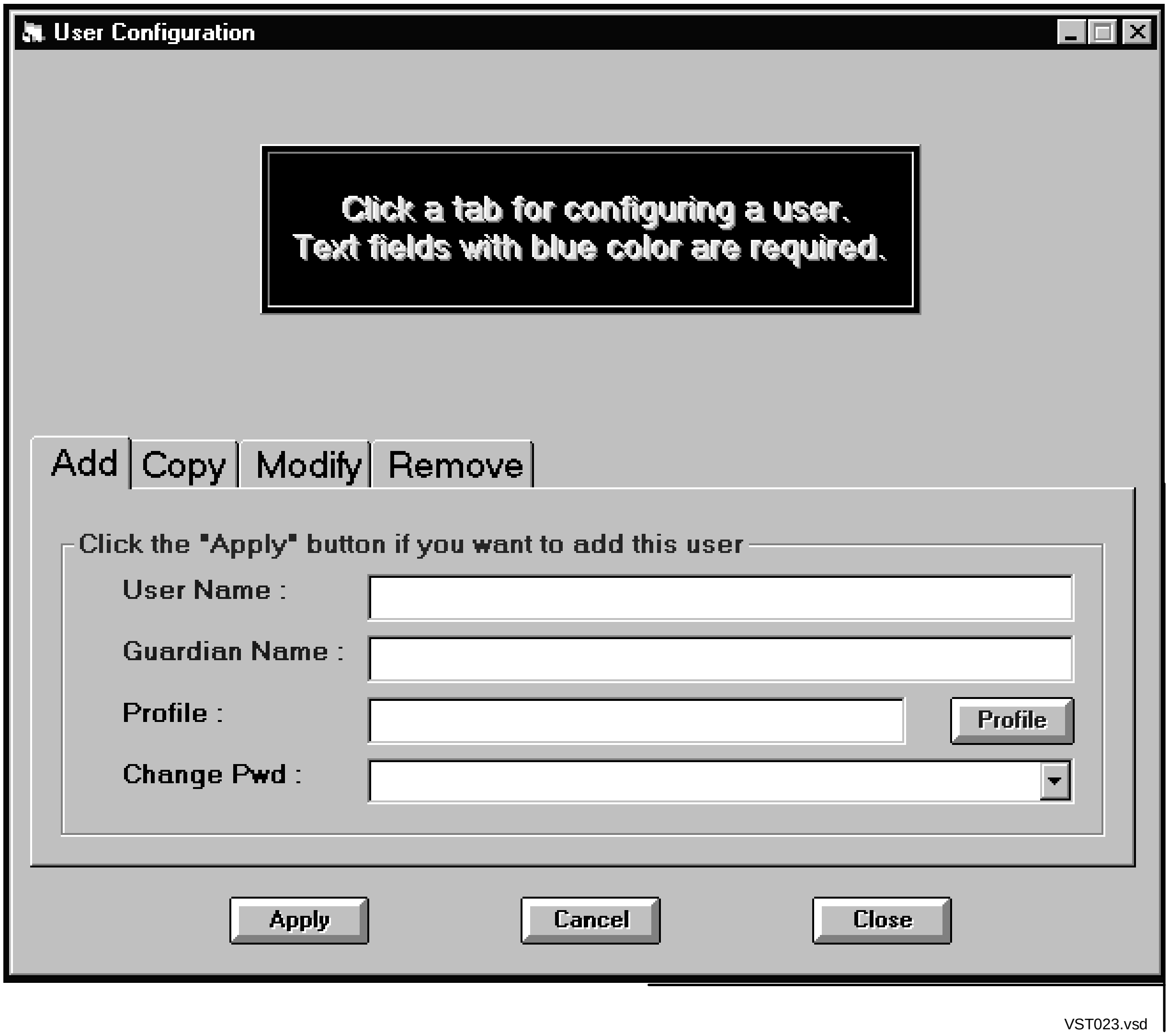
Because the Umap Configuration window is associated with an SCS/alias name relationship, you can do a compound-key search on both entities when using the search feature.

User Configuration Window

Use the User Configuration window to set up usernames.

Open the User Configuration window by choosing it from the Configuration menu with the mouse pointer or by using the keyboard equivalent ALT C U. The User Configuration window is shown in Figure 3-31.

Figure 3-31. User Configuration Window



User Configuration

Click a tab for configuring a user.
Text fields with blue color are required.

Click the "Apply" button if you want to add this user

User Name :

Guardian Name :

Profile :

Change Pwd :

VST023.vsd

Help Screens

You can open a help file by choosing it from the Help menu with the mouse pointer or by using the keyboard equivalent ALT H H. The contents of the help file that pertain to a selected topic are displayed on the monitor.

Configuration Sequence

To avoid receiving warning messages during the configuration process, you should use the NonStop ODBC Server Configuration Manager screens in the proper sequence so that no entity is configured before another entity upon which it depends for definition. For example, you should create a user profile before you reference it when adding a username.

The optimum screen sequence for configuration is illustrated in the sample configuration session that follows.

A Sample Configuration Session

This sample configuration session with the NonStop ODBC Server Configuration Manager, presented with step-by-step instructions, is provided with the intent to help the database administrator or NonStop ODBC Server administrator configure the system. It gives the administrator a sense of how to set up the new system catalog with the SCS, server class, profile, and username definitions. This sample session also illustrates the best sequence in which to configure the various NonStop ODBC Server entities to minimize warning messages that can result from unresolved dependencies.

This example is based on the assumptions that the product (client and server) has been installed according to the instructions in Section 2, “Installing the NonStop ODBC Server,” and the system catalog has been installed.

The first SCS a database administrator configures, during the installation phase, has only a default server class. The maximum number of servers is three. After this SCS has been configured, you can use the NonStop ODBC Server Configuration Manager to configure the new SCS and server classes.

Using the status command,

```
NOSCOM> status scs $dba
```

you can see that you have the SCS and one server running. You could also have RUNFILTR running and observe the EMS messages.

This sample configuration session includes the following steps:

Preliminary Steps

On the client side, configure a datasource, ODBCDBA, from the ODBC administrator. Enter a host name using the NET_NAME (from the ADD NET_SERVICE statement) DBAADMIN for NETBIOS, or ODBCIPX004 for SPX/IPX, and the IP Address and port number (corresponding to ODBCTCP006) for TCP/IP, and the database name MASTER.

Launch the NonStop ODBC Server Configuration Manager by double-clicking the left mouse button on the “conftool” icon.

Choose Connect from the session menu to reach the logon screen. Enter the datasource name you just configured and specify the user ID as DBO; the password should be the super ID password on your system. Click the OK button. You will receive a “successful logon” message.

Now you are ready to configure new a SCS and server classes, profiles, and users with the NonStop ODBC Server Configuration Manager.

The following is an example of configuring one SCS to support a maximum of 25 clients in five different server classes, five users per server class. At this time, you should have five different legitimate Guardian users, such as:

```
SOFTWARE.CORE1
SOFTWARE.CORE2
SOFTWARE.CORE3
SOFTWARE.CORE4
SOFTWARE.CORE5
```

From the menu bar in the NonStop ODBC Server Configuration Manager main window, choose the Configuration menu to display the list of configuration windows. You will be performing these steps:

- Configure Profiles
- Add Users
- Add the Net Service Configuration
- Add the SCS Configuration
- Configure the Server Classes
- Add Smap configuration
- Add Umap configuration

Step One: Configure Profiles

Configuration begins with the definition of one or more profiles, as follows:

1. Choose Profile from the Configuration menu in the main window.
2. Click the Add control tab in the Profile Configuration window.
    a. Type PROFILE_CORE1 in the Profile field.
    b. Click the Advanced button to open the Profile Advanced Configuration subwindow.
    c. Enter values for the following fields only (defaults will be used for all other values):

Default_DB:	MASTER
SQL Dialect:	TDM_CORE

    d. Click the Close button on the Profile Advanced Configuration subwindow.
    e. Click the Apply button.
    f. Click NO for the Update System Configuration dialog.
    g. Click OK for the SQL SUCCESS message.
3. Click the Add control tab again.
    a. Type PROFILE_CORE2 in the Profile field.
    b. Click the Advanced button.

c. Enter the following values for the specified fields (defaults will be used for other fields except as noted below):

Default_DB	MASTER (or the name of any customized database)
Default_Schema	SOFTWARE_CORE1
Default_Location	$*vol* (a valid volume on your system)
SQLAccessMode	RW or RO
MaxStmtCache	100

You can also give values to TRA_NAME, AccLogTbName, GOV_Name, and CON_Name if you want to; you should do so before proceeding to the next item.

d. Click the Close button on the Profile Advanced Configuration subwindow.

e. Click the Apply button.

f. Click NO for the Update System Configuration dialog.

g. Click OK for the SQL SUCCESS message.

4. Click the Copy control tab.

   a. Click the Search button.

   b. Double-click on PROFILE_CORE2 in the list box.

   c. Type PROFILE_CORE3 in the Profile field; let all the other attributes remain the same as for PROFILE_CORE2. You can modify any of the attributes after you have set up the system.

   d. Click the Apply button.

   e. Click NO for the Update System Configuration dialog.

   f. Click OK for the SQL SUCCESS message.

5. Double-click on PROFILE_CORE3 in the list box and type PROFILE_CORE4 in the Profile field.

   a. Click the Apply button.

   b. Click NO for the Update System Configuration dialog.

   c. Click OK for the SQL SUCCESS message.

6. Repeat all of Step 5, but type PROFILE_CORE5 in the Profile field.

7. Close the Profile Configuration window.

Step Two: Add Users

The next step is to map logical usernames and Guardian user names to each other and to the profiles.

1. Choose User from the Configuration menu in the main window.
2. Click the Add control tab in the User Configuration window.
    a. Type SOFTWARE_CORE1 in the User Name field,
    b. Type SOFTWARE.CORE1 in the Guardian Name field,
    c. Type PROFILE_CORE1 in the Profile field.
    d. Click the Apply button.
    e. Click NO for the Update System Configuration dialog.
    f. Click OK for the SQL SUCCESS message.
3. Repeat Step 2 using the following values:

User Name	SOFTWARE_CORE2
Guardian Name	SOFTWARE.CORE2
Profile	PROFILE_CORE2

4. Repeat Step 2 using the following values:

User Name	SOFTWARE_CORE3
Guardian Name	SOFTWARE.CORE3
Profile	PROFILE_CORE3

5. Repeat Step 2 using the following values:

User Name	SOFTWARE_CORE4
Guardian Name	SOFTWARE.CORE4
Profile	PROFILE_CORE4

6. Repeat Step 2 using the following values:

User Name	SOFTWARE_CORE5
Guardian Name	SOFTWARE.CORE5
Profile	PROFILE_CORE5

Close the User Configuration window.

Step Three: Add the Net Service Configuration

The next step is to configure the network service.

Note. Although the steps in this configuration session are ordered in the optimum sequence to minimize warning messages for integrity violations, you will receive such a warning when configuring the net service because of unavoidable interdependencies. You can safely ignore the warning message.

If you are using TCP/IP or SPX/IPX, configure your net service as follows (the information needed for these attributes can be found in existing Tedit SCS configuration files such as SCSCFGT1):

1. Choose Net Service from the Configuration menu in the main window.
2. Click the Add control tab in the Network Service Configuration window.
3. Type in the following values for the specified fields:

SCS Name	$SCSN
Net Protocol	TCP/IP or SPX/IPX
Net Name	ODBCTCP008 or ODBCIPX006 (exactly as it appears in the services file)
IOP Name	$ZTC0 (if conventional TCP/IP) or NV1
	<parallel tcp/ip process name> (if parallel TCP/IP)

4. Click the Protocol button.
5. Click the TCP/IP control tab in the Network Protocol Configuration window.
6. Type in the following values for the specified fields:

ServiceFileName	$*vol.subvol.services* (the location of the services file)
Keep Alive	0 (zero)
Oob In Line	0 (zero)
Linger	0 (zero)
Re Use Addr	0 (zero)

7. Click the Close button in the Network Protocol Configuration window.
8. Click the Apply button.
9. Click NO for the Update System Configuration dialog.
10. Click OK for the SQL SUCCESS message.
11. Click the Close button in the Network Service Configuration window.

If you are using NETBIOS, configure your net service as follows (the information needed for these attributes can be found in existing Tedit SCS configuration files such as SCSCFGT1):

1. Choose Net Service from the Configuration menu in the main window.
2. Click the Add control tab in the Network Service Configuration window.

3. Type in the following values for the specified fields:

SCS Name	$SCSN
Net Protocol	NETBIOS
Net Name	ODBCSCSN
IOP Name	$LAN1

4. Click the Protocol button
5. Click the NetBIOS control tab in the Network Protocol Configuration window.
6. Type in the following values for the specified fields:

Mlan Domain	N.A.
Mlan Gateway	#GWN1
Mlan Adapter	0 (zero)
Net Qualifier	\PIPE\SQL\QUERY

7. Click the Close button in the Network Protocol Configuration window.
8. Click the Apply button.
9. Click NO for the Update System Configuration dialog.
10. Click OK for the SQL SUCCESS message.
11. Click the Close button in the Network Service Configuration window.

Step Four: Add the SCS Configuration

Next, you need to configure the SCS.

1. Choose SCS from the Configuration menu in the main window.
2. Click the Add control tab in the SCS Configuration window.
3. Type $SCSN in the SCS Name field (this name should be the same as the SCS name in the Net Service Configuration screen.)
4. Click the Advanced button.
5. Type the following values for the specified fields:

NOSUTIL Run Options	
SCS Run Options	
SCS Object	$*vol.subvol.*SCSOBJ (the fully qualified name of the SQL Communication Subsystem object program file)
SwapVol	$*vol* (a valid volume on your system)
In File	N.A.

ExtSwapFile	N.A.
NOS Object	$*vol.subvol.*ZNOS (the fully qualified name of the NonStop ODBC Server object program file)
NOS LibFile	N.A
SCS Lib File	N.A
ERR File	N.A
Out File	N.A
DefaultVOL	$*vol* (a valid volume on your system)
NOSUTILobj	$*vol.subvol.*NOSUTIL (the fully qualified name of the NOS utility object program file)
NOSUTILLibFile	N.A
Job ID	N.A
NOSUTILCPU	3 (any valid CPU number)
EMIT Events	Y or N
InBufferSize	2448
Pipe Test	N.A
Datapages	4096
SCS Priority	160
ScsCreateOption	6, 22, or 23
NOSUTIL Priority	155
OutBufferSize	2448
Memory Check	Y or N
Hometerm	any valid terminal
CPU Primary	2
CPU Backup	3
SCS Debug Options	0, 8, 9, 12, or 13
NOSUTIL Debug Options	0, 10, 11, 13, or 14
NOSUTIL Create Options	6, 22, or 23

6. Click the Close button in the SCS Advanced Configuration window.

7. Click the Apply button.

8. Click NO for the Update System Configuration dialog.

9. Click OK for the SQL SUCCESS message.
10. Click the Close button in the SCS Configuration window.

Step Five: Configure the Server Classes

The next step is to configure the various attributes of the server classes.

1. Choose ServerClass from the Configuration menu in the main window.
2. Click the Add control tab in the ServerClass Configuration window.
3. Type CLASSCORE1 in the Serverclass field.
4. Click the Advanced button.
5. Type the following values for the specified fields:

NOS Run Options	N.A
Profile	PROFILE_CORE1
CPU List	0,1,2,3,4 (a valid list of CPUs on your system)
Priority	157
Available Servers	1
Max Servers	5
Init Heap Size KB	750
Max Heap Size KB	1024
Idle Delete Delay Sec	60
NOS Create Options	N.A
NOS Debug Options	N.A
Swapvol	$*vol* (a valid volume name on your system)
Guardian Name	SOFTWARE.CORE1
Login Timeout Sec	N.A
Cancel Timeout Sec	N.A

6. Click the Close button in the Serverclass Advanced Configuration window.
7. Click the Apply button.
8. Click NO for the Update System Configuration dialog.
9. Click OK for the SQL SUCCESS message.

To configure the other server class, you can copy the definition of the server class you just added and change the attributes as needed.

1. Click the Copy control tab in the Serverclass Configuration window.

2. Click the Search button.
3. Double-click on CLASSCORE1 in the list box.
4. Type CLASSCORE2 in the Serverclass field.
5. Click the Advanced button.
6. Change the Profile to PROFILE_CORE2.
7. Change the Guardian Name to SOFTWARE.CORE2.

   All other attributes can remain the same as they were defined for server class CLASSCORE1. You can modify them later, fine-tuning them to suit the needs of the users.
8. Click the Close button in the ServerClass Advanced Configuration window.
9. Click the Apply button.
10. Click NO for the Update System Configuration dialog.
11. Click OK for the SQL SUCCESS message.
12. Repeat Steps 1 through 11, using the following values:

Serverclass	CLASSCORE3
Profile	PROFILE_CORE3
Guardian Name	SOFTWARE.CORE3

13. Again repeat Steps 1 through 11, using the following values:

Serverclass	CLASSCORE4
Profile	PROFILE_CORE4
Guardian Name	SOFTWARE.CORE4

14. Again repeat Steps 1 through 11, using the following values:

Serverclass	CLASSCORE5
Profile	PROFILE_CORE5
Guardian Name	SOFTWARE.CORE5

15. Click the Close button in the ServerClass Configuration window.

You now have five server class definitions. You can modify any of the attributes while the system is running. The servers respond to any changes you make when CHECK_INTERVAL_SECS in the system configuration table has expired and the client disconnects.

