

Database Administration

This chapter contains information that the Database Management System (DBMS) system administrator will use to perform basic database maintenance. The tasks described require a detailed working knowledge of UNIX, network communications, and the Abstract Syntax Notation One (ASN.1) language.

This chapter includes the following sections:

- Logging in to the sa Account
- Administering the Sybase Server including starting and shutting down the server
- Modifying the Database Password
- Monitoring Transaction Log and Disk Space including Purging All Table Data
- Enlarging the Database
- Modifying the Database Transaction Log
- Backing Up the Database
- Recovering a Database
- Moving the Database
- Using `makesample` to Recreate a Table File

All of the tasks in this chapter are performed using root or sa privileges, unless otherwise noted. You can tell when a task requires these privileges by the use of the pound sign (#) in the examples or steps.



Caution Remember to shutdown the Sybase dataserer using the `$NMSROOT/etc/shutdown_nms` script before performing a system shutdown.

Logging in to the sa Account

The system administrator uses two accounts:

- *root*—Use the root account to administer the UNIX system.
- *sa*—Use the sa account to administer the Sybase database.

This chapter describes the tasks to perform while using these accounts. The Sybase sa account is the equivalent of the UNIX root account, because the Sybase database server does not check permissions while you are logged into the sa account.

If you forget your Sybase system administrator password, you cannot get system administrator access to the Sybase database unless you reinstall Sybase. During the reinstall, you will reestablish a password as part of the installation.

Administering the Sybase Server

This section includes information on how to start, query, shut down, and troubleshoot the server.

For information on how to disable the CiscoWorks default of automatically clearing the database, thereby customizing your backup procedures, refer to the section “Disabling the Automatic Clear Function” later in this chapter.

Note The NMSROOT and SYBASE environment variables must be set properly in order for you to perform the database administration tasks.

Performing System Administration

Use the following command to perform system administration tasks such as viewing or adding user accounts to the Sybase database:

```
hostname# user_nms
```

Starting the Sybase Server

The Sybase server must be running in order for your CiscoWorks applications to collect and display data.

To start the Sybase server use the **nmstartup** command. Typically, you run **nmstartup** when you boot your system. The **nmstartup** command also starts other background processes required by CiscoWorks. You set this functionality during the installation and configuration process as described in the *CiscoWorks Administration and Installation Guide*.

After you use the *shutdown_nms* script, perform the following steps to restart the Sybase server:

Step 1 Enter the following command to start all CiscoWorks processes that are not currently active:

```
# $NMSROOT/etc/nmstartup
```

After startup, Sybase reads the *interfaces* file to get TCP port numbers.

Step 2 Enter the **ps -ax** command to list all these processes.

Step 3 Verify that the *\$SYBASE/interfaces* file is similar to the following example:

```
# cat $SYBASE/interfaces
query tcp sun-ether lol-ss2 8100
master tcp sun-ether lol-ss2 8100
console tcp sun-ether lol-ss2 8101
```

In this example, lol-ss2 is the name of the workstation running the database, and 8100 and 8101 are the workstation ports.

Step 4 Enter the following command to access the sa account:

```
# $SYBASE/bin/isql -Usa -P password
1> quit
```

This command confirms that the database is up and can communicate with processes. A message may indicate that the network is unavailable or that Sybase is not currently running.

Querying the Server

The **isalive** command is used to query the server to determine if it is connected and operational. Use it to quickly find out if your database functions are available. The query uses the same arguments as in any SQL server client application. If the query succeeds in connecting to the server, it returns a successful status code (zero) to the command line. If the query fails to make a server connection, a message identifying the reason displays and returns a nonzero status code.

To query the Sybase server, make a server connection query by entering the following command syntax:

```
$NMSROOT/etc/isalive [-U username] [-P password] [-S server_name]
```

-U specifies a username to the server. The default is the environmental variable USER.

-P specifies the password. If you do not enter the password, the system will prompt you for it.

-S is the name of the server. The server names are located in the *interfaces* file in the home directory of the user sybase. (The home directory is defined by the *\$SYBASE* environment variable which is normally set to *\$NMSROOT/sybase*.) The default server name is SYBASE.

