



Backup and Recovery

This chapter explains how to maintain the Cisco CNS Network Registrar databases.

Backing Up Databases

Because the Network Registrar databases do a variety of memory caching and can be active at any time, you cannot rely on third-party system backups to protect the database. They can cause backup data inconsistency and an unusable replacement database.

For this purpose, Network Registrar provides a shadow backup utility, **mcdshadow**. Once a day, at a configurable time, Network Registrar takes a snapshot of the critical files. This snapshot is guaranteed to be a consistent view of the databases. The **mcdshadow** program backs up the DNS data even when you run the shadow backup on a secondary server. This backup is only a single generation backup.

Syntax and Location

Be sure to understand that the notation “.../data/db” in the following sections refers to directories in the Network Registrar product installation path, depending on the operating system:

- Windows—“.../data/db” means the data directory in the installation path, which by default is **\Program Files\Network Registrar\Local\data\db** or **\Program Files\Network Registrar\Regional\data\db**.
- Solaris and Linux—“.../data/db” means the data directory in the installation path, which by default is **/var/nwreg2/local/data/db** or **/var/nwreg2/regional/data/db**.

Network Registrar database utility programs mentioned in the following sections are located in the “.../bin” directory, which you run as its full path name:

- Windows—“.../bin/program” means the program file in the bin directory in the installation path, which by default is **\Program Files\Network Registrar\Local\bin\program** or **\Program Files\Network Registrar\Regional\bin\program**.
- Solaris and Linux—“.../bin/program” means the program file in the bin directory in the installation path, which by default is **/opt/nwreg2/local/usrbin/program** or **/opt/nwreg2/regional/usrbin/program**.



Note

Use only the approved utilities for each type of database. In Windows, if you want to run the utility from outside the installed path, you must set the CNR_HOME environment variable.

Backup Strategy

The backup strategy involves backing up the following databases using the **mcshadow** utility:

- MCD database—...data/db
- CNRDB databases—...data/dhcp/ndb, ...data/dns/ndb, ...data/dns/zchk, data/mcd/ndb, ...data/cnrsnmp/ndb, ...data/leasehist, ...data/subnetutil, and ...data/replica
- CCM database—data/mcd/ndb and data/ccm/ndb

The most basic component of a backup strategy is the daily shadow backup. When problems occur with the operational database, you might need to try recovering based on the previous day's shadow backup. Therefore, you must recognize and correct any problems that prevent a successful backup.

The most common problem is disk space exhaustion. To get a rough estimate of disk space requirements, take the size of the .../data directory and multiply by 10. System load, such as usage patterns, application mix, and the load on Network Registrar itself, may dictate that a much larger reserve of space be available.

You should regularly archive existing shadow backups (such as to tape, other disks, or other systems) to preserve them for possible future recovery purposes.



Caution

There are different database technologies Network Registrar uses with distinct sets of utility programs, as described in the following sections. Using a utility on the wrong type of database can cause database corruption. Use only the utilities indicated. Also, never use the database utilities on the operational database, only on a copy.

Setting Shadow Backup Times

You can set the time at which an automatic shadow backup should occur in the **cnr.conf** file, which is located in .../conf. Edit the file and change the **cnr.backup-time** variable to the hour and minute of the automatic shadow backup, in 24-hour *HH:MM* format. For example, the following is the default:

```
cnr.backup-time=23:45
```

Performing Manual Backups

You can also initiate a manual shadow backup with the **mcshadow** utility. Enter the **mcshadow** command at the prompt to perform the shadow backup.

Using Third Party Backup Programs with mcshadow

You should avoid scheduling third party backup programs while **mcshadow** is operating, with a margin of an hour in either direction. As described in the [“Setting Shadow Backup Times” section on page 7-2](#), the default shadow backup time is daily at 23:45.



Caution

Configure third party backup programs to skip the Network Registrar operational database directories and files, and to back up only their shadow copies. Otherwise, your server might crash. The operational files are listed in the [“Backup Strategy” section on page 7-2](#). The shadow copies that you can back up are the db.bak, dhcp.bak, dns.bak, mcd.bak, and ccm.bak files in the .../data directory.

Network Registrar also maintains lock files in the `.../tmp` directory on Solaris and Linux, and the `.../temp` directory on Windows. They are recreated during reboot, but are vital while a system is running. Any maintenance process (such as virus scanning and archiving) should exclude this temporary directory, together with the operational database directories and files.

Database Recovery Strategy

The database recovery strategy involves manually restoring up to eight Network Registrar databases using tools specific to each type of database. This is different from the backup procedure in which one mechanism backs up all the server data. The two database types are called MCD and CNRDB. The eight databases are:

- MCD Configuration database—Stores server configuration data
- Central Configuration Management (CCM) database—Stores centrally managed cluster data
- DNS database—Stores data for the DNS server
- DHCP database—Stores data for the DHCP server
- SNMP database—Stores data for the SNMP server
- Lease history database—Stores data for DHCP lease histories
- Subnet utilization database—Stores data for DHCP subnet utilization
- Replica database—Stores data replicated from the local clusters

Depending on what happens during database recovery, you might need to apply one of the following options to all or just one of the databases:

- Restore their integrity—For example, if a primary server has a hard disk failure and the Network Registrar data resides on a secondary disk, you can repair the current configuration and state databases once the system is back on line.
- Restore server data backups—For example, a server may have a hard disk failure and the Network Registrar data resides on a second disk. If you can repair two of the three existing databases, you can restore the third from backup.
- Restore configuration and state database backups—For example, if a system hard disk crashes and the Network Registrar database was on that disk, you can restore the database from a backup copy.