Step Six: Add Smap Configuration

At this time you need to map the server class definitions to the SCS.

1. Choose Smap from the Configuration menu of the main window.
2. Click the Add control tab in the Smap Configuration window.
3. Type the following values for the specified fields:

SCS Name	$SCSN
ServerClass	CLASSCORE1
UseAsDefault	N
Shutdown Server	N

4. Click the Apply button.
5. Click NO for the Update System Configuration dialog.
6. Click OK for the SQL SUCCESS message.

   Repeat Steps 2 through 5 for server classes CLASSCORE2, CLASSCORE3, CLASSCORE4, and CLASSCORE5, using the same SCS name, $SCSN, and the same UseAsDefault entry, N.

7. Click the Close button in the Smap Configuration window.

If you need a default server class, you can modify the Smap configuration and change UseAsDefault to “Y” for any one of the defined server classes. Only one server class can be named as the default server class for a given SCS.

Step Seven: Add Umap Configuration

The last step in the basic configuration is to map users to an SCS and server class.

1. Choose Umap from the Configuration menu of the main window.
2. Click the Add control tab in the Umap Configuration window.
3. Enter the following values for the specified fields:

SCS Name	$SCSN
Alias Name	SOFTWARE_CORE1
ServerClass	CLASSCORE1

4. Click the Apply button.
5. Click NO for the Update System Configuration dialog.
6. Click OK for the SQL SUCCESS message.
7. Repeat Steps 2 through 6, using the following values:

SCS Name	$SCSN
Alias Name	SOFTWARE_CORE2
ServerClass	CLASSCORE2

8. Repeat Steps 2 through 6, using the following values:

SCS Name	$SCSN
Alias Name	SOFTWARE_CORE3
ServerClass	CLASSCORE3

9. Repeat Steps 2 through 6, using the following values:

SCS Name	$SCSN
Alias Name	SOFTWARE_CORE4
ServerClass	CLASSCORE4

10. Repeat Steps 2 through 4, using the following values:

SCS Name	$SCSN
Alias Name	SOFTWARE_CORE5
ServerClass	CLASSCORE5

11. Click YES for the Update System Configuration dialog.
12. Click OK for the SQL SUCCESS message.
13. Click the Close button for the Umap Configuration screen.

Starting and Verifying the SCS

To start the newly configured SQL Communication Subsystem $SCSN from ODBCTest, do the following:

1. Connect to the datasource ODBCDBA.
2. Execute the pass-through statement:

   ```
   SELECT "TDM: UTIL START SCS $SCSN"
   ```

   to start the SQL Communication Subsystem named $SCSN.
3. Execute the pass-through statement:

   ```
   SELECT "TDM: UTIL STATUS SCS $SCSN"
   ```

   to obtain the status for $SCSN. If the process is running, you will receive a message stating that fact.

On the HP side, you can issue SCF commands to find the status of the SCS and the server classes.

At this time, you might want to perform additional, optional configurations, such as adding an alias name for user SOFTWARE_CORE1 and mapping that alias to a server class and the SCS, and adding tracing, resource governing, and resource accounting specifications to the configuration.

If, for a particular profile, you want to turn on tracing, resource accounting, or resource governing, you should add those configurations before you add or modify the profile to prevent integrity violations.

Optional: Add an Alias

This example adds an alias name, CORE1, for the user SOFTWARE_CORE1.

1. Choose Alias from the Configuration menu of the main window.
2. Click the Add control tab in the Alias Configuration window.
3. Type the following values for the specified fields:

Alias Name	CORE1
User Name	SOFTWARE_CORE1
Profile	PROFILE_CORE1

4. Click the Apply button.
5. Click the NO button for the Update System Configuration dialog.
6. Click OK for the SQL SUCCESS message.
7. Click the Close button for the Alias Configuration window.

Optional: Map the Alias

You need to add a Umap entry to map the alias CORE1 to server class CLASSCORE1 and SCS $SCSN.

1. Choose Umap from the Configuration menu of the main window.
2. Click the Add control tab in the Umap Configuration window.
3. Type the following values for the specified fields:

SCS Name	$SCSN
Alias Name	CORE1
ServerClass	CLASSCORE1

4. Click the Apply button.
5. Click YES for the Update System Configuration dialog.
6. Click OK for the SQL SUCCESS message.

Optional: Add Trace Functionality

To add trace functionality, you need to configure the trace log table, modify the profile to register that table, and configure the query status log table.

Define the Trace Log Table

From ODBCTest, do the following:

1. Connect to datasource ODBCDBA.
2. Execute the pass-through statement:

   ```
   select "tdm: util create tra_log in database.owner.nostrace"
   ```

   database is as in ZNSDB, *owner* is a logical username, and *nostrace* is just an example of a trace table name, which can be any table name.

In the NonStop ODBC Server Configuration Manager, do the following:

1. Choose Trace from the Configuration menu of the main window.
2. Click the Add control tab in the Trace Configuration window.
3. Type TRACE1 in the Trace Name field.
4. Click the Advanced button.
5. Enter the following values for the specified fields:

Logtable Name	*database.owner.nostrace* (the name of the trace table you created with the pass-through statement)
Log To Hometerm	Y or N
Input Stream	Y or N
Output Stream	Y or N
NSSQL	Y or N
Trace Error	Y or N
Cache Statistics	Y or N
SP Write	Y or N
SP Read	Y or N

6. Click the close button in the Trace Advanced Configuration window.
7. Click the Apply button.
8. Click YES for the Update System Configuration dialog.
9. Click OK for the SQL SUCCESS message.

Modify the Profile

At this time you need to modify the profile entry for which you want to start the trace.

1. Choose Profile from the Configuration menu of the main window.
2. Click the Modify tab in the Profile Configuration window.
3. Click the Search button in the search box.
4. Double-click on the name of the profile you want to update in the list box.
5. Click the Advanced button.
6. Type the name of the trace log table in the TRA_NAME field.
7. Type Y in the TRA Mode On field.
8. Click the Close button in the Advanced Profile Configuration window.
9. Click Apply.
10. Click YES for the Update System Configuration dialog.
11. Click OK for the SQL SUCCESS message.

Add the Query Status Log Table

From ODBCTest, do the following:

1. Connect to datasource ODBCDBA.
2. Execute the pass-through statement:

   ```
   select "tdm: util create qst_log in database"
   ```

   The database name is that which appears in ZNSDB.

Optional: Add Resource Governing

To add resource governing, you add a governing policy and modify the profile to register that policy.

Add a Governing Policy

1. Choose Governing from the Configuration menu of the main window.
2. Click the Add control tab in the Governing Policy Configuration window.
3. Type the following values for the specified fields:

GOV Name	POLICYA
GOV Attribute	EXECUTION_TIME (or any other value described for this field in the *NonStop ODBC Server Reference Manual*.)

Limit Value	5000
GOV Action	PRIORITY ++10
Log Qst On	Y

4. Click the Apply button.
5. Click YES for the Update System Configuration dialog.
6. Click OK for the SQL SUCCESS message.

Modify the Profile

Now you will modify the Profile entry to register the Governing policy.

1. Choose Profile from the Configuration menu of the main window.
2. Click the Modify tab in the Profile window.
3. Click the Search button in the search box.
4. Double-click on the name of the profile you want to update in the list box.
5. Click the Advanced button.
6. Type POLICYA in the GOV_Name field.
7. Type Y in the GOV Mode On field.
8. Click the Close button in the Advanced Profile Configuration window.
9. Click Apply.
10. Click YES for the Update System Configuration dialog.
11. Click OK for the SQL SUCCESS message.

Optional: Add Resource Accounting

To add resource accounting, define an accounting log table in the database and modify the profile to register that table.

Add an Accounting Log Table

From ODBCTest, do the following:

1. Connect to datasource ODBCDBA.
2. Execute the pass-through statement:

   ```
   select "tdm: util create acc_log in database"
   ```

   The database name is that which appears in ZNSDB.

Modify the Profile

Now you will again modify the Profile entry.

1. Choose Profile from the Configuration menu of the main window.
2. Click the Modify tab in the Profile Configuration window.
3. Click the Search button in the search box.
4. Double-click on the name of the profile you want to update in the list box.
5. Click the Advanced button.
6. Type the name of the resource accounting log table in the AccLogTblName field.
7. Type either SESSION, to summarize activity for an entire client session, or STATEMENT, to log every SQL statement, in the ACC_Level field.
8. Type Y in the ACC Mode On field.
9. Click the Close button in the Advanced Profile Configuration window.
10. Click Apply.
11. Click YES for the Update System Configuration dialog.
12. Click OK for the SQL SUCCESS message.

4

Managing the NonStop ODBC Server

This section describes how to manage a NonStop ODBC Server system. You should read this section in conjunction with Section 2, Installing the NonStop ODBC Server.

The following topics are covered in this section:

- Configuring a server class and reading server class statistics.
- Using statement caching; how to improve your SQL execution performance and how to limit resource consumption by a query.
- Techniques for improving performance; the options available for configuring ODBC datasources.
- Security implications.
- Using Event Management Service (EMS) events from the SQL Communication Subsystem (SCS) and some EMS messages generated by the NonStop ODBC Server to diagnose problems based on EMS events.
- Using the Subsystem Control Facility (SCF) interface to check on the status of an SCS process.
- Using the trace features provided by the NonStop ODBC Server
- Developing user applications and scripts for reading the NonStop ODBC Server catalog tables.

Configuring and Managing SCS

This subsection describes the considerations, methods, and tools used to configure and adjust the SCS process for optimum operation.

Configuring SCS Server Classes

First, review the server class statistics available from SCF. Entering the command STATUS PROCESS $ZYX, where $ZYX represents an SCS process, yields the following display:

```
Process Name    State   Num of Sessions  Num of Classes    Accepting

$ZYX            STARTED               0               3    YES

                                    Connected            Listening
ServerClass              |     Total    High   Current|Current     Hit |
ZZTDMMAPPER              |         2       1         0|      1       2 |
CLASS1                   |         9       2         0|      5       5 |
CLASS2                   |         5       1         0|      6       5 |
```

The preceding display shows two major classes of statistics: statistics on connections and connection attempts (“Connected” column) and statistics on listening servers. These statistics provide useful information on processes and their activities.

Connected Total The total number of client connections received by $ZYX. This number is incremented for each client connection received.

Connected High The highest number of simultaneous client connections serviced by $ZYX.

Connected Current The number of clients currently connected to $ZYX.

Listening Current The total number of available servers in each class.

Listening Hit The number of times a client connection request could be satisfied from the pool of available servers.

To begin configuring a server class, analyze customer applications and usage, possibly determining that any given workstation runs multiple ODBC/DBLIB applications. Additionally, each application probably is used by different groups of users. It is impractical to address all applications and user groups as a whole.

Instead, break down the problem. First, segregate the applications. Then, within each application, segregate users into groups. Each group represents a Guardian user ID under which servers in a server class will run.

Now examine the problem relative to each application or user group. Each application or user group will become a NonStop ODBC Server class.

Additionally, when using “off-the-shelf” applications, you should assume that an active user can consume two connections to SCS and, therefore, two NonStop ODBC Servers. This rule is based on experience attempting to connect some “off-the-shelf” tools to the NonStop ODBC Server.

Example

Your analysis reveals that an application or user group will have 100 members. Of these, only 30 are expected to be connected and doing work at any given time. Also, you are not familiar with the off-the-shelf application software. In this case, it is good to suppose that each active station will require two connections to SCS and thus two servers. Finally, you expect two or three stations to attempt to connect simultaneously at any given time.

For each application/user group, do the following:

- Set AVAILABLE_SERVERS slightly higher than the expected number of simultaneous connection attempts. For this example, set it to 10 to be administrator. This allocation provides an available pool of 10 servers for those attempting to connect, allowing up to five stations to connect simultaneously. Set MAXSERVERS to 75, the worst-case scenario if everyone in the application/user group is connected and working.

- Run the application and analyze the server statistics. If you get complaints about slow connection time, you must increase the AVAILABLE_SERVERS value. If you get no complaints, begin to reduce the AVAILABLE_SERVERS value.

  The time required for a client to connect is the smallest when the value of TOTAL CONNECTED is equal to the value of LISTENING HIT. If these values are equal, there was always a listening server available when a client connection request arrived.

- To find the smallest AVAILABLE_SERVERS value, you must begin to reduce the value assigned to this parameter until the TOTAL CONNECTED values exceed the LISTENING HIT number. This scenario occurs when a client attempts to connect and must wait for an available server (either for SCS to start one or for another client to release one).

- When TOTAL CONNECTED values exceed LISTENING HIT values, increase AVAILABLE_SERVERS by increments of one until the values are equal.

  The result should yield a system that gives the fastest connect time with a minimum number of servers kept in the available pool for the server class.

- Next, check the HIGH CONNECTED value. This value represents the highest number of concurrent connections to the server class. In this example, it is likely that this value will be considerably less than 75. If, for example, you see a HIGH CONNECTED value of 50, meaning at most 50 clients were connected at one time, you can reduce MAXSERVERS from 75. Determine the new MAXSERVERS value with the following formula:

  ```
  MAXSERVERS = (HIGH-CONNECTED + (HIGH-CONNECTED * 10%) )
  ```

  The additional 10 percent should cover those unlikely situations in which usage is higher than normal and also allows you to add new users without immediately taking down the SQL Communication Subsystem (SCS).

Checking the Status of the NonStop ODBC Server System

The SCS product is released with a TACL macro for collecting status information about the NonStop ODBC Server system. To use this macro, you must first attach the TACL segment file used by the macro.

To attach the TACL segment file, enter the following command:

```
RUN $vol.znos.ATTACH $vol.znos
```

where $*vol.znos* is the volume and subvolume in which the NonStop ODBC Server software is installed. Having attached this TACL segment, you can then use the commands described in this subsection for managing the SCS and NonStop ODBC Server system processes.

After starting the NonStop ODBC Server system, check on all the processes running for that system.

If you are running NonStop ODBC Release 1, use the TDBSRV:STATUS command at the TACL prompt.

The syntax for this command is:

```
TDBSRV:STATUS SCS-config-file
```

where *SCS-config-file* is the name of the SCS configuration file used to bring up the system. This command generates a TACL status display of all of the processes running for that NonStop ODBC Server system. An example of the output from this command is shown in Section 2, Installing the NonStop ODBC Server.

If you are using Release 2, use the NOSCOM STATUS command.The syntax for this command is:

```
        NOSCOM STATUS SCS-name
```

SQL Statement Caching

Statement caching is useful in an environment where similar SQL statements are generated in the same session, or across sessions within the same NonStop ODBC Server class, and SQL compilation time is significant compared to SQL execution time. Using statement caching in such an environment improves SQL statement execution time.

When statement caching is enabled, the NonStop ODBC Server caches the compiled statements. When an equivalent statement is resubmitted, the SQL compilation step is skipped, and the statement is executed from the cache.

Configuring Statement Caching

Statement caching is disabled by default, so it must be configured to turn it on.

Statement caching can be configured in each user's profile by setting the SQL_MAX_STATEMENT_CACHE field to a number greater than zero (cache *n* statements). Note that the NonStop ODBC Server requires additional heap space when statement caching is active.

Typically, a simple SELECT (100 characters long) with two input variables and two output variables of type INTEGER uses about 350 bytes of the NonStop ODBC Server cache. In addition, the plan is stored in the SQL Executor's extended memory for later reference. Therefore, you should ensure there is enough primary (or secondary) disk space.

The NonStop ODBC Server's heap space can be configured for a server class by altering the HEAP-SIZE parameter. A NonStop ODBC Server needs about 400 kilobytes to run even without caching.

Statement caching is turned on in the user's profile. Therefore, a server class can switch between statement caching and no statement caching, depending on the identity of the user who logs on to the server. The cache is resized as different users with different caching profiles log on to the same server. If the statement cache is resized to be smaller, the least recently used statements are dropped out of the cache. If the cache is full and a new entry needs to be added, the least recently used cached statement is dropped.

The following process shows how caching is turned on for the user SQL_ODBC. In this scenario, this user's name has not been added to the system's user table (ZNSUS).

1. Add a profile entry that enables statement caching:

   ```
   NOSCOM> ADD PROFILE CACHE SQL_MAX_STATEMENT_CACHE 100
   ```

2. Add the user SQL_ODBC to the NonStop ODBC Server system tables with NOSUTIL. The following command, using the NonStop interface to NOSUTIL, NOSCOM (the same action can also be performed using the PC interface, the NonStop ODBC Server Configuration Manager), also associates the user with a profile named CACHE that enables statement caching:

   ```
   NOSCOM> ADD USER SQL_ODBC SQL.ODBC PROFILE CACHE
   ```

When SQL_ODBC logs on to the server, the CACHE profile is read. The NonStop ODBC Server is set up to cache up to 100 statements. If this limit is reached, the least recently used cached statements are dropped.

Suppose that SQL_ODBC issues five different SQL DML statements. The user disconnects and reconnects to the same NonStop ODBC Server and issues the same statements. The statements are executed immediately, and the time that would otherwise be spent compiling these statements is saved.

To turn off statement caching, set SQL_MAX_STATEMENT_CACHE to zero.

Caching Considerations

Consider the following items when configuring and using SQL statement caching:

- Statement caching requires additional extended memory. Therefore, there should be sufficient space on the swap volume of the NonStop ODBC Server processes to accommodate caching.