Following are examples of the **isalive** command:

```
% isalive -Ulloyd -SSYBASE
% isalive -Psecret
```

If there is a problem with the server, an error message will appear.

Shutting Down the Server

Use the *shutdown_nms* script to shut down the SQL server. The script sequentially shuts down the database to prevent corruption that might otherwise occur if power were simply cut off.



Caution Remember to shutdown the Sybase dataserver using the *\$NMSROOT/etc/shutdown_nms* script before performing a system shutdown.

To use the *shutdown_nms* script, perform the following steps:

Step 1 Enter the following command:

```
# $NMSROOT/etc/shutdown_nms
```

The *shutdown_nms* script is in the *\$NMSROOT/etc* directory.

Step 2 Enter your DBMS system administrator (SA) password if requested.

The following message appears:

```
Server SHUTDOWN by request.
The SQL Server is terminating this process.
```

As the shutdown completes, the following message appears:

```
DB LIBRARY error: Unexpected EOF from SQL Server.
```

For details, see your system administrator for instructions or your *Sun System Administration Guide*.

Server Not Running

If the server does not appear to be running, perform the following steps:

Step 1 Enter the following command to check for the server process:

```
% ps -vax |grep dataserver
```

Step 2 If the server process is not listed, log in as a superuser and enter the following command:

```
# $NMSROOT/etc/nmstartup
```

Step 3 If the server process is not listed, enter the following command:

```
% cat $SYBASE/interfaces
```

The system name where your server is located appears. Ensure that this system is listed in your domain name server (DNS) or other name server.

For detailed information on Sybase error messages, refer to the *Sybase System Administration Guide*.

Modifying the Database Password

As shipped, null is the DBMS SA password to access the database is *null*. Press Return at the password prompt to accept the *null* password. You can change the password initially, or at any time by performing the following steps:

Step 1 To change the password at first usage, enter the following commands to log in as a superuser and change your password:

```
# $SYBASE/bin/isql -Usa -Ppassword
1> sp_password NULL,newpassword
```

Or

To change an established password, enter the following commands:

```
# $SYBASE/bin/isql -Usa -P current_password
1> sp_password current_password, newpassword
```

The following system message appears:

```
The newpassword is the password you want to establish.
```

Step 2 Enter the following command to continue:

```
2> go
```

The following system message appears:

```
Password changed.
return status=0
```

Step 3 Enter the following command to exit the command mode:

```
1> quit
```

Monitoring Transaction Log and Disk Space

By monitoring your disk and transaction log use, you can prevent system problems. If you do not provide enough disk or transaction log space, you will receive error messages. One of the consequences of not having enough database space is that your CiscoWorks processes will halt, and the database log will be unable to accept additional messages and will display error messages.

To calculate the amount of disk space used in your system, you must perform two separate procedures. Using the calculations from these procedures, you can determine the amount of disk space used. Check the size of the database periodically (we suggest weekly checks) to ensure that your system runs smoothly.

Allocating Database Disk Space

When CiscoWorks is shipped, the disk space allocated to database functions leaves about 10 megabytes (MB) of free space to record table information and polling. Polling can consume an enormous amount of space for retention of the data being acquired by the queries. The faster a poll rate (shorter interval), the faster disk space is consumed. For this reason, adjust the disk space allocation before polling with Device Polling. This means that when you create tables, be certain to save them with a *no poll interval* entry, so they do not start polling and fill the database.

Monitoring Transaction Log Utilization

First, you must determine the transaction log space used. Use the following to access data on how much space the transaction log is using and how much is available in the transaction log only.

To monitor your transaction log utilization data, perform the following steps:

Step 1 Log in as the system administrator and access the database by entering the following commands:

```
% $SYBASE/bin/isql -Usa -P password
1> use nms
2> go
1> dbcc checktable(syslogs)
2> go
```

The system displays the percentage of transaction log currently in use similar to the following:

```
Checking syslogs
The total number of data pages in this table is 2.
*** NOTICE: Space used on the log segment is 0.00 Mbytes, 0.10%.
*** NOTICE: Space free on the log segment is 4.00 Mbytes, 99.90%.
Table has 35 data rows.
DBCC execution completed. If DBCC printed error messages, see your
System Administrator.
```

Step 2 Enter the following to exit the command mode:

```
1> quit
```

Verifying Available Database Space

Sybase requires at least 1 MB of free space for reliable operation. When allocating your disk space, make sure you include this free space.

