

Chapter8

Database Administration

This chapter contains information your 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.

Included here are instructions for the following tasks:

- Administering the Sybase server
- Starting up and shutting down the database
- Allocating database disk space
- Changing the database password
- Monitoring transaction log use and disk space
- Purging data collected by Device Polling
- Enlarging the database disk space
- Enlarging the database transaction log size
- Backing up the database
- Recovering the database
- Moving the database from one server to another
- Using the makesample program

Administering the Sybase Server

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

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

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

Starting the Sybase Server

The Sybase server is executed using the **nmstartup** command. Typically, **nmstartup** is run when you boot your machine. The **nmstartup** command also starts the other background processes required by CiscoWorks. You set this functionality during the installation and configuration process.

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

Step 1: Enter the script name *nmstartup* at the prompt.

```
hostname% $NMSROOT/etc/nmstartup
```

Step 2: Press Return.

This script starts all daemon processes that are not currently active.

After startup, Sybase reads the interfaces file to get TCP port numbers. Enter the **ps -ax** command to show all these processes. Verify that the *\$SYBASE/interfaces* file is similar to the following example:

```
hostname# $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.

Enter the following command:

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

This confirms that the database is up and can communicate with processes. The 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 operational and connected for further activity. It represents a quick way to 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 calling shell. If the query fails to make a server connection, it will print a message identifying the reason and return a nonzero status code.

To query the Sybase server, perform the following steps:

Step 1: 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 the password is not given, it will prompt.

- **-S** is the name of the server. The server names are found 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.

The following are examples of the **isalive** command:

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

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

Shutting Down the Server

Use the shutdown script to shut down the SQL server. The script conducts a sequential process shutdown to prevent corruption of the database that might otherwise occur if power were simply cut off. The following describes the shutdown script method:

Step 1: Enter the following command at the prompt:

```
hostname% $NMSROOT/etc/shutdown_nms
```

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

Step 2: Press Return.

Step 3: Enter your DBMS system administrator (SA) password. Press Return.

This following message appears:

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

As the shutdown completes, the following message appears:

```
DB Library: Unexpected EOF from SQL Server.
```

See your system administrator for instructions or your *Sun System Administration Guide* for details.

Troubleshooting Sybase Errors

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

If the server does not appear to be running, complete the following procedure:

Step 1: Enter the following command:

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

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

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

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

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

The resulting output from this command should list the machine name of where your server is located. Ensure that this machine name is listed in your domain name server (DNS).

Modifying the Database Password

As shipped, the DBMS system administrator (SA) password to access the database is null. Press Return at the password prompt. You can change the password initially, or at any time.

Modifying Your Password at First Usage

To change the password at first usage, follow this sequence of commands:

Step 1: Enter the following commands at the command line:

```
hostname# $SYBASE/bin/isql -Usa -P
```

Step 2: At the prompt, enter the following:

```
1> sp_password NULL,newpassword
```

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

Step 3: Continue to enter the following command:

```
2> go
```

The following system messages appear:

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

These two lines are the response to the **go** command.

Step 4: Exit the command mode by entering the following command:

```
1> quit
```

Modifying an Established Password

If you want to change an established password, use the current password with the following sequence of commands:

Step 1: Enter the following command at the command line:

```
hostname# $SYBASE/bin/isql -Usa -P current_password
```

Step 2: At the prompt, enter the following:

```
1> sp_password current_password, newpassword
```

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

Step 3: Continue to enter the following commands:

```
2> go
```

Step 4: Exit the command mode by entering the following command:

```
1> quit
```

Monitoring Transaction Log and Disk Space

By monitoring your system's disk and transaction log use, you can prevent 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.

Allocating Database Disk Space

When CiscoWorks is shipped, the disk space allocated to database functions leaves about 6 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 interrogations. The faster a poll rate is (shorter interval) the faster disk space is consumed. For this reason, adjust the disk space allocation before starting to conduct any polling using Device Polling. This does not mean you should not create the tables. However if you do, be certain to save them with a no poll interval entry so they do not start polling, causing a full database.

Monitoring Transaction Log Utilization

First, you need to find the transaction log space used. The following procedure provides data on what space the transaction log is using and what 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 by entering the following:

```
hostname% $SYBASE/bin/isql -Usa -P password
```

The default Sybase password is *sybase*.

Step 2: At the prompt, enter the following:

```
1> use nms  
2> go
```

Step 3: At the prompt, enter the following:

```
1> dbcc checktable(syslogs)
2> go
```

The system displays the percentage of transaction log currently in use.