- If statement caching is being used and an INSERT statement has been parameterized, it is possible for a NonStop ODBC Server to return error 32214 the next time the INSERT statement is executed if the table referenced by the INSERT statement was dropped and re-created in an incompatible way. For example:

  ```
  create table t1 (c1 int)
  insert into t1 values (10)        is parameterized
  drop table t1
  create table t1 (c1 varchar(20))
  insert into t1 values ("abcdefgh")
  *** ODBC error 32214 reported
  ```

  Do not use statement caching if you are doing these types of DDL operations. If you encounter this problem, the easiest solution is to flush the cache (see the following subsection).

- When statement caching is enabled, the sort processes started when a SQL statement is first executed remain active on the system for the time the statement remains in the cache. For this reason, it is recommended that statement caching not be used where SQL statements can start many sort processes, unless you also use the NonStop SQL/MP DEFINE =_SQL_EXE_STOP_SORTPROGS to stop all sort processes started on behalf of a cursor whenever the cursor is closed. This DEFINE is described under Configuring SQL/MP on page 2-56.
- Statement caching is not used when the SQL compiler generates a query plan that would use parallel execution. The reason for this is that if a statement using parallel execution were cached, the ESP and sort processes used when the statement is executed would remain in the system until either the cache was cleared or the NonStop ODBC Server was stopped. This scenario could result in many ESP and sort processes remaining active on the system for extended periods, which would not be acceptable. In addition, the savings from caching compiled statements would normally be insignificant compared to the time required to execute a statement with parallel execution.
- Although a statement cache is preserved across session boundaries, each NonStop ODBC Server has its own statement cache, which means that on a subsequent connection a user might get a different NonStop ODBC Server with a different statement cache. This scenario can cause unexpected performance problems.

Flushing the Statement Cache

Flushing the statement cache means dropping all the cached statements from the cache. The cache can be flushed as a side effect by issuing a pass-through CONTROL statement to NonStop SQL/MP. For example:

```
SELECT "tdm: SQL CONTROL EXECUTOR PARALLEL EXECUTION OFF"
```

The CONTROL statement should be a non operative statement (in this case, parallel execution should already be off) so that you do not influence future SQL compilations. Any of the NonStop SQL/MP CONTROL statements have the same effect of flushing the cache.

Displaying Cache Statistics

The field SQL_MAX_STATEMENT_CACHE in the profile indicates how many statements should be cached per server (not per server class). Begin by configuring the cache size to be the maximum number of DML statements in your application. Then check cache statistics, as in the following procedure, and alter the cache size to obtain the minimum cache size with the most number of cache hits.

To obtain cache statistics, follow these steps:

1. Add a trace definition to record cache statistics:

   ```
   NOSCOM> ADD TRACE TRACE_CACHE
           TRA_LOG_TABLENAME SYSTEM_VOL_SUBVOL.SQL_ODBC.LOG1
           CACHE_STATISTICS Y
   ```

2. Change the user profile to reference the trace definition:

   ```
   NOSCOM> MODIFY PROFILE CACHE TRACE_NAME TRACE_CACHE
   ```

At the next polling cycle, the NOS server sets creates a trace record in the ZNSTRA system table. The server reads this record and logs cache statistics to the table specified after the LOG_TABLENAME option. It writes a record every time a statement comes in, listing whether it found the record and what the cost was. Statistics about the cache are written at the end of the session.

The following information is displayed in the log:

```
ID      DEBUG_FLAG  DEBUG_STRING
------  ----------  ----------------------------------------
96      LOGON       DEPT_ORDER
96      TIME        1995-01-12 16:25:46
96      CACHE       Number of Statements: 2
96      CACHE       Number of Lookups: 3
96      CACHE       Number of Hits: 1
96      CACHE       Number of Drops: 0
```

ID is the session's ID. A unique ID is generated for each NonStop ODBC Server session. The following information is shown for that ID.

Number of Statements	The total number of SQL statements in the cache at the end of the session
Lookups	The number of times the cache was referenced, once for each statement
Hits	The number of lookups that were successful; that is, the statement was found in the cache
Drops	The number of times the cache was full and a statement had to be dropped so a new statement could be cached

Performance Tuning

The issue of performance is concerned with metrics such as the response times seen by clients when they send requests to NonStop ODBC Servers. It also deals with resource usage on the NonStop system (CPU, memory, disk), on the LAN, and on the client PC or workstation. Performance tuning involves altering configurations in such a way as to increase or decrease the desired metric, such as response time, without letting other metrics, such as memory usage, exceed necessary limits.

Two of the major areas for performance tuning, using SCS server classes and statement caching, have been discussed previously in this section. This subsection provides additional suggestions on how to improve the performance of the system.

Configuring ODBC Datasources

By default, a NonStop ODBC datasource is set to Read-Write Standard, which allows the application to perform general write operations (INSERT, UPDATE, and DELETE) against the NonStop system, as well as UPDATE WHERE CURRENT OF operations on a cursor. A severe performance penalty can result from running in this mode, because the ODBC driver fetches only one row at a time from the server (each row is returned in a separate network packet). The NonStop ODBC driver provides other options that can be configured from the ODBC Administrator tool. These options are set from Data Access Option, and are described following:

Read-Only Standard

In this mode, the driver requests that the NonStop ODBC Server send as many rows as it can in a 4 KB block. The server actually sends the data back in 512-byte blocks, sending each block as it becomes full. Some performance measurements taken with this option have shown that performance can improve from a factor of 10 for very small rows (four bytes) to a factor of three for large rows (200 bytes). With this option, the application cannot perform any write operations. In this mode, data is returned in bursts, 4KB at a time. A delay occurs between 4KB blocks as the driver requests the server to send the next 4KB of data.

Read-Only Fast Transfer

In this mode, the driver requests that the server stream all rows back as soon as the data is ready. The server actually sends the data back in 512-byte blocks, sending each block as soon as it becomes full. Write operations are disallowed, and the application cannot have more than one statement handle active at a given time. If an attempt is made to open a second statement handle, the driver returns the error “Unable to support concurrent statement handle execution,” SQLSTATE=H4700. This transfer mode is the fastest, but might not be appropriate for a given application because of the restrictions it imposes.

Custom Mode

When configuring a datasource using the ODBC Administrator, this mode allows you to specify a more customized configuration. By choosing Custom, then clicking the Parameters button, you are given the options described next.

Access Mode

The parameters for this option are as follows:

SQL_MODE_READ_WRITE

The connection is configured to support full read/write access to the NonStop system.

SQL_MODE_READ_ONLY

The connection is configured for read-only access. Any attempt to perform a database update is rejected with an error.

SQL_MODE_NULL

The connection is configured to use the default for the system to which you are connecting. The default for NonStop is SQL_MODE_READ_WRITE.

Cursor Default Mode

The parameters for this option are as follows:

FOR_UPDATE

You can perform updates against a cursor.

READ_ONLY

No UPDATE WHERE CURRENT OF operations can be performed against the cursor; the driver uses the same blocking mechanism for getting rows from the NonStop ODBC Server as described for the “Read-Only Standard” data access option. This option is not related to the access mode described previously. In particular, this option still applies even if access mode is set to read/write.

READ_ONLY_STREAMING

No UPDATE WHERE CURRENT OF operations can be performed against the cursor; the driver uses the same streaming mechanism for getting rows from the NonStop ODBC Server as that for the “Read-Only Fast Transfer” data access option. This option is not related to the access mode described previously. In particular, this option still applies even if access mode is set to read/write.

It should be noted that there can be an advantage to using read/write mode when the amount of time it takes for SQL/MP to return a single row is on the order of hundreds of milliseconds or greater. In this case, individual rows are returned to the client faster in read-write mode because the ODBC driver requests only one row at a time; the client does not have to wait for the server to return multiple rows. Having individual rows returned faster can be better for certain types of applications.

The best way to determine the correct setting is to experiment with these options and select the one that gives the best overall performance.

This is not an issue for DBLIB clients because cursors are not supported in TRANSACT-SQL; all data is returned as though the “Read-Only Fast Transfer” option were being used.

Transaction Isolation

The parameters for this option are as follows:

SQL_TXN_READ_COMMITTED	The NonStop ODBC Server accesses SQL/MP by using STABLE access. This is the default.
SQL_TXN_READ_UNCOMMITTED	The NonStop ODBC Server uses BROWSE access.
SQL_TXN_REPEATABLE_READ	The NonStop ODBC Server uses REPEATABLE access.
SQL_TXN_SERIALIZABLE	The NonStop ODBC Server uses REPEATABLE (serializable) access.

See the *NonStop SQL/MP Reference Manual* for detailed information on these settings.

Auto Commit

The parameters for this option are as follows:

0 Disables auto commit; the application must commit a transaction.

1 Enables auto commit; the NonStop ODBC Server commits the transaction after each SQL statement.

Login Time-Out

The parameter for this option is as follows:

n The number of seconds that the NonStop ODBC driver waits for a connection to be completed. If *n* is zero, the driver waits indefinitely.

Note. Settings on the PC override the profile settings on the NonStop system set up for that user. Configure the PC settings for ODBC datasources carefully to ensure the best performance.

Using Batches for DBLIB Clients

For the best performance, you often want to minimize the number of messages between the client and the server. One good way of achieving this with DBLIB clients is by the use of batches, where a batch is a set of SQL statements that are all sent in one request to the server.

One caution is that each statement in a batch uses heap space in the NonStop ODBC Server, so it is recommended that a single batch not exceed more than a few hundred

SQL statements. If a batch is to include several hundred (> 300) SQL statements, the heap size for the NonStop ODBC Server process needs to be set higher than the recommended default of 750 KB. As a rule, allocate 2.5 KB per SQL statement when calculating the heap size.

Batches are not supported in ODBC.

Using Stored Procedures

Using a stored procedure, a client can submit a single execution request and have the server execute a complex logic sequence that can include as many SQL statements as needed. See Section 5, “Stored Procedures,” in the *NonStop ODBC Server Reference Manual* for further details on how to use stored procedures with the NonStop ODBC Server.

Managing Connect/Disconnect Processing

Even though the use of prelaunched servers provides good response time for connection requests, it is still true that, in general, applications should try to minimize the number of connect and disconnect requests that are used. It should be noted, however, that each connection request results in a NonStop ODBC Server process and a SQLCOMP process being assigned, so it is also true that you should not keep connections open for extended periods.

If you experience long connect times, you should use SCF to check the number of servers available for each of the server classes and adjust the numbers as necessary. See Configuring and Managing SCS on page 4-1, and Using the Subsystem Control Facility on page 4-29, for further information.

Configuring Communication Stacks

When configuring the communication stacks on the client, you must take care to ensure that the various configuration parameters have been set to optimize performance. For example, for some stacks there is a configuration parameter that specifies how long to wait on disconnection requests before returning to the client. If this parameter is set to a high value, it will affect the time taken for a client to disconnect from the server. This particular problem can be easily checked by seeing how long a disconnection takes by using either the NonStop ODBC Server Connectivity Tool under Windows or ISQL in a different environment.

Complete information related to installation of the communication stacks can be obtained from the vendor of the communication stack being used.

Setting Up SCS

For the best performance, it is recommended that the SCS process run in a different CPU from the NonStop ODBC Server processes and the TCP/IP process. If it is running in a separate CPU, the priority of the SCS process should be as high as possible. If SCS is running in the same CPU as NonStop ODBC Server processes, the priority of SCS should be lower than that of the NonStop ODBC Server processes.

Configuring SQL/MP

Set up the NonStop SQL/MP system to provide optimum performance. Ensure that the DP2 cache is configured correctly for the volumes where the NonStop ODBC Server catalogs reside, for example, and provide a general tuning of the database for optimal query performance.

To ensure the best performance from the NonStop ODBC Server, keep the SQL statistics current on the NonStop ODBC Server catalog tables COLUMNS, INDEXES, TABLES, ZNUIX, and ZNUOBJ. This strategy is particularly necessary for any NonStop ODBC Server databases in which there has been a lot of activity involving adding and dropping tables. A good indication of a problem in this area is if a NOSUTIL REFRESH operation takes a long time to finish.

A sample SQLCI command to update the statistics follows:

```
UPDATE ALL STATISTICS FOR TABLE ZNUOBJ NORECOMPILE;
```

Using the NonStop ODBC Server Connectivity Tool

In the Windows environment, you should use the Connectivity Tool for measuring performance and locating performance problems. Options provided by the Connectivity Tool follow:

CONNECT Use this option to determine the amount of time it takes to connect to a NonStop ODBC Server for a given user ID and in conjunction with the SCF STATUS commands to check that the correct server class is being used for a given user.

ECHO Use this option to send a pass-through statement to a NonStop ODBC Server; the only action the Server takes is to reply immediately. This option provides a measurement for the minimum amount of time it takes for an NonStop ODBC Server to process a request: the request latency time. In general, a long response time indicates either a LAN problem or a problem with the CPU in which the NonStop ODBC Server, or SCS, or network IOPs are running.

QUERY Use this option to determine how long it takes to execute a SQL query. For a SELECT statement, the time is the amount of time for the first result packet to reach the Connectivity Tool.

For the preceding options, you should generally request at least five samples, then display the resulting graph to look for unexpected variances. For example, the graph is very useful for showing the effects of statement caching, where the first statement execution time should be much longer than subsequent executions.

This tool is very useful for checking on the results of some of the tuning recommendations made earlier. You are encouraged to experiment with different options and to use the tool to evaluate the performance changes.

Authorization and Security

This subsection discusses the authorization needed on the NonStop system to perform actions within the NonStop ODBC Server environment. Security issues related to the PC, workstation, or LAN are not covered here.

Authorization is based on Guardian user names. The NonStop ODBC Server programs, NOS and NOSUTIL, that manage SQL/MP objects are regular SQL/MP programs, and as such they are subject to the normal SQL/MP rules that govern the access to SQL/MP objects. For example, to be able to create a table, the NOS process executing the request must have read/write access to the associated SQL/MP catalog tables, as well as read/write access to the NonStop ODBC Server catalog tables.

The actions you can do related to the NonStop ODBC Server are as follows:

Action	**Description**
NOSINST	Install NOS, NOSUTIL object files, build NonStop ODBC Server SYSCAT
NOSUTIL START	Start SCS and NonStop ODBC Servers
NOSUTIL ADD/DEL	Add/Delete user, alias, profile, trace
NOSUTIL SYSCAT	Install, deinstall, validate, refresh, cleanup NonStop ODBC Server SYSCAT
NOSUTIL USERCAT	Install, deinstall, validate, refresh, cleanup NonStop ODBC Server USERCAT
CLIENT	Connect and use NonStop ODBC Server
ADMIN	Perform NonStop ODBC Server system administration

The objects that the preceding actions involve are as follows:

Object	**Description**
SQL/MP SYSCAT	SQL/MP system catalog
SQL/MP USERCAT	SQL/MP user catalog
NonStop ODBC Server SYSCAT	NonStop ODBC Server system catalog
NonStop ODBC Server USERCAT	NonStop ODBC Server user catalog
SCSOBJ	SCS object file
NonStop ODBC Server	NonStop ODBC Server object file (NOS)
NonStop ODBC Server Utility	Utility program object file (NOSUTIL) and its interfaces, NOSCOM (NonStop product) and the NonStop ODBC Server Configuration Manager (PC-based)
SQLCOMP	SQL/MP compiler

Each process involved with NonStop ODBC Server runs under a Guardian user name, and that user name is used to authorize access to objects. Each user has a default security vector, and when a process creates an object for that user, that object is secured with the default security vector, unless options in the create process override. After being created, the security vector for an object can usually be changed.

Table 4-1 shows the authorizations needed to perform each NonStop ODBC Server action.

The following mnemonics are used in Table 4-1 to describe the authorization requirements:

R	Read the object
W	Write the object
E	Execute the object
O	Owner of the process
CR	Create the object (and SQL compile it, if applicable); creator owns the object.
CRO	Create SQL/MP objects (in the catalog)
LI	License the object

Entries shown in parentheses indicate that the authority is needed only under limited conditions. For USERCAT, for example, write authority to the NonStop SQL/MP system catalog is required only when executing the INSTALL function.

Table 4-1. NonStop ODBC Server Security Requirements (page 1 of 2)

Accessor	NSSQL/MP SYSTEM CATALOG	NSODBC SYSCAT	NSODBC USERCAT	SCSOBJ	NSODBC Server	NOSUTIL	SQLCOMP	Notes
NOSINST	R, W, CRO	CR			CR	CR	E	Must be privileged user (1)
NOSUTIL START	R	R		E	E	E	E	Must be privileged user
NOSUTIL ADD/ REMOVE	R	R, W				E	E	(2)

Table 4-1. NonStop ODBC Server Security Requirements (page 2 of 2)

Accessor	NSSQL/MP SYSTEM CATALOG	NSODBC SYSCAT	NSODBC USERCAT	SCSOBJ	NSODBC Server	NOSUTIL	SQLCOMP	Notes
NOSUTIL SYSCAT	R, W	R, W, (CR)				E	E	Must be privileged user
NOSUTIL USERCAT	R, (W)	R, (W)	CR			E	E	(3)
CLIENT	R, (W)	R, (W)	R, (W)		E	E	E	(4)
ADMIN	R, W	R, W	R	E, O	E	E	E	(5)

(1) In addition, to run NOSINST, you must have execute access permission on the SQL/MP process SQLCAT.

(2) To add a logical user name, you must either be logged on under the Guardian user name to be associated with the logical user name or be logged on as a super-group user. To delete a logical user name, you must be logged on as a super-group user.

(3) If a SQL/MP catalog must be created or dropped, the SQL/MP system catalog must also be written to.