CiscoWorks database files are preallocated, so they will not visibly grow as you fill the database. As data is added, the access times on the files *\$NMSROOT/sybase/data/*.dat* will change.

To determine the amount of available space in the database, perform the following steps:

Step 1 Enter the following commands:

```
# $SYBASE/bin/isql -Usa -P password
1> dbcc checkalloc(nms)
2> go
```

Output similar to the following displays:

```
Checking nms
Database 'nms' is not in single user mode - may find spurious
allocation problems due to transactions in progress.
Alloc page 0 (# of extent=32 used pages=69 ref pages=69)
Alloc page 256 (# of extent=31 used pages=49 ref pages=49)
Alloc page 512 (# of extent=32 used pages=111 ref pages=111)
Alloc page 768 (# of extent=27 used pages=199 ref pages=199)
Alloc page 1536 (# of extent=1 used pages=0 ref pages=0)
Total (# of extent=126 used pages=431 ref pages=784) in this database
DBCC execution completed. If DBCC printed error messages, see
your System Administrator.
```

The line of output containing the word “Total” indicates how many pages are used. In this example, the used pages equal 431. In other words, 431 pages are in use. Each page is 2 KB, so the total usage is 862 KB.

Step 2 To check for allocated data and log space in the *nms* database, enter the following commands:

```
% $SYBASE/bin/isql -User -P
1> sp_helpdb nms
2> go
```

Output similar to the following displays:

device_fragments	size	usage
-----	-----	-----
nms	4.000000 MB	log only
nms	6.000000 MB	data only
nms_data	16.000000 MB	data only

device	segment
-----	-----
nms	default
nms	logsegment
nms	system
nms_data	default
nms_data	system

(return status = 0)

In this example, the size of the total database (nms + nms+ nms_data) is 26 MB. Thus, 862 KB out of 26 MB is in use, indicating approximately 3.2 percent utilization.

Step 3 Enter the following command to quit:

```
1> quit
```

Calculating Disk Space Utilization

To calculate disk space utilization, subtract the transaction log size and the used data space from the total disk space. This is the free memory available. To calculate available transaction log space, refer to the section “Verifying Available Database Space.”

$$\text{Total database size} - \text{Transaction log size} - \text{Used data space} = \text{Disk space available for data}$$

The following example calculates the free memory available based on the previous example.

Take the total size of the database and subtract the transaction log size and the data space used for the amount of free disk space available.

$$26,624 \text{ KB (26 MB)} - 4096 \text{ KB (4 MB)} - 862 \text{ KB} = 21,666 \text{ KB}$$

Using the estimates in the previous examples, the size of the database is 26 MB (or 26,624 KB), minus the transaction log size of 4 MB (or 4096 KB), minus the used data space of 862 KB, which equals available disk space of 21,666 KB or 81.4 percent available.

Purging Data Collected During Device Polling

Data collected during device polling is stored in the polling table. For example, if you have a table containing 20 MIB objects, the data collected from those 20 objects is stored in the polling table. You should periodically review your need to retain much of this data because its value diminishes rather rapidly, while its consumption of space keeps increasing.

This data can be purged without disturbing the configuration table structure, thereby eliminating the need to destroy the complete table and then reconstruct table parameters to reestablish a viable poll base.

There are two methods to purge data: a total purge of the table poll data or a partial erasure. The following procedure uses the SQL **delete** command through the ISQL interface.

Note Ensure that your Sybase environment variable is set properly.

Purging All Table Data

To purge all data in a table, perform the following steps:

Step 1 To select the database table, enter the following commands:

```
# $SYBASE/bin/isql -U user -P
1> truncate table mytable
2> go
```

Mytable is the table name containing the data to be deleted.

Step 2 Enter the following command to exit:

```
1> quit
```

Partially Purging Table Data

To partially purge table data, perform the following steps:

Step 1 Enter the following commands:

```
# $SYBASE/bin/isql -U user -P
1> delete mytable
```

Step 2 Enter the following commands to indicate what data to erase:

```
2> where timestamp < "month day year"
3> go
```

The timestamp date tells the **delete** command to erase all data in the file prior to the date entered.