```
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 4: To exit the command mode, enter the following:

```
1> quit
```

Verifying Available Database Space

Sybase requires at least one Mbyte of free space for its operation. It will become unreliable when it reaches this limit. 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:

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

The system output will be similar to the following:

```
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 da-
tabase
DBCC execution completed. If DBCC printed error messages, see
your System Administrator.
```

The line of output containing the word “Total” tells you how many pages are used (in this example, used pages = 431 in other words; 431 pages are in use). Each page is 2 KB, so the total usage is 862 KB.

Step 2: Enter the following command sequence to check for allocated data and log space in the *nms* database:

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

You should see something similar to the following output:

```
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% utilization.

Step 3: To quit, enter the following command:

```
1> quit
```

Calculating Disk Space Utilization

To calculate disk space utilization, subtract the available transaction log space from the disk space. This is the free memory available.

Disk Space Example

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

Take the totals of the size of the database and subtract the data space used for the amount of free memory available.

Using the estimates provided in the previous examples, the size of the database is 26 MB (or 26,624 KB) minus the total usage of the database of 862 KB equals disk space available of 25,762 KB or 96.8% available.

The resulting number is the used disk space.

Purging Data Collected During Device Polling

Data collected during device polling is contained 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 table. You should periodically review your need to retain much of this data since its value diminishes rather rapidly, but its consumption of space keeps increasing.

This data can be purged without disturbing the configuration table structure. This eliminates 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; one is a total purge of the tables poll data, while the other does a partial erasure. The process is shown below using the SQL **delete** command through the ISQL interface.



Time Saver: You do not have to set the Sybase environment variable each time. It is presented in this format to ensure your Sybase environment is set properly.

To purge all of a table's data, perform the following steps:

Step 1: Enter the following command sequence:

```
hostname# setenv SYBASE $NMSROOT/sybase  
hostname# $SYBASE/bin/isql -U user -P
```

Step 2: Enter the following at the prompt:

```
1> truncate table mytable
```

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

Step 3: Enter the following at the prompt:

```
2> go
```

Step 4: To exit, enter the following:

```
1> quit
```

To partially erase a table's data, perform the following steps:

Step 1: Enter the following command sequence at the prompt:

```
hostname# setenv SYBASE $NMSROOT/sybase  
hostname# $SYBASE/bin/isql -U user -P
```

Step 2: Enter the following at the prompt:

```
1> delete mytable
```

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

Step 3: Enter the following at the prompt:

```
2> where timestamp < "Jul 2 1990"
```

The timestamp date tells the **delete** command to erase all data in the file prior to the date entered; in this example, before Jul 2, 1990.

Step 4: Enter the following at the prompt:

```
3> go
```

Step 5: To exit the command mode, enter the following:

```
1> quit
```

Enlarging the Database

There are two ways to implement a change to the database allocation. The first method is a shell script that prompts you as you perform the process. This method mandates a minimum additional space increment of 2 MB. In the second method, you 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 (always a minimum of 2 MB).

You can view how much space is currently being used by each table using the **dbcc** command in isql to verify the tables. It will also list the pages used. Do not attempt to use the **sp_spaceused** command; it will not provide the accurate information.

Enlarging Disk Space Using a Shell Script

The following procedure describes the sequence for expanding database memory allocation using the shell script method:

Step 1: Enter the routine name *enlarge_nms*. The following message is presented:

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

The default server is *Sybase*. Press Return.

Step 2: Enter your DBMS system administrator (SA) password. Press Return.

Step 3: 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 a 2, simply press Return).

Step 4: Enter the name of the file to be used by the database, or press Return to accept *../data/nms2.dat*. Press Return.

Step 5: 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 master. If you want to accept a default *nms_2.disk*, press Return; otherwise, enter a different filename and press Return.

A sample script is:

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

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

```
Perform database enlarge? [n] (enter Y for Yes)
```

Step 7: Confirm database enlargement by entering **Y**.

The following message appears:

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

Upon completion of the database enlargement, the system prompt is displayed.

Enlarging Disk Space Using a Command Sequence

The following procedure describes the sequence for expanding database space allocation using the command sequence method:

Step 1: Enter the following at the UNIX prompt (%):

```
hostname# $SYBASE/bin/isql -Usa -Ppasswd
```

Step 2: Enter the following command sequence to enlarge your disk space. The name must be a unique name each time the enlarge routine is run.

```
1> disk init  
2> name = "nms2",
```

The name must be a unique name each time the enlarge routine is run.

Step 3: Continue entering the following command sequence. Replace `../data/nms2` with any filename.

```
3> physname = "../data/nms2.dat",
```

Step 4: Continue entering the following command sequence. Enter any unique number that is greater than one. This number must be incremented by one each time the enlarge routine is run.