(4) The authorizations shown are minimal. If the client creates objects, SQL/MP and NonStop ODBC Server catalogs are also written to (the client needs the same authorizations as if using SQL/MP directly). If a pass-through NOSUTIL statement is used, add the authorizations needed for that statement.

(5) In general, the NonStop ODBC Server administrator needs full read/write access to the NonStop ODBC Server system catalog in order to manage the system (for example, to add profile and trace records). It is likely that ADMIN will be a super-group user to run the SYSCAT VALIDATE and REFRESH functions. ADMIN should also be the person who configures, starts, and stops SCS. ADMIN also needs to be able to execute SCF and read the EMS log.

Controlling Access to a NonStop ODBC Server System

This subsection discusses how you can secure your NonStop ODBC Server system to provide certain access privileges. To perform any operation within a NonStop ODBC Server system, you must have the correct access privileges to the SQL/MP catalogs, the NonStop ODBC Server catalogs, and the individual SQL/MP objects. By applying security at these different levels, it is possible to set up many different security schemes, as discussed following.

When securing the NonStop ODBC Server user catalog tables (ZNUOBJ and ZNUIX), you should secure them with the same security vector. The easiest way to secure the tables is by using the SQLCI SECURE command; for example:

```
SECURE ZNU* "CCCO";
```

This string only works when there are no user tables beginning with “ZNU.” You do not need to worry about the security of the NonStop ODBC Server catalog views (ZVU*), because they are all shorthand views.

Security Considerations for Installation

Installation of the NonStop ODBC Server is performed by using the TACL script NOSINST supplied with the product. To run NOSINST, you must be logged on as the super ID. After running NOSINST successfully, the program object files, NOS and NOSUTIL, and the NonStop ODBC Server system catalog have the super ID as owner.

NOS and NOSUTIL have a security vector of “NONO” to allow anyone to execute them. All of the NonStop ODBC Server system catalog tables are secured “NNNO,” with the exception of ZNUDT, ZNSDUMMY, ZNSPROT, and ZNSVALUE, which are secured “NONO.” The security vector “NNNO” provides open access to the system catalog table, allowing any user to create a database or use the utility functions such as the ADD USER function. The security of these programs and tables can be changed after installation. The other tables are not secured for general write access because they are never changed after the installation.

Client Access

The first level of security is provided through the logon name and password required before a client can connect to NonStop ODBC Server. Access can be further restricted by not configuring a default server class in the SCS configuration file. If no default class is configured, SCS accepts only connection requests where the logon username maps to one of the configured server classes.

Creating a Database

The easiest way to control who can create a new database, either by issuing a CREATE DATABASE statement from the client, or by using the USERCAT INSTALL statement, is to restrict write access to the NonStop ODBC Server system catalog table ZNSDB. For example, by securing the table as “NCOO,” only super-group users can write to the table and thus create a database. The same effect could be achieved by limiting access to the NonStop SQL/MP CATALOGS table, because this limits the creation of the NonStop SQL/MP catalog created as part of a NonStop ODBC Server database.

Default Database Security

When a database is first created, either by the client by sending a CREATE DATABASE request or by using the USERCAT INSTALL statement, the security of the database is set to the default Guardian security associated with the user creating the database.

Restricting Database Access

The easiest way to limit access to a database is by securing the NonStop ODBC Server catalog tables for the database appropriately. For example, if the NonStop ODBC Server catalog tables are secured “CCCO,” only members of the owner’s user group can access the database. This type of security prevents unauthorized users from seeing which tables exist in the database. Restricting access to individual SQL/MP objects within the database should be done by securing the individual SQL/MP tables and views.

Database Access Controls

NonStop ODBC Server uses information from the SQLConnect command to complete Guardian user authentication. NonStop ODBC Server aliases allow a user to have more than one named user context utilizing the connection parameters of username and datasource. Each user context defines the default catalog, transaction locking, cursor characteristics, resource policies, and so on. NonStop ODBC Server aliases and Safeguard aliases can be used together to achieve multiple contexts, individual user authentication/audit, and a combination of the features provided by each.

Where SQL/MP uses a physical system configuration, NonStop ODBC Server uses a logical system configuration. NSODBC_SYSTEM_CATALOG points to an SQL system catalog or user catalog, and indicates a collection of SQL tables used by NonStop ODBC Server to describe a system configuration or multiple configurations. A NonStop ODBC Server configuration describes a collection of ODBC databases, their mapping to SQL/MP catalogs, their logical usernames, and their context specification (PROFILE).

As an optional feature NonStop ODBC Server can limit which SQL tables are visible, limit user access (READONLY or READWRITE), and limit which users can create objects by applying one or more optional settings to ODBC databases and SQL catalogs.

Limiting Access to One or More Catalogs

The following example shows how to limit user access to one or more catalogs:

1. Choose a subvolume for the ODBC Server System Catalog, and customize it using the following statement.

   ```
   ADD DEFINE = _NSODBC_SYSTEM_CATALOG,CLASS CATALOG, SUBVOL
   $data1.odbcsys
   ```

2. Install the NonStop ODBC Server. For details on how to install the NonStop ODBC Server, see Installation on page 1-1.

3. Add each user who is authorized to have access to this particular configuration, using the following statement:

   ```
   NOSCOM> ADD USER LOGICAL_NAME GUARDIAN_NAME
   ```

4. For each SQL/MP catalog the user is authorized to access, use the following statement:

```
NOSUTIL> USERCAT INSTALL $data1.cat1
```

Users do not have access to tables in catalogs that are not explicitly installed into this ODBC system configuration. Passthrough commands allows users to issue any SQL/MP statement directly. However, because each user opens a file as a Guardian-authenticated user, standard SQL/MP access controls govern access to the catalog.

Limiting the Table Names Listed for a Data Source

For security purposes, or to limit the use of memory resources, you can use the NonStop ODBC Server shadow catalogs and protection views to restrict the list of table names returned to an application.

- **Creating a Shadow Catalog For Specific Tables**

  A shadow catalog is a NonStop ODBC server customized catalog, which contains logical references to one or more SQL/MP tables that are registered in other SQL/MP catalogs.

  To create a shadow catalog, perform the following steps.

  1. Create a shadow catalog in a separate subvolume using the following statement:

     ```
     NOSCOM> USERCAT INSTALL \node.$data1.privcat
     ```

  2. Logically map each table for which access is to be granted in the shadow catalog.

     ```
     NOSCOM> ADD TABLE $data2.payroll.y01q01 AS
     node_data1_privcat.dbo.y01q01
     ```

  Only table objects that have been added are now accessible from an application that is accessing objects in the database or catalog *node_data1_privcat*.

- **Using a Protection View to Obtain Column Data**

A protection view allows you to restrict the amount of table data returned, but provides associated column data. If you need to restrict the amounts of data returned, but you need to see the column data, you must add a protection view.

To obtain column data for a limited set of tables, perform the following steps:

1. Add an SQL/MP protection view to the catalog *$data1.privcat*

   ```
   SQLCI> CREATE VIEW $data1.privcat.y01q01 AS SELECT * FROM
   $data2.payroll.y01q01 FOR PROTECTION
   ```

2. Logically map each table in the shadow catalog *node_data2_payroll*

   ```
   NOSCOM> ADD VIEW $data1.privcat.y01q01 AS
   node_data1_privcat.dbo.y01q01
   ```

This method uses more storage than simply creating a shadow catalog, but supports metadata on the tables, which some applications require.

- **Specifying Published Columns in a Protection View**

You can further restrict access to specific columns by specifying particular columns to be published in the protection view, as follows.

1. Create a shadow catalog in a separate subvolume by using the following statement:

```
NOSCOM> USERCAT INSTALL $data1.privcat
```

2. Add a protection view to the shadow catalog with a column list

```
SQLCI> CREATE VIEW $data1.privcat.y01q01 AS SELECT c1, c2 ...
FROM $data2.payroll.y01q01 FOR PROTECTION
```

The created view can be updated if all the columns in the base table, not included in the column list, are defined in SQL/MP with a default value. NonStop ODBC Server will list only those columns named in the CREATE VIEW statement.

Restricting Access to Read-Only

To restrict a client to read-only sessions, the recommended method is to secure the relevant SQL/MP objects accordingly. Setting the ACCESS_MODE option in the user's profile (in ZNSPROF) to read-only is a less secure choice because the client can reset the ACCESS_MODE option by using the pass-through SET ACCESS_MODE statement to change the session behavior.

Restricting the Creation of Objects in a Database

The simplest way to restrict the creation of SQL objects in a database is to secure the NonStop ODBC Server catalog tables for the appropriate write access. For, by securing the NonStop ODBC Server catalog tables as "COCO," only the owner can write to them, which means that only the owner can create objects in the database. In addition, the SQL/MP catalog tables associated with that database also must be secured to provide read/write access to the same group of users.

Security for Stored Procedures

Security for a stored procedure is controlled by the security of the NonStop ODBC Server catalog tables in which the stored procedure is registered, the same as any other SQL object registered in the NonStop ODBC Server catalog tables. In addition, there is the security of the Pathway server where the stored procedure has been set up. It is the system administrator's responsibility to set up the security of the Pathway server to provide the correct level of user access.

Security for Utility Functions

Many of the utility functions, such as SYSCAT VALIDATE, REFRESH, CLEANUP, ADD USER, and REMOVE USER, have their own set of authorization checks built in. For example, SYSCAT REFRESH can only be performed by a super-group user. The best way to impose additional security for the other statements is by restricting write access to the NonStop ODBC Server system catalog tables ZNSALT, ZNSPROF, ZNSTRA, and ZNSUS. For example, if the security of these tables is set to “NCNO,” only super-group users can write to the tables, so only super-group users would be able to add aliases and set up profile or trace records.

Note. Do not use a SQLCI command of the form SECURE ZNS*... to set the security of the preceding tables, because there are other NonStop ODBC Server system catalog tables whose security should not be changed. Most notably, write access to ZNSDB is needed by other users for them to be able to create and drop databases.

PROGID Security

PROGID is a Guardian file security parameter that allows you to designate a single user ID for execution of a program regardless of the user ID of the calling process. PROGID support satisfies a customer’s need to allow a set of users to be able to run NOSCOM and perform administrative functions of adding and deleting users without requiring the users to be first authenticated as super-group users or the super ID.

When PROGID is set for NOSUTIL, access should be limited either by setting the security string to allow use by only a well-defined group of intended users or by using the Safeguard product to explicitly define an allowed list of users. Additionally, this NOSUTIL program should be separated from the NonStop ODBC Server system and not be the executable when started by SCSOBJ or the NOS program.

Setting the PROGID Attribute

The syntax for setting the PROGID attribute for NOSUTIL is as follows:

```
FUP SECURE NOSUTIL , ["] security-string ["] , PROGID
```

The *security-string* value should be set according to the installation’s security requirements for limiting access; the use of the Safeguard product is recommended for a NOSUTIL program with PROGID security.

When NOSUTIL is run, the PROGID attribute sets the process accessor ID to the program file owner’s ID, instead of the ID of the actual accessor.

To use the FUP SECURE command, you must be the file owner, the owner’s group manager, or a super-group user.

PROGID Behavior

PROGID security affects the behavior of all NOSUTIL statements. Additionally, PROGID affects the behavior of the SQL/MP utility statements that NOSUTIL uses to create SQL objects.

The following rules apply to PROGID security:

- If the PROGID attribute is set, the creation of a NonStop SQL catalog causes each of the catalog tables (TABLES, COLUMNS, and so on) to have the user ID of the PROGID-secured program—instead of the user ID of the invoking user or user program—and a file security of “NNNO,” regardless of the default file security of the invoking user or the PROGID user ID.
- If the PROGID attribute is set, access to objects is determined as if the user were authenticated as the PROGID user ID.
- If the PROGID user ID is set to a super-group user ID, the management functions that NOSUTIL by default restricts to only super-group users can be executed by any user with the right to execute this PROGID-secured NOSUTIL without having to log on as a super-group user.

These rules can result in perhaps unintended but reasonable and predictable consequences. The following example demonstrates the result of setting the PROGID attribute on the NOSUTIL process.

The users involved with this example are as follows:

User Name	User ID	Default File Security
SUPER.NSODBC	255,127	“NUNU”
SQL.USER1	175,51	“NNNO”

The PROGID attribute has been set for NOSUTIL; the pertinent characteristics of the NOSUTIL program file are as follows:

```
              Code  ...     Owner   RWEP    ...
NOSUTIL       100P  ...   255,127  "NUNU"   ...
```

When SQL.USER1 logs on successfully, the current session environment is established as follows.

```
Current volume:  $disk1.db1
Saved volume:    $disk1.db1
Userid: 175,51  Username: sql.user1  Security: "NNNO"
```

The system catalog is installed in $SYSTEM.SQL. A SQL user catalog has already been created in $DISK1.DBA. No SQL files reside in $DISK1.DBB.

Based on the preceding assumptions, user SQL.USER1 can perform any of the NOSUTIL statements shown in Table 4-2 with all of the privileges of user SUPER.NSODBC (user ID 255,127), the owner of the NOSUTIL program file.

Table 4-2. Effect of PROGID Security on NOSUTIL Operations (page 1 of 3)

Command	Conditions Met	Results
System Catalog Commands		
SYSCAT INSTALL	User 255,127 can create tables, indexes, and views in the SQL system catalog.	The NonStop ODBC mapping tables are created with owner 255,127 and file security “NNNO.”
SYSCAT DEINSTALL	User 255,127 can drop tables, indexes, and views composing NonStop ODBC mapping tables.	The NonStop ODBC mapping tables are dropped.
SYSCAT REFRESH	Same as SYSCAT INSTALL; also, user 255,127 can READ/WRITE against the NonStop ODBC mapping tables.	Same as SYSCAT INSTALL for any object re-created (a missing mapping table, for example).
SYSCAT UPGRADE	Same as SYSCAT REFRESH.	Same as for REFRESH.
SYSCAT VALIDATE	Similar to SYSCAT REFRESH that requires only READ access.	None—no updates occur.
User Catalog Commands for Existing SQL Catalogs		
USERCAT INSTALL $disk1.dba	User 255,127 can create views against local SQL tables in $disk1.dba and against ZNSDT, USAGES, ZNSUS in the SQL system catalog.	The NonStop ODBC mapping tables are created with owner 255,127 and file security “NNNO.”
USERCAT DEINSTALL disk1.dba	User 255,127 can drop tables, indexes, and views composing NonStop ODBC mapping tables.	The NonStop ODBC mapping tables are dropped.
USERCAT REFRESH $disk1.dba	Same as USERCAT INSTALL; also, user 255,127 can READ/WRITE against the NonStop ODBC mapping tables.	Same as USERCAT INSTALL for any object re-created.
USERCAT UPGRADE $disk1.dba	Same as USERCAT REFRESH.	Same as USERCAT REFRESH.
USERCAT VALIDATE $disk1.dba	Similar to USERCAT REFRESH that requires only READ access.	None—no updates occur.

Table 4-2. Effect of PROGID Security on NOSUTIL Operations (page 2 of 3)

Command	Conditions Met	Results
User Catalog Commands for Existing SQL Catalogs		
CLEANUP	For a user catalog, same as USERCAT REFRESH; for the system catalog, same as SYSCAT REFRESH.	All temporary tables are removed.
User Catalog Commands for Empty Subvolumes		
USERCAT INSTALL $disk1.dbb	User 255,127 can create views against local SQL tables in $disk1.dbb and against ZNSDT, USAGES, ZNSUS in the SQL system catalog.	The NonStop ODBC mapping tables and the SQL catalog are created with owner 255,127 and file security “NNNO.”
USERCAT DEINSTALL disk1.dbb CASCADE NSSQL	User 255,127 can drop tables, indexes, and views composing NonStop ODBC mapping tables and the SQL catalog tables.	The NonStop ODBC mapping tables and the SQL catalog are dropped.
USERCAT REFRESH $disk1.dbb	Same as USERCAT INSTALL; also, user 255,127 can READ/WRITE against the NonStop ODBC mapping tables.	Same as USERCAT INSTALL for any object re-created.
USERCAT UPGRADE $disk1.dbb	Same as USERCAT REFRESH.	Same as USERCAT REFRESH.
USERCAT VALIDATE $disk1.dbb	Similar to USERCAT REFRESH that requires only READ access.	None—no updates occur.
CLEANUP	For a user catalog, same as USERCAT REFRESH; for the system catalog, same as SYSCAT REFRESH.	All temporary tables are removed.
Other Management Commands		
ADD USER	User 255,127 can enter data into the table ZNSUS in the SQL system catalog.	ZNSUS is changed.
ADD ALIAS	User 255,127 can insert into the table ZNSALT in the SQL system catalog.	ZNSALT is changed.
ADD PROFILE	User 255,127 can insert into ZNSPROF.	ZNSPROF is changed.
ADD TRACE	User 255,127 can insert into ZNSTRA.	ZNSTRA is changed.

Table 4-2. Effect of PROGID Security on NOSUTIL Operations (page 3 of 3)

Command	Conditions Met	Results
ADD PROCEDURE	User 255,127 can insert into tables ZNSPROC and ZNSPCOL in the current catalog.	The procedure entry is added. If the procedure name did not indicate owner qualification, the invoking logical username is used. For example, `ADD PROCEDURE p1 $m s1` would add NODE_DISK1_DB1.SQL_USER1.P1.
REMOVE USER	User 255,127 can delete rows from the table ZNSUS in the SQL system catalog.	ZNSUS is changed.
REMOVE ALIAS	User 255,127 can delete rows from the table ZNSALT in the SQL system catalog.	ZNSALT is changed.
REMOVE PROFILE	User 255,127 can delete rows from ZNSPROF.	ZNSPROF is changed.
REMOVE TRACE	User 255,127 can delete rows from ZNSTRA.	ZNSTRA is changed.
REMOVE PROCEDURE	User 255,127 can delete rows from tables ZNSPROC, ZNSPCOL in the current catalog.	The procedure entry is deleted. If the procedure name did not indicate owner qualification, the invoking logical username is used. For example, `REMOVE PROCEDURE p1` would delete NODE_DISK1_DB1.SQL_USER1.P1.