Each of the recovery options follows the same general approach:

1. Stop the Network Registrar server agent.
2. Restore or repair the data.
3. Restart the server agent.
4. Monitor the server for errors.

After you are certain that you executed a successful database recovery, always manually execute the `mcshadow` utility to make a backup of the current configuration and state.

Backing Up and Recovering MCD Data

The operational database that the `mcshadow` utility uses is in `.../data/db` and the shadow copy is in `.../data/db.bak`. The actual filenames are `mcddb.d01`, `mcddb.d02`, `mcddb.d03`, and `mcddb.dbd`.

**Caution**

For the MCD database, use only the **dbcheck** and **keybuild** utilities. Do not use the **cnrdb_archive**, **cnrdb_recover**, or **cnrdb_verify** utility that applies to the CNRDB database. See the [“Troubleshooting Databases” section on page 7-9](#) for details. The **dbcheck** utility encounters errors if you try to check the databases while the servers are running. You must stop the servers before you run the utility.

Checking MCD Database Integrity

You can use the **dbcheck** utility to check the integrity of the MCD database. Stop all Network Registrar servers, then go to the `.../data/db` directory for the MCD database. As a safety check, enter the command `.../bin/dbcheck -a mcddb` (this requires system administrator or root privileges).

Recovering MCD Data from Damaged Databases

Depending on the event that caused the database corruption, you can restore the database to a healthy state using the current data. Here are the steps:

-
- Step 1** Stop the Network Registrar Server Agent.
 - Step 2** Change to the `.../data/db` directory.
 - Step 3** Rebuild the key files by entering the command `.../bin/keybuild mcddb`.
 - Step 4** As a safety check, enter the command `dbcheck mcddb`. This requires system administrator or root privileges.
-

Remember to stop the servers first. If there are any indications that the database recovery was unsuccessful, restore the database from a backup, as described in the [“Recovering MCD Data from Backups”](#) section. Also see the [“MCD Recovery Errors”](#) section on page 7-5.

Recovering MCD Data from Backups

Use the shadow backup to recover MCD data, either because a system crash corrupted the regular working database or the disk on which it resides is corrupted.

-
- Step 1** Stop the Network Registrar Server Agent. Be certain that enough disk space is available for a copy of the database files, plus a 15% safety margin.
 - Step 2** Ensure that the following four files are in `.../data/db.bak`:
 - `mcddb.d01`
 - `mcddb.d02`
 - `mcddb.d03`
 - `mcddb.dbd`
 - Step 3** Copy these files to `.../data/db`. Do not use the **move** command, because you may need these files again. You do not need to delete the current contents of the `.../data/db` directory, as these files are cleaned up automatically.
 - Step 4** Change to the `.../data/db` directory.

- Step 5** To rebuild the key files, enter the command `../../bin/keybuild mcddb`.
- Step 6** As a safety check, enter the command `dbcheck mcddb`. This requires system administrator or root privileges.

**Caution**

The CCM CNRDB database maintains centrally managed configuration data that is synchronized with the server configuration data. After restoring from backup, be sure also to restore the CCM CNRDB database from the same backup to maintain a consistent use of your configuration data.

MCD Recovery Errors

You should not have errors. If you do, ensure that:

- The Network Registrar Server Agent is stopped.
- Your current working directory is `.../data/db` directory.
- The `mcddb.dbd` and `mcdschema.txt` files are present in the `.../data/db` directory. If the files are lost or corrupted, backup files are stored in the `.../data/db.bak` directory.

MCD Data Files

You need the files listed in [Table 7-1](#) for a fully functioning MCD database. As described earlier, only a subset of these files are present in a shadow backup. You need at a minimum the `mcddb.dbd`, `mcddb.d0x`, and `mcdschema.txt` files to rebuild the database from a backup.

Table 7-1 MCD Data Files in `.../data/db` Directory

Data File	Description
<code>mcddb.dbd</code>	Template file that describes the low level data schema for the MCD runtime library. Network Registrar cannot run without this file.
<code>mcddb.k01-k03</code>	Key files that contain the redundant data—Network Registrar does not back up these files because they can be completely rebuilt with the keybuild command.
<code>mcddb.d01-d03</code>	MCD data repository files.
<code>mcConfig.txt</code>	Text file from which Network Registrar configures the initial install time database.
<code>mcdschema.txt</code>	Text file containing a version number of the schema in the <code>mcddb.dbd</code> file. Network Registrar does not try to open the database unless the number in this file matches a version constant in the libraries. Network Registrar cannot run without this file.
<code>vista.taf</code> , <code>tcf</code> , <code>tjf</code>	Working files MCD runtime uses to ensure transactional integrity. When database files are restored from a backup, the server uses or discards these files as needed.