Step 3 Enter the following command to exit:

```
1> quit
```

Enlarging the Database

There are two ways to increase disk space allocated to the database.

- The first method is a shell script, **enlarge_nms**, that prompts you as you perform the process. This method mandates a minimum additional space increment of 8 MB.
- The second method allows you to enter a series of commands (command sequence method) to increase allocation to a specific size. In both instances, the value used is an increase over the configured base.

You can display the amount of space currently being used by each table using the **dbcc** command in **isql** to verify the tables. For information on how to use the **dbcc** command, refer to the section “Monitoring Transaction Log Utilization,” earlier in this chapter. Do not use the **sp_spaceused** command because it will not provide accurate information.

Enlarging Disk Space Using a Shell Script

The **enlarge_nms** script creates a new file of the size you want to enlarge the database by. The default filename is *nms.dat* and it adds new files by naming them as *nms2.dat*, *nms3.dat*, and so on. The default location for the file is *\$SYBASE/data*, but you can change the location.

To expand database memory allocation using the shell script, perform the following steps:

Step 1 Enter the following command to start the shell script:

```
# $NMSROOT/etc/enlarge_nms
```

The following message displays:

```
Please enter the name of server you wish to modify or press return to modify
the default server.
```

The default server is *Sybase*.

Step 2 Press Return to accept the default server.

Step 3 Enter your DBMS SA password.

Step 4 Enter the size in megabytes by which you want to increase the database memory allocation.

This must be a minimum of 2 MB. 8 is the default value. (If you elect to increment by 8, you do not have to enter 2, simply press Return.)

Step 5 Enter the name of the file to be used by the database, or press Return to accept `../data/nms2.dat`.

Step 6 Enter the device name this file will be known as to the database.

A device can be either a UNIX file (including path) on which the newly enlarged database will store its data, or it can be a raw partition. The preferred approach is a raw partition because this decreases the likelihood of database corruption. The default will build another UNIX file parallel to the current UNIX files holding the master.

If you want to accept a default `nms_2.disk`, press Return; otherwise, enter a different filename and press Return.

A sample script follows:

```
Device to be created is nms2
UNIX file to be created is ../data/nms2.dat
Size of that file will be 8 MB
Are these values correct? [n] (enter Y for yes)
```

Step 7 After you accept the defaults or enter your customized information in response to the questions in the script, confirm that the values are correct by entering **Y**.

The following message appears while the memory allocation is being adjusted:

```
Do you want to use the new space for the transaction log? [N]
```

If you enter **n**, the new space is used for the database. If you enter **Y**, the new space is used for the transaction log.

Step 8 Confirm database enlargement by entering **N**.

The following message appears:

```
Increasing the size of nms database by 8 Megabytes.
Please be patient.
Extending database by 4096 pages on disk nms2.
```

Upon completion of the database enlargement, the system prompt displays.

Enlarging Disk Space Using a Command Sequence

To expand the database space allocation using the command sequence method, perform the following steps:

Step 1 Enter the following commands to enlarge your disk space:

```
% $SYBASE/bin/isql -Usa -Ppassword
1> disk init
2> name = "nms2",
3> physname = "../data/nms2.dat"
```

The name must be a unique name each time the enlarge script is run. Replace `../data/nms2.dat` with any filename.

Step 2 To continue, enter the following command:

```
4> vdevno = n,
```

Enter any unique number greater than one. This number must be incremented by one each time the enlarge script is run.

Step 3 To continue, enter the following command:

```
5> size = n
```

Enter the size of disk space desired in 2-kilobyte (KB) increment blocks. For example, 2 x 4096 = 8 MB.

Step 4 Start the sequence by entering the following command:

```
6> go
```

Step 5 Alter the database by entering the following command:

```
1> alter database nms
```

Substitute your database name for *nms*.

Step 6 Enter the following command:

```
2> on database_name = n
```

Enter size in megabytes. This number cannot exceed the size specified in the enlarge script. For example, *nms2* is a database name and it might be an 8-MB file.