Troubleshooting Security Problems

If you are having trouble installing or creating objects, check for the necessary security. Client applications do not always relay detailed error information (instead, they report something like “server failure”). See Tracing on page 4-34 to learn how to use tracing to display error information.

A quick check for clients that have problems is to run again as the super ID. If this strategy works, look for a security problem.

If you have problems bringing the NonStop ODBC Server system up, check the security of the NonStop ODBC Server program files and the security of the NonStop ODBC Server system catalog tables and the file $SYSTEM.SYSTEM.ODBCMSG. In the event of security errors in these areas, you should receive errors reported to the EMS log.

Using Error and Event Messages for System Diagnosis

This subsection summarizes error and event messages emitted by the SCS process and by NonStop ODBC Servers. You can use these messages to diagnose configuration errors, internal errors, system errors, and communication errors.

EMS Event Messages From the SCS Process

The following is a list of possible Event Management Service messages from the SCS process. Detailed descriptions of EMS event messages that can be generated by SCS, complete with exact display format and all tokens involved, are given in the *NonStop ODBC Server Messages Manual*.

The SMB minor error describes the specific error condition. The common minor error codes that might appear in an SMB-ERROR event follow:

10 ERROR_BROKEN_PIPE
9 The SCS process experienced an I/O error communicating with the gateway process.

23 ERROR_PIPE_BUSY
1 The SCS process was unable to service a Named Pipes connection request from a client program. This error occurs when the connection attempts exceed the number of gateway processes configured for the SCS process. You have under-configured your SCS process.

23 ERROR_PIPE_NOT_CONNECTED
3 This error indicates that a gateway process, previously connected to a client, has closed the connection unilaterally. In this case, the gateway formally disconnected from the SCS process. This error signals the client that the gateway process is no longer available. Check the EMS log to determine why the NonStop ODBC Server program terminated.

The following SMB minor errors indicate serious problems.

6 ERROR_INVALID_HANDLE
Client driver provided a bad pipe handle number.

11 ERROR_BAD_FORMAT
A PC client incorrectly issued a DosCallNmPipe request.

87 ERROR_INVALID_PARAMETER
This error indicates a problem with the PC driver. If emitted as a result of a transact operation, SMB command 21, the PC is committing a protocol use violation. For any other command, this error indicates that an incorrect parameter was submitted at the client API level and was not correctly screened by the client driver.

23
0 ERROR_BAD_HANDLE
The client software is attempting to communicate with a bad handle number. Probably a client driver problem.

-1 ERROR_NOT_SUPPORT
The SMB sent in by the client program is not supported by the SCS process. This error should be reported to your HP representative, although this error does not always indicate a HP problem. SCS was designed to conform to the LAN Manager/X, version 1.1, test suites. The receipt of unsupported SMBs might indicate that the Microsoft ODBC product has extended the protocol in some later version or that a third-party implementation of the protocol is not entirely correct. In the former case, HP might consider upgrading the SCS product to support the extended protocol. In the latter case, it might be the responsibility of the third party to resolve the situation.

Multilan Errors

This subsection describes the possible entries relating to NETBIOS you might see in an MLAM-IO-ERROR event and how to interpret them.

The following is a list of NETBIOS command codes:

NETBIOS Command	**Why It Might Appear in an MLAM-IO-ERROR Event**
11 (LISTEN)	SCS encountered an error while listening for a new client connection request. This code indicates a possible LAN problem or abnormal client termination.
20 (SEND)	SCS encountered an error trying to send data to the client. If the error code is 9, there is insufficient MLAD memory space to do the send. SCS retries an error 9 up to three times. Other errors cause the session to be terminated.
21 (RECEIVE)	SCS encountered an error trying to receive data from a client. If the error code is 9, there is insufficient MLAD memory space to do the send. SCS retries an error 9 up to three times. Other errors cause the session to be terminated.
48 (ADD NAME)	SCS was unable to add the NETWORK-NAME to the LAN. Another process is probably using the same name.

The following is a list of common NETBIOS return codes that might be displayed by the SCS process in an MLAM-IO-ERROR event. For a complete list of return codes, see the *Multilan/TLAM Programming Manual*.

Most other NETBIOS return codes cause an abnormal termination of the session that encountered the error.

TCP/IP Errors

This subsection describes the TCP/IP errors you might see in a TCPIP-IO-ERROR event or an NW-IO-ERROR event.

The following is a list of common TCP/IP errors and meanings.

TCP/IP Error	**Meaning**
4001	Not owner, permission denied
4002	No such file or directory
4003	No such process or table entry
4004	Interrupted system call
4005	I/O error
4006	No such device or address
4007	Argument list too long
4008	Exec format error
4009	Bad file descriptor
4010	No children
4011	No more processes
4012	Insufficient user memory
4013	Permission denied
4014	Bad address
4016	Mount device busy
4017	File already exists
4018	Cross-device link
4019	No such devices
4020	Not a directory
4021	Is a directory
4022	Invalid function argument
4023	File table overflow
4024	Maximum number of files already open
4025	Not a typewriter
4027	File too large
4028	No space left on device

4029	Illegal seek
4030	Read only file system
4031	Too many links
4032	Broken pipe or no reader on socket
4033	Argument out of range
4034	Value out of range
4045	Deadlock condition
4046	No record locks available
4099	Function not implemented
4101	Operation would block
4102	Operation now in progress
4103	Operation already in progress
4104	Socket operation on non-socket
4105	Destination address required
4106	Message too long
4107	Protocol wrong type for socket
4108	Protocol not available
4109	Protocol not supported
4110	Socket type not supported
4111	Operation not supported on socket
4112	Protocol family not supported
4113	Address family not supported
4114	Address already in use
4115	Can't assign requested address
4116	Network is down
4117	Network is unreachable
4118	Network dropped connection on reset
4119	Software caused connection abort
4120	Connection reset by remote host
4121	No buffer space available
4122	Socket is already connected
4123	Socket is not connected
4124	Can't send after socket shutdown

4126	Connection timed out
4127	Connection refused
4128	Host is down
4129	No route to host
4131	File name too long
4132	Directory not empty
4195	Out-of-band data available
4196	Invalid socket call
4197	File type not supported
4198	C file (code 180) not odd-unstructured
4199	Insufficient internal memory

NonStop ODBC Server Shutdown Messages

This subsection describes messages passed by the NonStop ODBC Server when you stop SCS. At that time, the following messages appear in the EMS log for each running NonStop ODBC Server process:

```
95-01-21 13:54:27 \TESS.1,70       TANDEM.EMS.Dxx       000512 T8666Dxx NOS

95-01-21 13:54:27 \TESS.1,70       TANDEM.EMS.Dxx       000512 Time :
                                   1995-1-21 13:54:27

95-01-21 13:54:27 \TESS.1,70       TANDEM.EMS.Dxx       000512 ODBC Server
                                   message (32312 : Error 233 connecting to
                                   SCS.
                                         ) :

95-01-21 13:54:27 \TESS.1,70       TANDEM.EMS.Dxx       000512 T8666Dxx NOS

95-01-21 13:54:27 \TESS.1,70       TANDEM.EMS.Dxx       000512 Server Name :
                                   $X285

284
                                       1995-1-21 13:54:27

95-01-21 13:54:27 \TESS.1,70       TANDEM.EMS.Dxx       000512 ODBC Server
                                   message (32310 : Error 3 opening SCS.
                                   Server terminating.
                                                      ) :
```

The NonStop ODBC Server generates these messages because it has detected that the SCS process has terminated and the server attempts to reconnect. No errors have occurred, but the NonStop ODBC Server reports errors because, at present, it cannot distinguish between a normal shutdown and an SCS process terminating abnormally. You can ignore these messages.

Using the Subsystem Control Facility

The NonStop ODBC Server includes a limited SCF capability to query object information and status for the SCS process, server class, and server process objects and to generate a trace. The SCS process (SCSOBJ) is the management focal point for the NonStop ODBC Server product. This new SCF capability allows you to query SCS and to determine the effects of various configuration changes.

Following successful completion of the INSTALL utility, two new files, ZSCSSF and ZSCSTEXT, appear in $SYSTEM.SYSTEM. The SCF capability cannot function if these files are not present on $SYSTEM.SYSTEM.

The following three SCF commands are supported for SCS:

INFO PROCESS
STATUS PROCESS
TRACE PROCESS

To use the SCF interface, enter the following command at the TACL prompt:

```
SCF
```

You will then see the following display:

```
SCF - T9082D30 - (30NOV95) (06NOV95) - 05/28/96 15:48:47 System \TESS
Copyright Tandem Computers Incorporated 1986 - 1996
(Invoking \TESS.$SPOOL.DK.SCFCSTM)
1->
```

INFO PROCESS Command

The INFO PROCESS command displays information about the configuration of the running SCS process, including the number of PCs that can connect to the SCS process and the number of connections that are allowed.

The syntax for this command is as follows:

```
INFO PROCESS $SCSProcessName [ , DETAIL ]
```

The following is a sample report generated by INFO PROCESS using the DETAIL option:

```
SCS Detailed Info PROCESS \TESS.$XYZ

Network Name ..............  GQL314TM
Network Qualifier .........  \PIPE\SQL\QUERY

Maximum Sessions ..........  40
Server Classes ............  3
Server FileName ...........  \TESS.$SQLCON.ZNOS.NOS
Input Buffer ..............  2448
Output Buffer .............  2448
Transport Protocol ........  NETBIOS
IOP Device Name ...........  \TESS.$LAN1.#WSPTP
MLAD Adapter ..............  0
MLAD Resource .............  #GWT1
MLAD Resource Type ........  1

ServerClass                  MaxServers   MinServers   Priority
ZZTDMMAPPER                           1            1        144
```

```
                           CPUS (0,1,2,3,4,5,6,7)

ServerClass                  MaxServers   MinServers   Priority
CLASS1                               20            1        170
                           CPUS (1,2,3)

ServerClass                  MaxServers   MinServers   Priority
CLASS2                               20            1        170
                           CPUS (1,2,3)
```

ZZTDMMAPPER is the special server class SCS uses for the NOSUTIL process that does username mapping during connection request processing.

STATUS PROCESS Command

The STATUS PROCESS command gives status information about the SCS process and server class statistics.

The syntax for this command is as follows:

```
STATUS PROCESS $SCSProcessName  [ , SUB ALL     ]
                                [ , SUB SERVER  ]
                               [ , SUB SESSION ]
```

The following describes the information provided by the options of the STATUS PROCESS command.

STATUS PROCESS $XYZ

The following is an example of the report generated when no options are specified to the STATUS PROCESS command:

```
Process Name    State   Num of Sessions  Num of Classes    Accepting

$XYZ            STARTED              2               3    YES

                                    Connected           Listening
ServerClass              |      Total    High   Current|Current    Hit |
ZZTDMMAPPER              |          1       1         0|      1      1 |
CLASS1                   |          0       0         0|      1      0 |
CLASS2                   |          5       2         2|      3      5 |
```

The various fields are described as follows:

Process Name Name of the SCS process

State Current operational state of the SCS process

Num of Sessions Current number of sessions established with clients

Num of Classes Current number of classes

Accepting Indicates if SCS will accept any more client connections

Server Class Name of a server class defined to SCS

Connected Total Total number of connection requests made to this server class

Connected High Maximum number of servers ever connected to clients

Connected Current	Number of NonStop ODBC Servers connected in that class to clients
Listening Current	Number of servers available for client connections
Listening Hit	Total number of client connection requests satisfied by listening servers

STATUS PROCESS $XYZ, SUB ALL

The ALL option causes the STATUS PROCESS command to display all the information for the PROCESS, SESSION, and SERVER options.

STATUS PROCESS $XYZ, SUB SERVER

The following is an example of the report generated when the SUB SERVER option is specified to the STATUS PROCESS command:

```
Server Class Name ...   DBA_DEFAULT
Server ..............   \TESS.Z011
State ...............   LISTENING
```

The various fields are described as follows:

Server Class Name	Name of the SCS server class to which the NonStop ODBC Server belongs
Server	Name of a NonStop ODBC Server process
State	Current operational state of NonStop ODBC Server

STATUS PROCESS $XYZ, SUB SESSION

The following is an example of the report generated when the SUB SESSION option is specified to the STATUS PROCESS command:

```
Client Name                               State           Num of Servers

@Listener                                 STARTING                     0
```

The various fields are described as follows:

Client Name	Name of the client
State	Operational state of the session
Num of Servers	Number of NonStop ODBC Server processes used by a client session

TRACE PROCESS Command

The TRACE PROCESS command generates a trace of the SCS activity. The trace file can then be examined using the PTrace program. For more information on using PTrace, see the *PTrace Reference Manual* .

The syntax for this command is as follows:

```
TRACE PROCESS $SCSProcessName  [ , STOP [ , BACKUP ]    ]
                               [ { , trace-option } ... ]

trace-option is

   { , BACKUP              }
   { , COUNT count         }
   { , NOCOLL              }
   { , PAGES pages         }
   { , RECSIZE size        }
   { , SELECT  { ALL   }   }
   {           { DATA  }   }
   {           { LOGIC }   }
   { , TO file-spec        }
   { , WRAP                }
```

The TRACE PROCESS command has the following restrictions:

- The BACKUP trace option is not supported.
- The RECSIZE trace option is ignored; it defaults to 256.
- The WRAP trace option is ignored and is the default setting.
- The NOCOLL trace options is ignored and is the default setting.
- A TRACE PROCESS command must always contain either a STOP parameter or a TO trace option.

As an example, you could start a trace with the following SCF command:

```
TRACE PROCESS $XYZ, SELECT ALL, TO $MYVOL.MYSUBVOL.TFILE
```

Unless otherwise directed by the TNSC, always use the ALL attribute of the SELECT option to gather information about a problem.

For general information about SCF and, in particular, additional considerations that might effect use of the preceding SCF commands, see the appropriate SCF manual.

Tracing

This subsection describes the use of the NonStop ODBC Server trace features.

The NonStop ODBC Server provides trace features that can be used to record the following information:

- All input requests received from the client by the NonStop ODBC Server
- All output returned from the NonStop ODBC Server to the client
- All SQL/MP statements generated by the NonStop ODBC Server
- All error and informational messages generated by the NonStop ODBC Server
- The statistics related to statement caching
- The IPC messages between a NonStop ODBC Server and a Pathway server

You can write the log output to a NonStop SQL table, to a terminal, or both.

The Trace Log

The following is an example of a trace log for an ODBC session:

```
ID      DEBUG_FLAG  DEBUG_STRING
------  ----------  -------------------------------------------------------
16      LOGON       SQLTEST_EAP11
16      TIME        1995-01-27 20:41:56
16      IN_STREAM   0C7E  0101 0037 0000 0100 4445 434C 4152   ...7....DECLAR
16      IN_STREAM   0C8C  4520 4331 4339 4520 4355 5253 4F52   E C1C9E CURSOR
16      IN_STREAM   0C9A  2046 4F52 2073 656C 6563 7420 2A20    FOR select *
16      IN_STREAM   0CA8  6672 6F6D 2073 7973 7479 7065 7300   from systypes
16      NSSQL       -CORE-MODE-----------------------------------------------
16      NSSQL
16      NSSQL
16      NSSQL       STATEMENT CACHING:  ACTIVE
16      NSSQL
16      NSSQL       THIS STATEMENT:   CACHED
16      NSSQL
16      NSSQL       SELECT ALL * FROM \TESS.$PUMA.EAP11.ZVUDT FOR STABLE
                    ACCESS
16      NSSQL
16      NSSQL       ---------------------------------------------------------
16      OUT_STREAM  3114  0400 0056 0000 0600 A04B 0003 5549   ...V.....K..UI
16      OUT_STREAM  3122  4408 5553 4552 5459 5045 0856 4152   D.USERTYPE.VAR
16      OUT_STREAM  3130  4941 424C 450A 414C 4C4F 574E 554C   IABLE.ALLOWNUL
16      OUT_STREAM  313E  4C53 0454 5950 4506 4C45 4E47 5448   LS.TYPE.LENGTH
16      OUT_STREAM  314C  0854 4445 4641 554C 5406 444F 4D41   .TDEFAULT.DOMA
16      OUT_STREAM  315A  494E 044E 414D 4508 5052 494E 5446   IN.NAME.PRINTF
16      OUT_STREAM  3168  4D54                                 MT
16      OUT_STREAM  3114  0400 0041 0000 0700 2A36 0000 0000   ...A....*6....
16      OUT_STREAM  3122  0034 0000 0000 3400 0000 0034 0000   .4....4....4..
16      OUT_STREAM  3130  0000 3400 0000 0034 0000 0000 3400   ..4....4....4.
16      OUT_STREAM  313E  0000 0034 0000 0000 3400 0000 00E7   ...4....4.....
16      OUT_STREAM  314C  1E00 0000 0000 EF01 0006             .........
16      OUT_STREAM  3114  0401 0011 0000 0800 FD00 0000 00FF   ..............
16      OUT_STREAM  3122  FFFF FF00                            ...
16      CACHE       Number of Statements: 2
16      CACHE       Number of Lookups: 3
```

```
16      CACHE       Number of Hits: 1
16      CACHE       Number of Drops: 0
```

The trace log contains 3 fields:

- ID. This field is an ID assigned by the NonStop ODBC Server for each user session. The ID is assigned when the first trace record is written to the trace log.