```
4> vdevno = 3,
```

Step 5: Continue entering the following command sequence:

```
5> size = 4096
```

This is the size of the disk space desired in 2 KB increment blocks, for example, 2 x 4096=8 MB.

Step 6: Start the sequence by entering:

```
6> go
```

Step 7: Now, alter the database by entering the following at the command line. In this example you would create an 8 MB file.

```
1> alter database nms
```

Step 8: Continue entering the following command sequence. Enter size in megabytes. This number cannot exceed the size specified in the `enlarge` routine.

```
2> on nms2 = 8
```

Step 9: Start the sequence by entering the following command:

```
3> go
```

Step 10: Exit the command mode by entering the following command:

```
4> quit
```

You can run the **alter database** portion of the `enlarge` sequence at any time after it has been initialized. You do not need to initialize the disk again, unless you want more expansion space allocated than was 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 since 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, you can disable the clear function by entering the following commands:

```
hostname# setenv SYBASE $NMSROOT/sybase
hostname# $SYBASE/bin/isql -Usa -P
1> sp_dboption nms,"trunc.", false
2> go
1> use nms
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 `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 string at the prompt:

```
hostname% $NMSROOT/etc/nmlogreset
```

Enlarging the Transaction Log Space

Complete the following steps to add space to the transaction log.

Step 1: Enter the following commands to choose a virtual device number. This virtual device number (vdevno) will be used in step 2. The commands are as follows:

```
hostname% setenv SYBASE $NMSROOT/sybase
hostname% $SYBASE/bin/isql -Usa -P
1> sp_helpdevice
2> go
```

You will see a screen with lines similar to the following example:

```
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 the example above, 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 = 3,
5> size = 1024
6> go
```

Vdevno is the number you recorded in step 1. Size equals the transaction log size in 2 kilobyte (KB) blocks. (The recommended size is 2 MB = 1024*2 KB blocks.)

Step 4: Enter the following command sequence to add more database space:

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

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

The message “Extending database by 1024 pages on disk nmslog” appears.

Step 5: Enter the following command sequence to take the space added in step 3 and use it as transaction log space:

```
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
```

A screen similar to the following appears:

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

Step 7: Enter **quit** to exit:

```
1> 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. The second method is used to back up only data in the database on a regular maintenance schedule. The *nms_backup* script is used for this purpose.

Refer to “Disabling the Automatic Clear Function” for information on how to disable the CiscoWorks default of clearing the database automatically so you can customize your backup procedures.

Backing Up the Database and CiscoWorks Software

Follow the backup recommendations and processes described in Chapter 7 of the *Sun System and Network Administration Manual*. For detailed information on the UNIX **dump**, **cpio**, and **tar** commands refer to the appropriate manual pages or the *Sun UNIX Reference Manual*.



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* program first to clear the transaction log. This step will ensure that your backup will not be impeded by an overflow problem with your transaction log.

Note: Cisco Systems recommends your complete system be backed up using **dump** or a similar command at least once a week. Perform a cron invoked, multilevel dump backup every other day between the weekly full dump process to assure that changes are being captured. Two dump levels should be considered minimum.

Because network maps can change frequently and there could be several different maps, save at least one copy of maps 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.

Using this method you can create a backup environment by completing the first backup. After the first backup has been conducted, the script permits the backup to be done by simply entering the backup script filename and the administrator password.

You must be a super user 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.

The following describes the first backup sequence for the database using a shell script method:

Step 1: Become a super user by entering **su** at the command line.

```
hostname% su
```

Step 2: Enter the super user password and press Return.

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

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

Step 4: Enter the following to start the database back up operation:

```
hostname# backup_nms
```

You will be prompted for your DBMS system administrator (SA) password.

Step 5: Enter your SA password. Press Return.

If this is the first backup, the following message will appear:

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

Step 6: Tell the system what device or file is to accept the backup. Enter the UNIX device or file to accept the backup and press Return. Or press Return and accept the default of `$NMSROOT/sybase/data/nms_backup.dat`.

Step 7: You will be prompted with the following:

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

Reply **yes** if you are certain of your device selection. The device is initialized.

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

Step 8: The first backup is going to start. The prompt will be:

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

Reply **yes** if you are ready to do the backup.

After completing the first backup, the only requirement for subsequent backups is to enter the backup command and the SA password:

Step 1: Enter the following command at the prompt:

```
hostname% backup_nms
```

You will be prompted for your DBMS SA password.

Step 2: Enter your DBMS SA password. Press Return.

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 crontab.

Subsequent backups will overwrite the file created in the first backup. If you want to retain the first or a subsequent backup, be certain to copy the backup file to another filename for safekeeping.

You can find the date of the last backup in file `$NMSROOT/etc/DBMS_backup.log`.

Recovering a Database

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

To recover a database, complete the following steps:

Step 1: Change to root user.

Step 2: Enter the following command:

```
hostname# recover_nms
```

You will be prompted for your DBMS SA password.

Step 3: Enter your DBMS SA password. Press Return.

Step 4: You will be prompted with:

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

Step 5: Press Return to accept the default response of yes.

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 machine or in different machines. This is for installations that have a preexisting Sybase database.

The process of moving a database involves a script to define a destination server, create a dump device in the source server and the destination server, create the database (*nms* directory) on the destination server, dump the database on the source server, and load it on the destination server.

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

Both server names must be found in the interfaces file on the source machine. If you were intending to using the same server name, the default SYBASE, for example, you would edit the server name in the interface file on the source machine on which you are running the script.

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

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 command:

```
hostname% cd $NMSROOT/etc
```

Step 2: Enter the following command:

```
hostname% move_nms
```

The following message will appear:

```
This script moves the nms database to another Sybase SQL Server
To run it you must:
save enough room there to hold a backup
Have entries for both servers in the file
nms/sybase/SUNX/interfaces. Do you wish to proceed?
[n]
```

Enter **Y** for yes if you want to proceed. Press Return.

Step 3: At the following message, enter the name of the source server (the server from which the *nms* database is being moved).

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

The reply should be **SYBASE** unless you have created your system in an environment different from that delivered by Cisco Systems.

Step 4: Enter the server name and press Return.

Step 5: 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 6: At the following message, enter the name of the destination server (the server to which the *nms* database moved).

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

Press Return.

Step 7: You will be asked if the SA password is the same on the destination server. If the answer is yes, press Return. If the answer is no, type an **N** and press Return. You will be prompted for a password.

```
Is the password for "sa" on destination the same? [Y]
```

The size of the *nms* database on the source server is displayed. The message will appear as (example data included for reference):

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

Step 8: Press Return.

Step 9: Define the size in a whole megabyte value to be allocated to the database on the destination server. The value entered must not be less than the value (in megabytes) given listed under the reserved field.

Step 10: The following message appears:

```
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 11: Define 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 must be at least equal to the value listed under the reserved column above. You must define a path to place the previous temporary file on the destination machine. The message is:

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

Press Return.

```
Where will the temporary file be on the destination machine?
```

Step 12: Enter the destination and press Return.

Step 13: In summary, this confirmation list will be displayed:

```
Home directory of user sybase is:  
Server to be read is:  
Server to be modified is:  
UNIX file to be created on source machine is:  
UNIX file transferred to destination machine is:  
Size of that file will be less than:  
Are these values correct? [n]
```

Data you have entered will appear following the colons. Enter a **Y** for yes. If you are not satisfied with the list, just press Return.

If you press **Y** and press Return, you will get the message:

```
Please be patient, moving a database takes time.  
Please transfer temporary file from /usr/temp on source machine to  
/usr/temp on destination machine and press Return.  
Copy the temp files from source machine to destination machine.  
Creating and loading 10 Mb database SYBASE on default.  
Please be patient.  
Database transfer complete.
```

The database transfer is complete.

Using Makesample Program to Recreate a Table File

Several CiscoWorks applications use the example polling table named *sample*. If you write over the sample table or accidentally delete it, you can recreate it using the **makesample** program.

The sample table provides you with an example polling table. You can choose which variables to poll for by selecting the pick menus located in the Device Polling fields.

To recreate your *sample* table, perform the following steps:

Step 1: Enter the following command at the prompt:

```
hostname% $NMSROOT/etc/makesample
```

The following output appears on your screen:

```
hostname% $NMSROOT/etc/makesample username
(0 rows affected)
Ignore error message about existing sample_view
Msg 2714, Level 16, State 1:
Procedure 'sample_view', Line 5:
There is already an object named 'sample_view' in the database.
Starting copy...
1 rows copied.
Clock Time (ms.): total = 234    Avg = 234    (4.27 rows per sec.)
Starting copy...
11 rows copied.
Clock Time (ms.): total = 116    Avg = 10    (94.83 rows per sec.)
Ignore error message about duplicate row in attr
Starting copy...
Msg 2601, Level 14, State 3:
Attempt to insert duplicate key row in object 'attr' with unique
index 'index1'
0 rows copied.
Clock Time (ms.): total = 90
The sample table has been added to the database.
```

Step 2: Verify the *sample* table was copied into Sybase by restarting Polling Summary. The *sample* table should appear in the Poll Groups scroll window.