Step 7 Enter the following command to start the sequence:

```
3> go
```

Step 8 Enter the following command to exit:

```
4> quit
```

You can run the **alter database** portion of the enlarge sequence at any time after the database has been initialized. You do not need to initialize the disk again, unless you want more expansion space allocated than you defined the first time. For example, you run **disk init** with an expansion setting of 2048 (4 MB) and later attempt to run alter database with a setting of 5 MB. This would not be accepted because the value used in the **alter database** sequence exceeded the limit established in the **disk init** sequence.

Modifying the Database Transaction Log

The transaction log is automatically cleared periodically by the system. You can turn off the automatic clear function and clear the log manually, as well as increase the transaction log space.

Disabling the Automatic Clear Function

If you do not want the log to be cleared automatically, disable the clear function by entering the following commands:

```
# $SYBASE/bin/isql -Usa -P
1> sp_dboption nms,"trunc.", false
2> go
1> use database_name
2> go
1> checkpoint
2> go
1> quit
```



Caution If you plan to use your transaction log as a backup, be certain you do not run the script *nmlogreset*, or the existing log will be destroyed.

If the automatic clear function is off, you can turn it on by using the same command sequence, substituting the word **true** for the word **false** as the last word in the first line (1>) of the command sequence.

Manually Clearing the Transaction Log

To manually clear the transaction log, enter the following command:

```
$NMSROOT/etc/nmlogreset
```

Enlarging the Transaction Log Space

There are two ways to enlarge the transaction log space on your CiscoWorks workstation:

- **sp_extendsegment**—Enlarges the transaction log that already exists
- **sp_logdevice**—Creates and transfers disk space

Enlarging Using sp_extendsegment

To add space to the transaction log with the **sp_extendsegment** command, enter the following commands:

```
# $SYBASE/bin/isql -Usa -Ppassword
1> use nms
2> go
3> sp_extendsegment logsegment, nmslog
4> go
```

A message appears confirming that the transaction log was enlarged.

Enlarging Using sp_logdevice

To add space to the transaction log with the **sp_logdevice** command, perform the following steps:

Step 1 Enter the following commands to choose a virtual device number:

```
# $SYBASE/bin/isql -Usa -Ppassword
1> sp_helpdevice
2> go
```

This virtual device number (vdevno) will be used in step 2. Output similar to the following displays:

device_name	physical_name	description		
nms..	/data/nms.data	special, physical disk, 8 MB		

status	cntrltype	device_number	low	high
2	0	2	33554432	33558527

In this example, the vdevno is 2.

Step 2 Select a number other than 0, 1, or any other number already used by your database. Record the number.

Step 3 Enter the following commands to create a transaction log file named *nmslog*:

```
1> disk init
2> name = "nmslog",
3> physname = "../data/nmslog.dat",
```

```
4> vdevno = n,  
5> size = 1024  
6> go
```

Substitute the number you recorded in step 1 for the vdevno option. Size equals the transaction log size in 2-kilobyte (KB) blocks. (The recommended size is 2 MB = 1024 2-KB blocks.)

Step 4 To add more database space, enter the following commands:

```
1> alter database nms  
2> on nmslog = 2  
3> go
```

The 2 defines the transaction log size in megabytes. The recommended size is 2 MB.

The following message appears:

```
Extending database by 1024 pages on disk nmslog.
```

Step 5 To take the space added in step 4 and use it as transaction log space, enter the following commands:

```
1> sp_logdevice nms,nmslog  
2> go
```

Step 6 To verify that you have created a separate space for the transaction log, enter the command:

```
1> sp_helpdb nms
```

Output similar to the following appears:

device_fragments	size	usage
nms	4 MB	data only
nmslog	2 MB	log only

Step 7 Enter the following command to exit:

```
2> quit
```

For further details, refer to the *Sybase System Administration Guide*.

Backing Up the Database

There are two database backup procedures. The first is used to back up CiscoWorks software and database information. Use this method when you first receive your software. The second method is used to back up only data in the database on a regular maintenance schedule by using the *backup_nms* script.

For information on how to disable the CiscoWorks default of clearing the database automatically, so you can customize your back up procedures, refer to the section “Disabling the Automatic Clear Function.”