  Note. IDs are randomly assigned and duplicates can occur if the same table is used by many users over a long period of time. Minimize the chances of this happening by using a different trace table for each profile.

- DEBUG_FLAG. This field identifies the type of information being written to the trace log. The following types are supported:

  LOGON The logical username supplied in the logon request.

  TIME The time that the trace was started.

  IN_STREAM Data received from the client. The data is displayed in both hexadecimal and character format.

  OUT_STREAM Data returned by the NonStop ODBC Server to the client. The data is displayed in both hexadecimal and character format.

  INFO An informational message generated by the NonStop ODBC Server.

  ERROR An error message generated by the NonStop ODBC Server.

  NSSQL NonStop SQL/MP statements generated by the NonStop ODBC Server.

  CACHE Information about the statement cache. This information is generated at the end of the session when the client disconnects from the NonStop ODBC Server.

  SP_WRITE The stored procedure IPC request sent from the NonStop ODBC Server to the stored procedure server.

  SP_READ The stored procedure IPC request sent from the stored procedure server to the NonStop ODBC Server.

- DEBUG_STRING. This field contains the information identified by each DEBUG_FLAG that appears in the trace log.

Setting Up Tracing

You can set up tracing in one of two ways: by using the NOSUTIL ADD TRACE statement or by using pass-through statements.

Using the ADD TRACE Statement

The ADD TRACE statement allows you to set up a trace definition, which can then be referenced by a user profile. See “Managing Customized Catalogs,” in the *NonStop*

ODBC Server Reference Manual for a detailed description of the ADD TRACE statement.

The tracing options available through the ADD TRACE statement are as follows:

Option	**Value**	**Description**
TRA_NAME	Y or N	Name of the trace definition.
TRA_LOGTABLE_NAME	*table name*	Name of the SQL table to be used for logging.
LOG_TO_HOMETERM	Y or N	Set to Y to log to terminal.
INPUT_STREAM	Y or N	Set to Y to trace client input.
OUTPUT_STREAM	Y or N	Set to Y to trace NonStop ODBC Server output.
NSSQL	Y or N	Set to Y to trace NonStop SQL/MP statements generated by the NonStop ODBC Server.
TRA_ERROR	Y or N	Set to Y to trace all messages generated by the NonStop ODBC Server.
CACHE_STATISTICS	Y or N	Set to Y to dump cache statistics at the end of a session.
SP_WRITE	Y or N	Set to Y to trace stored procedure execution requests sent to a stored procedure server.
SP_READ	Y or N	Set to Y to trace replies from a stored procedure server.
LAST_UPDATED	date & time	Date and time this row was last updated.

The following is a sample ADD TRACE statement:

```
NOSCOM> ADD TRACE trace4 tra_logtable_name tess_puma_dc.sql_dave.trace4
trace_mode_on y input_stream y nssql y
```

This statement sets up tracing of the input stream and generated NonStop SQL/MP statements to the SQL table TESS_PUMA_DC.SQL_DAVE.TRACE4.

It is important to note that the value specified for TRA_LOG_TABLENAME must be of the form *database.owner.table-name*, where *database* is a valid user catalog or the system catalog. Tracing cannot occur if this table name is incorrect.

Using Pass-Through Statements

You can use pass-through statements from the client to specify the setting of any of the trace attributes described in the preceding tracing options table. Trace attribute values specified in a pass-through statement have the same settings and the same meanings as when they are used in an ADD TRACE statement.

The following is an example of a pass-through statement used to set trace attributes:

```
SELECT "tdm:set tra_log_tablename tess_puma_dc.sql_dave.trace4 input_stream y
nssql y"
```

Any changes made by using the pass-through mechanism are valid only for the duration of the current user session. When the session is terminated, by whatever means, the trace attributes revert to the settings in place when the session was started.

Stopping a Trace

You can turn off tracing by setting the TRACE_MODE_ON flag in the user profile to “N,” as shown in the following example:

SELECT “tdm: set trace_mode_on n”

Tracing is then disabled for any future connections using that profile.

Log Tables

The following describes the use of log tables in conjunction with tracing.

Creating Log Tables

Use the CREATE TRA_LOG statement, as described in the *NonStop ODBC Server Reference Manual*, to create a trace log table.

Displaying Log Tables

You can display a log table by issuing a normal SELECT statement from the log table, as shown in the following example:

```
SELECT * from tess_puma_dc.sql_dave.trace4
```

Note. You must *not* select from the log table from a client logged on under a user ID that has tracing enabled and the OUTPUT_STREAM option set to Y. To do so causes the NonStop ODBC Server to enter an infinite loop writing to the log table. If you need to display the contents of the log table from such a client, you must first issue the pass-through statement to disable tracing, as follows:

```
SELECT "tdm:set trace_mode_on n"
```

Clearing Log Tables

You can clear a log table by deleting all rows from the log table by using a normal DML statement from the client, as shown in the following example:

```
DELETE from tess_puma_dc.sql_dave.trace4
```

Dropping Log Tables

Use the DROP TRA_LOG statement, as described in the *NonStop ODBC Server Reference Manual*, to drop a trace log table.

Logging to a Terminal

If you want to log to a terminal, set the LOG_TO_TERMINAL trace option to Y. When tracing is enabled (by setting the TRACE_MODE_ON option), using the LOG_TO_TERMINAL option causes the trace information to be sent to the terminal used to start the SCS process.

Note. You *must* pause the terminal used for displaying the trace before starting the trace; otherwise, the NonStop ODBC Server hangs, waiting to output trace information. In addition, you should not send trace output to a workstation running an application sending requests to NonStop ODBC Server. For example, in a Windows environment, you should not use one TTE window for tracing and another to run Microsoft Access to send requests to the NonStop ODBC Server.

Displaying Trace Options

You can display the trace options currently in effect after a client has logged on the client by use of the pass-through statement SHOW:

```
SELECT "tdm: show *"
```

The following is an example of the output of this statement:

```
       Show Value                         Current Setting
----------------------------   --------------------------------------
 USER_NAME                          SQL_DAVE
 PROFILE_NAME                       DEFAULT
 DATABASE_NAME                      MASTER
 SQL_MAX_STATEMENT_CACHE            0
 SQL_ACCESS_MODE                    RW
 SQL_CURSOR_MODE                    RW
 SQL_DIALECT                        TSQL
 SQL_MAX_ROWS                       0
 SQL_MAX_STATEMENT_COST             0
 SQL_TXN_ISOLATION                  1
 SQL_UNSUPPORTED                    ERROR
 TRACE_MODE_ON                      YES
 LOG_TABLENAME                      TESS_PUMA_DCTEST.LOG
 LOG_TO_TERMINAL                    NO
 INPUT_STREAM                       YES
 OUTPUT_STREAM                      NO
 NSSQL                              YES
 TRACE_ERROR                        NO
 CACHE_STATISTICS                   NO
 SP_READ                            NO
 SP_WRITE                           NO
```

Developing Custom Applications

The NonStop ODBC Server catalog tables are normal SQL tables, which are themselves registered in the NonStop ODBC Server catalog tables. Therefore, it is possible to access the catalog tables directly from a client by using a third-party tool or by developing custom applications or scripts.

You should not develop applications or use tools to update the NonStop ODBC Server catalog tables directly, however. The reason for this is that the integrity rules defined for the catalog tables are enforced only by the NonStop ODBC Server products (NonStop ODBC Server and NOSUTIL), and end-users are not aware of these rules. It is also possible that the integrity rules or the NonStop ODBC Server catalog structure might change with future releases, which would could cause end-user applications to fail or to corrupt the NonStop ODBC Server catalog tables.

It is possible for you to use tools or develop applications to display information in the NonStop ODBC Server catalog tables. There are two considerations in this area:

- The tool or application should not update the NonStop ODBC Server catalog tables directly, as integrity rules in the catalog must be maintained. Instead, it should use pass-through statements to invoke NOSUTIL to make the necessary changes.
- The schema and semantics of the NonStop ODBC Server catalog should be expected to change in the future to meet new requirements. Consequently, the tool or application must be prepared to evolve as necessary. The NonStop ODBC Server supports making versions of the catalogs so you can discern when changes occur. The NonStop ODBC Server system catalog table ZNSDB, which contains an entry for each catalog on the system, records the version of the catalog in the T_VERSION column. For the D43 release, the catalog version number is 200.

5

Managing NonStop ODBC Server Resources

This section discusses the management of NonStop ODBC Server resources, and gives suggestions for allocating resources and handling usage.

Customers often need to run lengthy and resource-intensive queries. The need to manage system use often increases when system resources are shared by a mixed workload of small queries and large batch-like queries.

To manage the system resources optimally for such queries, the NonStop ODBC Server provides two facilities to monitor and control system resource use.

- The resource accounting facility provides the ability to record the resource usage for satisfying the end-user database requests.
- The resource governing facility provides the capability to control the resources used by SQL statements. For example, a limit can be set for the amount of resources to be used for a single query; if the limit is exceeded, the query can be stopped.

Resource Accounting

The NonStop ODBC Server resource accounting facility collects metrics used to estimate resource usage and, thereby, to predict resource cost. Data is written directly to an SQL table. Two types of metrics are collected: CPU time and SQL statement statistics from SQL/MP.

The NonStop ODBC Server resource accounting facility can be turned on or off and can be configured to record the data at the end of each statement or at the end of a user session. These options are defined as part of a user profile.

Resource Governing

Management of system resource varies depending on the customer’s resource-usage objectives. At times, limiting the resources for a particular user might be sufficient; at other times, setting priorities for resource use might be the best decision.

One customer might decide that all simple queries should be run at a higher priority than medium queries, and medium queries should be run at a higher priority than complex queries. The customer might specify that a simple query is one that has a total cost less than 5,000, a medium query is one that has a cost in the range of 5,000 to 200,000, and a complex query is one with a cost greater than 200,000.

Another customer might like to be able to have the system set the priority based on the estimated query cost and to adjust this as needed at run time if the query exceeds a certain cost.

Many customers might want to be able to stop queries that exceed user-specified thresholds (for example, CPU time or number of rows fetched) and to have an explicit interface to stop a given query.

The solution is to implement a resource governing facility in the NonStop ODBC Server product to monitor and govern the SQL statement execution based on a predefined policy.

Previously, two functions were available in the NonStop ODBC Server to govern SQL statements:

- The ability to define a statement cost limit for a DML statement and prevent executing of this statement if it exceeds the cost predicted by the SQL/MP optimizer
- The ability to define the maximum number of rows to be returned to the client for a SELECT statement

The resource governing facility adds to those functions the following capabilities:

- The ability to set priorities for resources
- The ability to limit I/O resources for a single query
- The ability to limit the amount of time spent on a single query
- The ability to display query status

The Resource Accounting Facility

The configuration that enables resource accounting is defined in the user profile table, ZNSPROF. Resource accounting data is recorded in the ZNUMTRX table. The configuration data consists of the Resource Accounting Profile and the Resource Accounting Log Table. Attributes are defined following:

Resource Accounting Profile

The resource accounting profile is defined in the following three attributes of the user profile table:

ACC_MOD_ON Turns resource accounting on or off.

Y Accounting is on for this user
N Accounting is off for this user (default).

ACC_LOGTABLE_NAME A name of the form *database.owner.*ZNUMTRX. The accounting log table is created using the CREATE ACC_LOG statement.

ACC_LEVEL Specifies whether data is to be recorded on a per-session or per-statement basis:

SESSION Data is recorded for one session.
SQL_STATEMENT Data is recorded for each statement.

Resource Accounting Log Table

Resource data is recorded in the resource accounting log table specified by the ACC_LOGTABLE_NAME entry in the user profile. The following describes the columns in an accounting log table and their meanings:

CLIENT_ID A unique identifier, such as an IP address, to identify the client connection.

SESSION_ID A unique identifier for this user session.

START_TIME Julian timestamp when statement execution began.

END_TIME Julian timestamp when statement execution ended.

LOGON_USERNAME Client logon name—can be a username or alias.

APPLICATION Application ID, used to determine session environment.

G_USERNAME Guardian group and user name.

PROFILE_NAME User profile used for this session.

SER_NAME Server class name for this session.

NODE_NAME Node name in which the NonStop ODBC Server is running.

CPU_PIN Process ID for the NonStop ODBC Server.

START_PRIORITY
Priority under which the NonStop ODBC Server is running.

ACC_LEVEL
Recording level:

SESSION per-session basis
SQL_STATEMENT per-statement basis

IN_MESSAGES
Number of input messages from the SCS process read by the NonStop ODBC Server process.

IN_MESSAGE_B
Total input message bytes read by the NonStop ODBC Server.

OUT_MESSAGES
Number of output messages written to the SCS process by the NonStop ODBC Server process.

OUT_MESSAGE_B
Total output message bytes written by the NonStop ODBC Server.

SCS_EXECUTION_TIME
Time (milliseconds) used by the SCS process in managing the session during the START_TIME to END_TIME interval.

NSODBC_EXECUTION_TIME
Execution time (milliseconds) used by the NonStop ODBC Server during the start-end interval. Includes calls to the file system and other external processes.

NSODBC_ELAPSED_TIME
Time (milliseconds) elapsed under NonStop ODBC Server control during the start-end interval. The time the system is working on the request, including internal wait times. The difference between this and NSODBC_EXECUTION_TIME is time spent in the NOS process.

NSSQL_EXECUTION_TIME
SQL/MP compilation and execution time in milliseconds during the start-end interval. Includes calls to the file system and other external processes.

NSSQL_ELAPSED_TIME
Time (milliseconds) elapsed under SQL/MP executor control during the start-end interval. The time the system is working on the request, including internal wait times. The difference between this and NSSQL_EXECUTION_TIME is internal processing time.

STMT_TYPE

Type of statement being processed, as follows:

S	SELECT statement
D	DELETE statement
U	UPDATE statement
I	INSERT statement
L	DDL statement
C	Control statements
O	Others

STMT_STATUS

Status of statement execution, as follows:

C	Statement execution completed
P	Statement prepared
X	Statement canceled by user
Q	Statement encountered a SQL/MP error

STMT_ORIGIN

Indicates whether the statement was issued by the client, as follows:

C	Statement issued by the client
I	Statement was generated internally by the NonStop ODBC Server

ESTIMATED_COST

Estimated cost for the statement, returned by the SQL/MP optimizer.

STMT_CACHED

Indicates whether the statement is being cached, as follows:

N	Not cached
Y	Cached
R	Cache is being used (cache hit)

RECORDS_ACCESSED

Number of records accessed (SQL/MP).

RECORDS_USED

Number of records returned or altered (NonStop SQL/MP).

DISK_READS

Number of disk reads and writes (NonStop SQL/MP).

MESSAGES

Number of messages sent to the disk process (SQL/MP).

MESSAGE_BYTES

Number of bytes in all messages sent to the disk process.

LOCK_WAITS

Number of times the executor waited for a lock request to finish (SQL/MP).

LOCK_ESCALATIONS

Number of times a record lock was escalated to a file lock.

LAST_GOV_ATTRIBUTE
Indicates this entry was generated by the triggering of a governing action rather than a change of statement status.

LAST_LIMIT_VALUE
The value of the governing attribute.

LAST_GOV_ACTION
Action taken when the governing attribute limit was exceeded.

STMT_TEXT
Statement text (up to 3000 bytes).

The resource accounting log table is created with the following file attributes and defaults:

NO AUDIT
BUFFERED
EXTENT (16,64)
MAXEXTENT 160
ENTRY-SEQUENCED

The EXTENT and MAXEXTENT size can be configured, but for performance reasons the NO AUDIT and BUFFERED attributes should not be changed.

The following is an example of two rows from an accounting log table, the first recorded on a per-statement basis and the second recorded on a per-session basis:

```
CLIENT_ID           SESSION_ID          START_TIME            END_TIME
LOGON_USERNAME      APPLICATION         G_USERNAME            PROFILE_NAME
USER_NAME           NODE_NAME           CPU_PIN               START_PRIORITY
ACC_LEVEL           IN_MESSAGES         IN_MESSAGE_B          OUT_MESSAGES
OUT-MESSAGE_B       SCS_EXECUTION_TIME  NSODBC_EXECUTION_TIME NSSQL_EXECUTION_TIME
NSODBC_ELAPSED_TIME NSSQL_ELAPSED_TIME  STMT_TYPE             STMT_STATUS
STMT_ORIGIN         ESTIMATED_COST      STMT_CACHED           RECORDS_ACCESSED
RECORDS_USED        DISC_READS          MESSAGES              MESSAGE_BYTES
LOCK_WAITS          LOCK_ESCALATIONS    LAST_GOV_ATTRIBUTE    LAST_LIMIT_VALUE
LAST_GOV_ACTION     STMT_TEXT

130.252.135.92      47                  21665866216443676     21665866234013070
codbc               MSACCESS            SQL.ODBC              coreprof
class2              \TESS               007,180               155
SQL_STATEMENT       1                   27                    3
505                 0                   44724                 16271
686406              159398              S                     C
U                   2                   Y                     9
9                   4                   4                     708
0                   0
                    SELECT ALL * FROM \TESS.$SQL.NTDB.TP2 FOR STABLE ACCESS

130.252.135.92      48                  21665866181258587     21665866257208422
codbc               MSACCESS            SQL.ODBC              coreprof
class2              \TESS               007,180               155
SESSION             5                   621                   7
810                 0                   59508                 49511
1135178             299412              135                   25
13                  11                  3904                  0
0
```

Collecting Resource Accounting Data

If resource accounting is on, collection of metrics begins when the client connects to the NonStop ODBC Server. All subsequent compilation and execution of SQL statements is recorded.