Backing Up the Database and CiscoWorks Software

Follow the backup recommendations and processes described in your network management platform documentation. For detailed information on the UNIX **dump**, **cpio**, and **tar** commands, refer to the appropriate manual pages.



Caution Shut down the server before backing up your database. If you do not shut down the database before backing it up, the integrity of your backup cannot be ensured. Before starting backup procedures, check your transaction log. If it is full, run the `$NMSROOT/etc/nmslogreset` script first to clear the transaction log. This step will ensure that your backup will not be impeded by a transaction log overflow.

Note We recommend that you back up your complete system using **dump** or a similar command at least once a week. Use the Global Command Scheduler application to perform a multilevel dump backup every other day between the weekly full dump process to assure that changes are captured. Perform a least two dump levels at a minimum.

Because network maps can change frequently, and there could be several different maps, save at least one copy of each map in a separate directory for quick access.

Backing Up the Database Only

The `backup_nms` script backs up the contents of the CiscoWorks database. It does not back up the CiscoWorks software.

Use this method to create a backup environment. After the first backup is completed, simply enter the backup script filename and the administrator password.

You must be a superuser to run the `backup_nms` script.



Caution Before starting backup procedures, check your transaction log. If it is full, run the `nmslogreset` program to clear the transaction log. This step ensures that your transaction log will not overflow and impede the backup.

Using backup_nms the First Time

To use the `backup_nms` script to back up the contents of the CiscoWorks database, perform the following steps:

Step 1 Become a superuser by entering the `su` command:

```
% su
```

Step 2 Enter the superuser password.

Step 3 Enter the following to change directories to `$NMSROOT/etc`:

```
# cd $NMSROOT/etc
```

Step 4 Enter the following to start the database backup operation:

```
# backup_nms
```

You will be prompted for your DBMS SA password.

Step 5 Enter your SA password.

If this is the initial backup and no other backup exists, the following message appears:

```
No nms/etc/DBMS_backup.log file, this must be the first backup.
```

Step 6 Enter the UNIX device or file to accept the backup.

Or

Press Return to accept the default of *\$NMSROOT/sybase/data/backup_nms.dat*.

You will be prompted as follows:

```
Is this OK? [n]
```

Step 7 Enter **y** if you are certain of your device selection.

The device is initialized. You will be informed that a disk device has been added.

The backup will start after you answer the following prompt:

```
Are you ready? [n]
```

Step 8 Enter **y** if you are ready to do the backup.

Using backup_nms for Subsequent Backups

After completing your first backup, the only requirement for subsequent backups is to enter the **backup_nms** command and the SA password. Subsequent backups will overwrite the file created in the first backup. If you want to retain the first or a subsequent backup, copy the backup file to another filename for safekeeping. The date of the last backup is in the file *\$NMSROOT/etc/DBMS_backup.log*.

To complete a subsequent backup, perform the following steps:

Step 1 Enter the following command:

```
% backup_nms
```

You will be prompted for your DBMS SA password if you are not a superuser.

Step 2 Enter your DBMS SA password.

The backup process is automatic from this point. The backup is directed to the filename and device used in the first backup sequence.

If the environmental variable *PASSWORD* is set, the script can be run automatically from the Global Command Scheduler. For information on how to use the Global Command Scheduler application, refer to the section “Using the Global Command Scheduler” in Chapter 5.

Recovering a Database

If you have to recreate your database, run the CiscoWorks *recover_nms* script to perform the recovery.

To recover a database, perform the following steps:

Step 1 Become a superuser.

You will be prompted for your DBMS SA password.

Step 2 Enter the following command:

```
# recover_nms
```

You will be prompted with the following message:

```
Nmpolld is running. It must be stopped before performing the recovery. Kill  
it? [yes]
```

Step 3 To accept the default response of yes, press Return.

You will be informed that a recovery is being performed and when that recovery is complete.

Moving the Database

This section describes how to move the database from one server to another. The servers can be in the same physical workstation or in different workstations. This procedure is for installations that have a preexisting Sybase database.

The process of moving a database involves running a script to define a destination server, creating a dump device in the source server and the destination server, creating the database (*nms* directory) on the destination server, dumping the database on the source server, and loading it on to the destination server. To point to the new dataserver, set your DSQUERY to the new dataserver and define it in the *\$SYBASE/interfaces* file.

Note Be aware of different Sybase user IDs when moving databases.

You need to ensure that both server names are located in the *interfaces* file on the source machine. If you intended to use the same server name, for example, the default SYBASE, edit the server name in the *interface* file on the source machine on which you are running the script.

Note This discussion is about Sybase accounts that are internal to the database. Do not confuse them with UNIX user accounts that appear in the */etc/passwd* file.

Sufficient disk space must be available to hold the contents of *nms* (and your data) in a UNIX file (temporarily). The file will be written by Sybase on the source machine and read by Sybase on the destination machine.

The source server name will default to SYBASE in the absence of input.

Note Before moving the database, you must add a Sybase account named *nms* with a password of *beta*. The *nms* account is an alias for the Sybase account named *user*.

To move the database, perform the following steps:

Step 1 Enter the following commands:

```
% cd $NMSROOT/etc
% move_nms
```

A message appears asking you if you want to proceed.

Step 2 Enter **Y** if you want to proceed.

The following message appears:

```
Please enter the name of server FROM which you are moving nms:
```

Step 3 Enter the name of the source server (the server from which the *nms* database is being moved).

The server name should be **SYBASE** unless you have created your system in an environment different from that delivered by the factory.

The following message appears:

```
Please enter password for sybase user 'sa' (on source server):
Enter your DBMS system administrator (sa) password (source
server).
Press Return.
```

Step 4 Enter your SA password.

The following message appears:

```
Please enter the name of server TO which you are moving nms:
```

Step 5 Enter the name of the destination server (the server to which the *nms* database moved).

Step 6 If the SA password is the same on the destination server, the following message appears:

```
Is the password for "sa" on destination the same? [Y]. Press Return if the
answer is yes.
```

If the answer is yes, enter **Y**.

The size of the *nms* database on the source server displays. Output similar to the following message displays:

```
The size of the nms on the source server is:
database_name  database_size  reserved  data  index_size  unused
nms            4MB          3490KB   2422KB  142KB      926KB
```


How large would you like nms to be on your destination server?
Please enter the size of the database nms (in bytes). It must be equal or
larger than the size given above:

Step 7 Enter the size in whole megabytes to be allocated to the database on the destination server.

The value entered must be equal or larger than the size listed in the reserved field.

The following message appears:

Where can the temporary file be created on the source machine?
Please enter a path for the temporary file or press Return to accept usr/tmp:

Step 8 Enter the path to create the temporary file used in this process.

The location used must contain sufficient disk space to hold the *nms* database temporarily. This space must be at least equal to the value listed under the reserved field in the previous example. You must define a path to place the previous temporary file on the destination machine.

After you enter the path, the following message appears:

Where will the temporary file be on the destination machine?

Step 9 Enter the destination machine.

A confirmation summary list displays.

Step 10 If the values are correct, enter **Y**.

or

If you are not satisfied with the list, press Return.

You will return to the beginning of the interactive script to answer the questions again.

If you entered **Y**, a confirmation message displays when the database transfer is complete.

Step 11 Set your DSQUERY environment variable to point to the new dataserver and define it in the *\$SYBASE/interfaces* file.

Using makesample to Recreate a Table File

Several CiscoWorks applications use the example polling tables named *sample**. If you write over the sample tables or accidentally delete them, you can recreate them using the *makesample* script. The *makesample* script recreates the following tables: *sample_mix*, *sample_traffic*, and *sample_error*.

You can choose which variables to poll in the sample tables by selecting the pick menus located in the Device Polling fields. For more information on the sample tables, refer to the section “Creating a New Polling Table” in Chapter 4.

To recreate your sample tables, perform the following steps:

Step 1 Enter the following command:

```
% $NMSROOT/etc/makesample -Unmsuper -Ppassword
```

A message confirming the recreation of the sample tables displays.

Step 2 Verify that the sample tables were copied into Sybase by restarting the Polling Summary application. The sample tables will appear in the Poll Groups scroll window.