If the recording is on a per-statement basis, every statement is recorded; at the end of the user session, a summarized set of statistics is recorded as the session metrics.

If the recording is on a per-session basis, only the summarized session metrics are recorded.

The complete metrics include the process time of the NonStop ODBC Server, the process time in SQL/MP, the I/O cost (run-time statistics) from SQL/MP execution, and the number of messages and message bytes between SCS and the NonStop ODBC Server, as follows:

- NonStop ODBC Server process time includes the following:
  - NonStop ODBC parses and translates a SQL statement.
  - It reads the NonStop ODBC catalog to map the logical table name.
  - NonStop ODBC issues a PREPARE statement to invoke SQLCOMP to compile the query. The estimated cost and compiled time are collected separately.
  - NonStop ODBC converts the data according to client data types and returns the results.

- SQL/MP execution cost includes only the master executor time.

  Currently, the SQL/MP master executor does not record the process time of its child processes. To obtain a reasonable set of data for cost estimation, both the run-time statistics generated by the SQL/MP executor and the process time in the NonStop ODBC Server components must be recorded. The statistics can then be used to predict resource cost for an SQL statement.

- SQL/MP statement execution includes the following run-time statistics:
  - Records accessed
  - Records used
  - Disk reads
  - Message counts
  - Message bytes
  - Number of lock waits (concurrency cost)
  - Number of times record locks are escalated to file locks
- Gateway messages and message bytes between the SCS process and the NonStop ODBC Server are also collected.

Certain queries require temporary tables to be used by the SQL/MP executor. The statistics returned in the SQLSA contain the resource used in processing both the SQL tables and the temporary tables. Therefore, the total rows accessed can at times appear to be greater than the number of rows in the SQL tables.

When a transaction is rolled back, the resource accounting data for the statements executed is still recorded; it is not backed out.

Usage Considerations

The following subsections describe things to be considered when using the resource accounting facility.

Timing of Resource Accounting

The resource accounting data includes all types of statements. Depending on how the resource accounting is configured, however, this data can be written out either at the end of a session (per-session basis) or after each statement is executed (per-statement basis).

Record Data for an Entire Session

If a per-session basis is specified for the ACC_LEVEL attribute, all metrics are collected and accumulated after each statement execution, but data is written to the table only once, at the end of the session when the client disconnects. Certain data, such as CACHED, STATEMENT_TYPE, and STATEMENT_TEXT, is blanked out.

Such items are meaningless in the context of a session. Use per-session basis if the resource accounting data is to be used for charge-back.

Record Data for Each Statement

If a per-statement basis is specified for the ACC_LEVEL attribute, all metrics are collected and written to the table at the end of each statement execution. Both statements issued by the client and statements generated by the NonStop ODBC Server are recorded. A per-statement basis is recommended if the resource accounting data is to be used to monitor query performance or to identify use patterns.

Statement Handling

Client applications used with the NonStop ODBC Server can contain CORE-SQL statements, TSQL statements, or SQL/MP statements in pass-through mode. DML statements are translated to SQL/MP DML statements, and the resource use is collected by the NonStop ODBC Server.

When a CORE-SQL SELECT statement is executed with the cursor mode set to READ_ONLY or READ_ONLY_STREAMING, it is executed in a select loop such that, after the statement is prepared, the cursor is opened, rows are fetched until there is no more data, then the cursor is closed. The resource usage data is collected at the end of the session for the SELECT statement.

TSQL batch statements are treated as individual statements, and the resource usage data is collected per- each statement, for example:

```
select *from tl
insert into tl values ...
go
```

The data for the SELECT statement is collected and recorded separately from that for the INSERT statement.

When a CORE-SQL SELECT statement is executed with the cursor mode set to FOR_UPDATE, each row is fetched only when the SQL_FETCH statement is invoked. DML statements such as DELETE or UPDATE WHERE CURRENT OF are allowed among the SQL_FETCH statements. The resource usage data for each SQL_FETCH statement and the data generated internally by the OPEN CURSOR statement is accumulated to produce one set of resource usage data for the SELECT statement. The resource usage data for the DELETE or UPDATE statement is recorded separately, for example:

```
SQL_EXECUTE select * from tl
SQL_FETCH
SQL_FETCH
SQL_EXECUTE delete from tl where current of
SQL_FETCH
SQL_EXECUTE update from tl where current of
```

In this scenario, three sets of resource usage data are generated. The first set is for the SELECT statement, consolidating data from the open cursor and all individual

SQL_FETCH statements. The second set is for the DELETE statement, and the third set is for the UPDATE statement.

The driver translates ODBC functions that retrieve catalog information, such as SQLColumns, and TSQL functions such as DBNAME, to SQL/MP SELECT statements; resource use for execution of such SELECT statements is collected also.

For a DDL statement, only the NonStop ODBC Server process time is collected, because in SQL/MP the process time returned by SQL Executor does not include the Catalog Manager's process time. No resource statistics are being gathered by the Catalog Manager.

Managing the Resource Accounting Configuration

A full set of utility statements is available to create a resource accounting log table and to set up and change the accounting profiles. Dynamic reconfiguration and managing of the resource accounting table is also provided.

Only the database administrator (a member of the super group) has the authority to establish and change the resource accounting configurations. This restriction is because the data is used by the database administrator to perform administrator functions such as chargeback to client departments and monitoring system use.

No SET command is allowed to alter the accounting configurations. The reason is that the SET command is issued by each user to change the run-time behavior; and the individual user does not have the authority to change the accounting configuration set up by the database administrator.

Managing the Resource Accounting Table

Only the database administrator has the authority to create or delete a resource accounting log table.

Managing Resource Accounting Data

A resource accounting log table must have sufficient space to accommodate the amount of data being collected, especially the data for each SQL statement. When the table becomes full, a message is written to $0, and resource accounting is disabled. It is the responsibility of the client and the database administrator to determine whether to continue—with the collection of resource accounting data suspended—or to free up enough space to resume data collection.

In an overflow situation, the database administrator could choose to perform following:

- Alter the MAXEXTENT size of the accounting log table (if it is below the maximum limit).
- Create a new accounting log table in a different location and change all user profiles to point to the new table.
- Purge data from the existing table (if the report has been generated) and continue.

- Continue, realizing that accounting data will not be recorded for any subsequent statements.

It is recommended that the database administrator create a tool (such as a system-stored procedure) either to summarize the data to another history table or to archive it to a tape. The database administrator should periodically generate the resource accounting report and run this tool to clean up the ZNUMTRX table to prevent an overflow situation.

Security Considerations

Resource accounting data is stored in SQL tables, which means that data is easily accessible. To protect any users from inadvertently updating the data, these tables are owned by the database administrator and are write/purge protected. Tools or clients with read access should be able to generate reports by using the data.

The catalog table for storing the resource accounting profile is owned by the database administrator and write protected so only the database administrator can change these specifications.

Performance Considerations

There is minimal effect on performance when resource accounting data is being recorded on a per-session basis. If the data is being recorded on a per-statement basis, however, it incurs a cost of one extra write for every SQL statement. These INSERT statements for recording the resource accounting data do not appear in the ZNUMTRX table.

It is important to note that turning tracing on can trigger many INSERT statements and could generate a large amount of data written to the resource accounting table.

Error Handling

Resource accounting errors are not critical to the execution of user applications. Such an error is treated as a warning, and a message is logged to the EMS $0 as part of the NonStop ODBC Server error messaging scheme.

Limitations

It is important to note that the metrics to be collected can be derived only from data currently available. Because of limitations in SQL/MP, the following metrics are not available:

- Process time of the master executor's child processes, such as individual external stored procedures and SORTPROGs
- I/O statistics from SORTPROGs
- Process time and I/O statistics of the catalog manager from execution of a DDL statement

Resource Governing Facility

Setting the priority of SQL statements to influence resource use can provide some benefit to a multiuser environment. Methods of balancing and controlling the use of system resources vary from system to system. As a system administrator, you should experiment with this facility and adjust resources accordingly to best meet your system needs. You can always reconfigure the priority settings in the NonStop ODBC Server catalog by using utility statements.

The data provided by the resource accounting facility could be used in conjunction with the resource governing facility to help define resource governing. Accounting data provides information about resources used over time.

The resource governing facility applies only to SQL statements generated directly by the user, not to SQL statements generated internally by the NonStop ODBC Server product. Queries to look up and map the NonStop ODBC Server object name to the SQL/MP object name, for example, are not covered.

Architecture of the Resource Governing Facility

The NonStop ODBC Server resource governing facility consists of the following components:

- The capability to allow the system administrator to set up a query scheduling policy and a run-time governing policy by defining the limits and the trigger actions.
- The capability to use the scheduling policy to determine how to run a query after a statement has been compiled but before the statement is executed.
- The capability to use the run-time governing policy to govern the query execution and to take action when a query exceeds its limit.
- The capability to record currently active user queries in a query status table.

The limits are defined, through governing policies that are then referenced from a user profile. The NonStop ODBC Server retrieves the governing policies at user logon time and uses the predefined policies to limit each SQL statement execution.

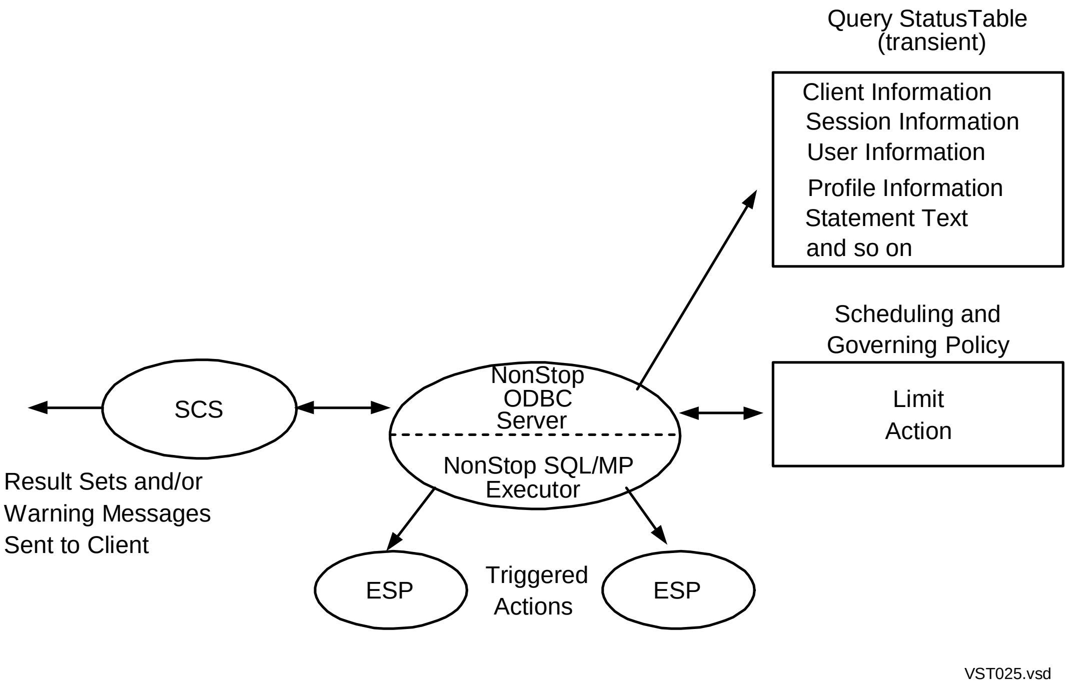
All scheduling and governing actions are enforced by the NonStop ODBC Server. A NonStop ODBC Server process can enforce governing only when it has control of the process. When SQL/MP is processing a query, control may not be returned for an extended period of time; during this time, no governing can take place. In this case, using the query status table is the best way to monitor those queries and then manually intervene as needed, such as stopping a NonStop ODBC Server process.

A status record is written to a NonStop ODBC Server query status table at the time a query starts. There is one record for each user SQL statement being executed. When the query is finished, this record is deleted from the table. If you do not require resource managing, you can configure the NonStop ODBC Server to not record any status in the status table. You should probably generate the status record only for large queries with an estimated cost greater than *n* (some maximum amount determined by the individual user). For small queries, the status information is not essential because the time to execute such a query is not significant. By limiting the entries in the query status table, you can eliminate unnecessary SQL writes and deletes to the query status table.

During run time, the NonStop ODBC Server periodically checks whether a query has exceeded its defined limits and initiates an appropriate action.

Figure 5-1 shows the relationships among the NonStop ODBC Server, the governing policy table from which it obtains governing policies, and the query status table to which it logs query status data, as well as the SQL/MP executor and the actions it initiates based on information from the NonStop ODBC Server.

Figure 5-1. Resource Governing Facility Architecture



Defining Policies

Resource governing is controlled through the use of governing policies. A governing policy defines a set of limits and the actions to be performed when a given limit is exceeded. Three utility statements have been provided for maintaining these policies: ADD, MODIFY, and REMOVE GOVERNING. Resource governing is enabled or disabled through the user profile. A single policy can be established for all users or different policies established for individual users.

User Profile

Three entries in the user profile associate a user with a resource governing policy. The relevant user profile attributes and their meanings are as follows:

GOV_MODE_ON Specifies whether governing policy is in effect.

GOV_NAME Name of the governing policy that controls the query execution (if GOV_MODE_ON is Y (yes)).

QST_LOG_ON Specifies whether to log all statements. If GOV_MODE_ON is not Y, no status is generated.

Governing Policies

The following describes the attributes associated with governing policies:

GOV_NAME Name of the governing policy that controls a query execution.

GOV_LEVEL Specifies whether governing is done on a per-statement basis or a per-session basis. In this version of the NonStop ODBC Server, all governing is done on a per-SQL-statement basis.

GOV_ATTRIBUTE Attribute to which a governing action can be applied or one of the following:

ELAPSED_TIME
EXECUTION_TIME
ROWS_ACCESSED
ROWS_USED
ESTIMATED_COST

LIMIT_VALUE Threshold (lower bound) to be used to limit the amount of resources used for a query.

GOV_ACTION — Action to be taken when the limit value is exceeded, or one of the following:

PRIORITY + + *n*	Increase the priority by *n*.
PRIORITY – – *n*	Lower the priority by *n*.
PRIORITY *n*	Set the priority to *n*.
COMMIT”	Stop and commit the transaction.
ROLLBACK	Stop and roll back the transaction.
CONTINUE	Continue without any changes.
STOP	Do not execute the statement; this action can be used only with estimated cost.

QST_LOG_ON — Specifies whether to log the statement. Used only with estimated cost attributes. This entry allows you to log query status for only those queries with a given cost range.

If QST_LOG_ON is set in the user profile (ZNSPROF), causing every statement to be logged, QST_LOG_ON in the user profile overrides the QST_LOG_ON attribute in the governing policy.

LAST_UPDATED — Timestamp when this table was last updated.

The following example illustrates how resource policies are set. User SQL_ODBC uses COREPROF as a user profile; user profile COREPROF has been configured to use Policy-A as the scheduling policy (by means of estimated cost) and governing policy.

User profile:

```
PROFILE_NAME   GOV_MODE_ON   GOV_NAME   QST_LOG_ON
coreprof       Y             Policy-A   N            (do not log all statements)
```

Governing policy:

```
GOV_NAME   GOV_ATTRIBUTE    LIMIT_VALUE   GOV_ACTION      QST_LOG_ON
Policy-A   estimated_cost   500           priority --10   N
Policy-A   estimated_cost   3000          priority --30   Y
Policy-A   estimated cost   5000          stop            N
Policy-A   elapsed time     10 min.       commit
Policy-A   rows accessed    10000         priority --30
```

Scheduling A Query

There are two ways to schedule a query:

- A query can be launched with a different priority.
- A query can be prevented from running if its estimated cost exceeds the predefined cost threshold.

These approaches for scheduling queries are defined by using the estimated cost attribute.

You can use multiple cost ranges (ESTIMATED_COST values in the preceding example) to categorize queries into different cost groups—such as small, medium, and large queries—and specify an action to take effect for each cost range.

When a statement is prepared, the NonStop ODBC Server determines what action to take based on the estimated cost returned by the SQL/MP optimizer and then performs it before the statement is executed.

When multiple cost ranges are defined, the action to take is associated with the lower bound of the cost. To illustrate this, suppose the same governing policy settings apply as used in the previous example:

```
GOV_NAME   GOV_ATTRIBUTE    LIMIT_VALUE   GOV_ACTION       QST_LOG_ON
Policy-A   estimated_cost   500           priority --10    N
Policy-A   estimated_cost   3000          priority --30    Y
Policy-A   estimated_cost   5000          stop             N
```

Suppose the current priority is 150, that a statement such as

```
SELECT * FROM T
```

has an estimated cost of 1000, and a statement such as

```
SELECT * FROM S WHERE ...
```

(a more complex statement) has an estimated cost of 8000.

In this scenario, SELECT * FROM T runs at priority 140. Because its cost is 1000, which is between 500 and 3000, the action to take is defined by the limit value of 500, which is to lower the current priority (150) by 10. Because the cost of SELECT * FROM S WHERE … is greater than 5000, the action specified is to not execute the statement at all.

Governing

At run time, the resource governing facility controls resource use by either stopping query execution or by changing the priority of a query. You as system administrator can limit the time (CPU time or elapsed time) permitted for a statement or limit the I/O consumed for a statement.

Resource governing can be defined by specifying the governing attributes to be limited and the amount of resource to limit in the governing policy table ZNSGOV. Multiple limits and actions can be defined, but only one action is triggered. When the first limit is exceeded, the governing action associated with the first attribute is triggered (it is an OR condition, not an AND condition). A warning message is returned to the client stating that a particular governing action is in effect. If the action is to change the priority of a query, an additional EMS message is logged to $0.

The following example illustrates how the resource policies are enforced. Suppose the initial priority is 150.

User profile:

```
PROFILE_NAME   GOV_MODE_ON     GOV_NAME        QST_LOG_ON
coreprof       Y               Policy-A        N
```

Governing policy:

GOV_NAME	GOV_ATTRIBUTE	LIMIT_VALUE	GOV_ACTION	QST_LOG_ON
Policy-A	estimated_cost	500	priority --10	N
Policy-A	estimated_cost	3000	priority --30	Y
Policy-A	estimated_cost	5000	stop	N
Policy-A	elapsed_time	10 min.	commit	
Policy-A	rows_accessed	10000	priority --30	

When user SQL_ODBC logs on to the system, both scheduling and governing policies are enforced according to the setting of Policy-A, and the user profile specifies not to log status for every statement. After the statement has been prepared, because it has an estimated cost of 1000, the statement is scheduled to run with a new priority of 140 (lowered by 10). This statement does not generate a query status record, because the QST_LOG_ON flag is set to “N” for this cost range.

During the execution of this statement, if the elapsed time exceeds 10 minutes, the query is stopped, and the transaction is committed. A message of “query stopped because the elapsed time limit is exceeded” is returned to the client, as are the rows of data fetched.

If 10,000 rows have been accessed in less than 10 minutes, priority is lowered by 30, and the query is run with priority 110 until completion. A message of “priority for this query has been changed to 110” is returned to the client. In this release, this governing action can be applied to the query only once. No further limit is applied.

Query Status Records

The query status record can be used to manually monitor and manage a long-running query. It provides information such as which statements are currently running, when did the statement start, and who is running a particular statement. You can specify that query status is to be logged by using the QST_LOG_ON attribute for user profiles and governing policies.

In the case of governing runaway queries, by using the statement-started time in the query status table, a user with the proper authority, or a user-supplied governing process, would be able to stop the query by terminating the NonStop ODBC Server process.

If resource accounting has been configured, after the query is completed, you can find the query with more details logged in the ZNUMTRX metrics table, as described previously for the resource accounting feature.

Query Status Table

There is one query status table, ZNUQST, that is active at any given time in a system. The table location is specified by the QST_LOGTABLE_NAME attribute in the system configuration table, which is built by the CREATE QST_LOG command.

The table entries and their meanings are as follows:

CLIENT_ID	A unique identifier, such as an IP address, to identify a client
SESSION_ID	A unique identifier associated with a session; same as SESSION_ID in ZNUMTRX
LOGON_USERNAME	Client logon name
G_USERNAME	Guardian user name
PROFILE_NAME	User profile name
SER_NAME	Server class name
NODE_NAME	Node name
CPU_PIN	CPU and PIN for the NonStop ODBC Server
START_PRIORITY	NonStop ODBC Server’s original priority
CURRENT_PRIORITY	Priority at which the query is running
TRANSACTION_ID	TMF transaction ID; absence of a transaction ID means no active transaction is associated with this statement
STMT_TYPE	Indicates the type of the statement by using same codes as in ZNUMTRX table
ESTIMATED_COST	Estimated cost for this statement
STMT_TEXT	Partial statement text, up to 500 bytes

Timing of Query Status Records

A query status record is written to the ZNUQST table when a statement is executed. This status is logged if the user profile is configured to log all statements or if there is a governing policy configured to log statements of a particular cost range. If for any reason a status record cannot be written, a warning message is sent to EMS $0 log and also to the user; execution of the query continues, however.

The query status table hold records temporarily while a query is running. After statement execution finishes, the status record is deleted. The NonStop ODBC Server deletes status records for a user session at the end of user session to avoid monitoring obsolete data and governing a query that is no longer running. Only if the NonStop ODBC Server terminates abnormally could there be records left from queries no longer running. The NonStop ODBC Server always initializes the query status table at start-up time by checking to see if records with the same CPU and PIN exist and removes them from the table.

Usage Considerations

When the user logs on, the NonStop ODBC Server reads the user profile to find out whether this user needs to be governed. If a governing policy exists, the NonStop ODBC Server retrieves the policy and actions to be used later. If governing is disabled or the policy name is omitted from the user profile, or if there is a problem retrieving the governing policy, all queries will be executed as if there were no limit to be applied. These configurations could be dynamically reconfigured according to the current dynamic reconfiguration rules.

What Statements Can Be Scheduled

When the statement caching is on, query scheduling policies apply to all user-issued SQL DML statements. When caching is not on, scheduling policies are applied to all SELECT statements and to user-prepared INSERT, UPDATE, and DELETE statements only. Internally, SQL statements prepared by the NonStop ODBC Server, such as EXECUTE IMMEDIATE statement, are governed.

When Scheduling Action Takes Effect

After a statement has been compiled, the resource governing facility determines what action to take based on the estimated cost. Action to change the priority or to prevent the statement from running takes effect before the statement is executed. If a change of priority is performed, the priority is restored at the end of the query to its original value set by SCS. If an action is applied, a message is returned to the client to report that action. As this is an informational message only, the client application could report the message or could choose to ignore it.

If no estimated cost is specified in the governing policy, or if a particular cost range is omitted, queries are run with the current environment setting. If for any reason the priority cannot be changed, a warning message is written to the EMS log, but the query continues to run.

When the Run-Time Governor Takes Effect

The resource governing facility governs query execution by recording time and I/O consumed by the query. The facility periodically checks to see if one of the limits specified has been exceeded. If it finds such a case, it triggers the action defined by the governing policy to either stop the statement execution, if possible, or lower the priority. As with scheduling, if the priority has been changed, the priority is restored at the end of the query to the original priority set by SCS.

All user prepared statements are governed. In this release, the governing policy can be triggered only once; after reaching one of the limits, the query is run with new priority (or stopped), and the resource governing facility does not manage this query again.

Controlling Runaway Queries

As mentioned previously, the NonStop ODBC Server might occasionally have no control over a runaway query. In this case, you can find the query by using the query status table and manually stop it by terminating the server process.

Note that ExecuteDirect statements (which are executed through the SQLExecDirect API call), for UPDATE, DELETE, or INSERT, are not prepared beforehand by the NonStop ODBC Server. Consequently, if you did not set up the configuration to log every statement, no status records are generated for these statements. Therefore, no information will be available to manually govern a runaway ExecuteDirect statement such as INSERT...SELECT * FROM.... HP recommends you always use SQLPrepare and SQLExecute on long-running ExecuteDirect statements other than SELECT statements, which you can do only from a scripting or programming language. Some tools may not give you the ability to control which ODBC API calls are performed. You can set up the configuration to log every statement, but be aware that this incurs a large amount of overhead writing and deleting every status record.

The Transaction Model

When AUTOCOMMIT is on, the NonStop ODBC Server starts and ends a transaction for every statement. If statement execution exceeds the limit and execution is stopped by the resource governing facility, the NonStop ODBC Server automatically commits or rolls back the transaction, depending on how the governing action is defined.

When AUTOCOMMIT is off, the ODBC driver explicitly starts a transaction each time an application submits a transaction and no transaction is open. This transaction can consist of one or more SQL statements. If one of the SQL statements exceeds a limit and the associated governing action is to stop the execution and either commit or roll back the transaction, the NonStop ODBC Server does not end or abort the transaction, but sends a warning message to the user along with the result sets. The warning message indicates the recommended action from the policy; the user must then explicitly commit or roll back the transaction. This behavior is the same for a Sybase application that starts a user transaction.

The only situation in which the NonStop ODBC Server does not start a transaction is when the transaction isolation is set to SQL_TXN_READ_UNCOMMITTED, AUTOCOMMIT is set to off, and the SQL statement is a SELECT.

Effect on SET commands

Setting SQL_MAX_STATEMENT_COST is disallowed. The estimated cost is now defined in the scheduling policy and cannot be changed by using the SET command. This attribute can be changed using the utility statement MODIFY GOV; it follows the dynamic reconfiguration rule.

Setting SQL_MAX_ROWS affects the rows-fetched limit and overrides the limit specified in the governing policy.

Limitation

The NonStop ODBC Server supports prepare-and-execute operations, and a client can work on multiple concurrent statements. However, the request for changing priority is done only once; the new priority persists until the execution of the given statement is finished. The reason for this restriction is to minimize the swapping of priorities between multiple statements.

An example of priority changing is shown following:

In the following example, the initial priority is 150:

```
Prepare s1 from select * from t1    (scheduling action specifies lowering priority by
                                    10 for s1)
Prepare s2 from select * from t2    (scheduling action specifies lowering priority by
                                    30 for s2)
Declare c1 cursor for s1
Declare c2 cursor for s2
Open c1                             (priority lowered to 140 for s1)
Fetch cl
Open c2                             (priority is still 140 for s2)
Fetch cl
Fetch c2
Close c1                            (priority is restored to 150 for s1)
Fetch c2
Close c2
```

A

Summary of System Installation Defaults

Table A-1 lists the values stored in the ZNSSCFG table during system installation. Each value in ZNSSCFG is stored in ASCII VARCHAR format in the VALUE column. When a system default value is used in other tables (such as ZNSPROF, ZNSSER), the value is stored as the data type show in the Table Data Type column.

For more information on default values, see “Default Values for NonStop ODBC Server Attributes” and “ZNSSCFG (For System Configuration Values)” in the *NonStop ODBC Server Reference Manual*.

Table A-1. System Default Values Installed in ZNSSCFG (page 1 of 4)

Column Name or Attribute (Stored in the ITEM Column)	Table Data Type	System Default (Stored in the VALUE Column)
ACC_LEVEL	VARCHAR (60)	SESSION
ACC_LOGTABLE_NAME	VARCHAR (182)	" " (no default is assigned)
ACC_MODE_ON	CHAR (1)	N
AVAILABLE_SERVERS	INT (2) UNSIGNED	0
CACHE_STATISTICS	CHAR (1)	N
CANCEL_TIMEOUT_SEC	INT (1)	0
CHECK_INTERVAL_SECS	CHAR (20)	-1 (indicates no polling takes place regarding the validity of the attribute name in the ITEM column of the ZNSSCFG table)
CON_MODE_ON	CHAR (1)	N
CON_NAME	VARCHAR (60)	" " (no default is assigned)
CON_TEXT	VARCHAR (3900)	None
CPU_BACKUP	INT (2)	-1 (indicates PROCESS_ CREATE will assign the CPU)
CPU_LIST	VARCHAR (37)	-1 (indicates PROCESS_ CREATE will assign the CPU)
CPU_PRIMARY	INT (2)	-1 (indicates PROCESS_ CREATE will assign the CPU)
DATAPAGES	INT (2)	0
DEFAULT_DATABASE	VARCHAR (60)	"MASTER"
DEFAULT_LOCATION	CHAR (16)	" " (no default is assigned)
DEFAULT_SECURITY	CHAR (4)	" " (no default is assigned)

Table A-1. System Default Values Installed in ZNSSCFG (page 2 of 4)

Column Name or Attribute (Stored in the ITEM Column)	Table Data Type	System Default (Stored in the VALUE Column)
DEFAULT_SCHEMA	VARCHAR (60)	" " (no default is assigned)
DEFAULT_VOLUME	CHAR (25)	" " (no default is assigned)
DEFINE_ATTRIBUTE	CHAR (16)	" " (no default is assigned)
DEFINE_CLASS	CHAR (16)	" " (no default is assigned)
DEFINE_NAME	CHAR (24)	" " (no default is assigned)
DEFINE_VALUE	VARCHAR (512)	" " (no default is assigned)
END_TIME	TIMESTAMP	None
EMIT_EVENTS	CHAR (1)	Y
ERR_FILE	CHAR (31)	" " (no default is assigned)
EXT_SWAPFILE	VARCHAR (35)	" " (no default is assigned)
G_USERNAME	CHAR (17)	" " (no default is assigned)
GOV_ACTION	VARCHAR (60)	None
GOV_ATTRIBUTE	VARCHAR (60)	None
GOV_MODE_ON	CHAR (1)	N
GOV_NAME	VARCHAR (60)	" " (no default is assigned)
HOMETERM	CHAR (24)	$OSP
IDLE_DELETE_DELAY_SEC	INT (4)	0
IN_BUFFER_SIZE_B	INT (2)	2448
IN_FILE	CHAR (35)	" " (no default is assigned)
INIT_HEAP_SIZE_KB	INT (2)	750
INPUT_STREAM	CHAR (1)	N
JOB_ID	INT (2)	1
LAST_UPDATE_SYSTEM_CONFIG	CHAR (27)	Timestamp at time of installation
LIMIT_VALUE	INT (8)	None
LOG_TO_HOMETERM	CHAR (1)	N
LOGIN_TIMEOUT_SEC	INT (1)	0
MAX_HEAP_SIZE_KB	INT (2)	750
MAX_SERVERS	INT (2) UNSIGNED	1
MEMORY_CHECK	CHAR (1)	N
MLAN_ADAPTOR	INT (2)	0
MLAN_DOMAIN	VARCHAR (60)	None
MLAN_GATEWAY	VARCHAR (60)	" " (no default is assigned)

Table A-1. System Default Values Installed in ZNSSCFG (page 3 of 4)

Column Name or Attribute (Stored in the ITEM Column)	Table Data Type	System Default (Stored in the VALUE Column)
NET_NAME	VARCHAR (60)	None
NET_PROTOCOL	VARCHAR (60)	None
NET_QUALIFIER	VARCHAR (128)	" " (no default is assigned)
NOS_ALIASNAME	VARCHAR (60)	None
NOS_CREATE_OPTIONS	INT (2)	0
NOS_DEBUG_OPTIONS	INT (2)	0
NOS_LIBRARY_FILE	CHAR (35)	" " (no default is assigned)
NOS_OBJECT	CHAR (34)	$SYSTEM.SYSTEM.NOS
NOS_RUN_OPTIONS	VARCHAR (240)	" " (no default is assigned)
NOS_USERNAME	VARCHAR (60)	None
NOSUTIL_CREATE_OPTIONS	INT (2)	22
NOSUTIL_DEBUG_OPTIONS	INT (2)	0
NOSUTIL_OBJECT	CHAR (34)	$SYSTEM.SYSTEM.NOSUTIL
NOSUTIL_RUN_OPTIONS	VARCHAR (240)	" " (no default is assigned)
NSSQL	CHAR (1)	N
OBJ_CACHE_LEVEL	INT (4)	1
OUT_BUFFER_SIZE_B	INT (2)	2448
OUT_FILE	CHAR (35)	" " (no default is assigned)
OUTPUT_STREAM	CHAR (1)	N
PIPE_TEST	CHAR (1)	N
PRIORITY	INT (2)	-1 (indicates the caller’s priority is used)
PROFILE_NAME	VARCHAR (60)	DEFAULT
QST_MODE_ON	CHAR (1)	N
QST_LOGTABLE_NAME	VARCHAR (182)	" " (no default is assigned)
SCS_CREATE_OPTIONS	INT (2)	0
SCS_DEBUG_OPTIONS	INT (2)	0
SCS_FILENAME	CHAR (16)	None
SCS_LIBRARY_FILE	CHAR (35)	" " (no default is assigned)
SCS_OBJECT	CHAR (35)	$SYSTEM.SYSTEM.SCSOBJ
SCS_PRIORITY	INT (2)	-1 (indicates the caller’s priority is used)
SCS_RUN_OPTIONS	VARCHAR (240)	" " (no default is assigned)
SER_NAME	CHAR (32)	None

Table A-1. System Default Values Installed in ZNSSCFG (page 4 of 4)

Column Name or Attribute (Stored in the ITEM Column)	Table Data Type	System Default (Stored in the VALUE Column)
SO_KEEPALIVE	INT (4)	0
SO_OOBINLINE	INT (4)	0
SO_LINGER	INT (4)	0
SO_REUSEADDR	INT (4)	0
SP_READ	CHAR (1)	N
SP_WRITE	CHAR (1)	N
SQL_ACCESS_MODE	CHAR (2)	RW
SQL_CURSOR_MODE	CHAR (2)	RW (SQL_CURSOR_ DEFAULT affects cursors not specified as FOR READ ONLY or FOR UPDATE. "RW" indicates the cursor defaults to FOR UPDATE)
SQL_DIALECT	CHAR (8)	TDM_CORE
SQL_MAX_STATEMENT_ CACHE	INT (8)	0
SQL_TXN_ISOLATION	CHAR (1)	1 (indicates stable access)
SQL_UNSUPPORTED	CHAR (1)	E (indicates any unsupported syntactical token is reported as an error and the statement fails)
START_TIME	TIMESTAMP	None
STMT_CACHED	CHAR (1)	None
STMT_NAME_CACHE	CHAR (1)	Y
STMT_ORIGIN	CHAR (1)	None
STMT_STATUS	CHAR (1)	None
STMT_TYPE	CHAR (1)	None
SWAPVOL	CHAR (25)	" " (no default is assigned)
TRA_ERROR	CHAR (1)	N
TRA_LEVEL	VARCHAR (60)	" " (no default is assigned)
TRA_MODE_ON	CHAR (1)	N
TRA_NAME	VARCHAR (60)	" " (no default is assigned)

Index