Backing Up CNRDB Data

In the case of the CNRDB databases, the **mcshadow** utility copies the database and all log files to a secondary directory in the directory tree of the installed Network Registrar product. For:

- DHCP—The operational database is in the `.../data/dhcp/ndb` directory, with the log files in the `.../data/dhcp/ndb/logs` directory. The shadow copies are in the `.../data/dhcp.bak/ndb` directory.

- DNS—The operational database is in the `.../data/dns/ndb` directory. The important operational components are the change set database and zone checkpoint files. The change set database is in the `.../data/dns/ndb/dns.ndb` file, with log files in the `.../data/dns/ndb/logs` directory. The zone checkpoint files are in the `.../data/dns/zchk` directory. The shadow copies are in the `.../data/dns.bak` directory.
- SNMP—The operational database and log files are in the `.../data/cnrsnmp/ndb` directory. The shadow copies are in the `.../data/cnrsnmp.bak/ndb` directory.
- CCM—The operational database and log files are in the `.../data/ccm/ndb` directory. The shadow copies are in the `.../data/ccm.bak` directory.
- MCD—The operational database and log files are in the `.../data/mcd/ndb` directory. The shadow copies are in the `.../data/mcd.bak` directory.
- Lease history—The operational database and log files are in the `.../data/leasehist` directory. The shadow copies are in the `.../data/leasehist.bak` directory.
- Subnet utilization—The operational database and log files are in the `.../data/subnetutil` directory. The shadow copies are in the `.../data/subnetutil.bak` directory.
- Replica—The operational database and log files are in the `.../data/replica` directory.

The actual file naming convention is:

- Database—`dhcp.ndb` and `dns.ndb`.
- Log files—`log.0000000001` through `log.9999999999`. Typically only a small number of log files are present. The specific file name extensions at a site vary over time as the database is used. These log files are not humanly readable.

Recovering CNRDB Data from Damaged Databases

Depending on the event that caused the database corruption, you can restore the database to a healthy state by using the current data. This is the best option. Always attempt recovery on a copy of the database file and associated log files, never on the operational files. This is a simple file copy operation, distinct from a shadow backup. Also, never attempt a recovery while Network Registrar is running.



Caution

It is possible to damage the CNRDB database files without the damage being immediately obvious. Such damage could occur because of (a) inappropriately deleting log files; (b) mixing pre- and post-recovery database and log files; (c) attempting recovery of database files currently in use by an application; or (d) using tools intended for other databases, such as MCD database tools. For the CNRDB database, be sure to use only the `cnrdb_archive`, `cnrdb_recover`, and `cnrdb_verify` utilities. Do not use the `keybuild` and `dbcheck` utilities that apply to the MCD database.

Use the `cnrdb_recover` utility, included in the Network Registrar product distribution, for database recovery. Use this tool with care. You should never use it directly on an operational database, or on files another application is concurrently accessing. On a successful database recovery, do not intermingle the recovered files (database file and log files) with files from another source, such as the operational database or shadow backups. Recovered database files acquire state information that make them incompatible with older database files.

- Step 1** Stop the Network Registrar server agent. This stops all the protocol servers. Ensure that enough disk space is available for a copy of the database files, plus a 15% safety margin.

Step 2 Create a temporary directory, `recover`, outside the Network Registrar installation tree, for safety. On Windows, this could be `C:\temp\recover`. On Solaris and Linux, this could be `/tmp/recover`.

Step 3 Copy the following database subdirectories under `.../data` to the `recover` directory:

- DHCP—`.../data/dhcp/`
- DNS—`.../data/dns/`
- SNMP—`.../data/cnrsnmp/`
- CCM—`.../data/ccm/`
- MCD—`.../data/mcd/`
- Lease history—`.../data/listhist/`
- Subnet utilization—`.../data/subnetutil/`
- Replica (optional)—`.../data/replica/`

Double-check that the database file and all log files were copied correctly. Do not allow these files to be modified in any way. Do not run any utilities or servers on these files.

Step 4 Try to recover the current database files:

- a. Change to the appropriate subdirectory of the `recover` directory, then run the recovery utility program, `cnrdb_recover -c -v`. It is helpful to use the `-v` option; otherwise, the utility displays no output in the absence of errors. For details on this utility and its further options, see the [“Using the `cnrdb_recover` Utility” section on page 7-12](#).
- b. Run the verification utility program for each of the servers. There is no output if the verification was successful. For details on the `cnrdb_verify` utility and its options, see the [“Using the `cnrdb_verify` Utility” section on page 7-13](#). For:
 - DHCP—`cnrdb_verify dhcp.ndb` in the `/dhcp` subdirectory of the `recover` directory.
 - DNS—`cnrdb_verify dns.ndb` in `.../recover/dns` subdirectory of the `recover` directory.
 - CCM—`cnrdb_verify` each `*.db` file in the `/ccm` subdirectory of the `recover` directory.
 - MCD—`cnrdb_verify` each `*.db` file in the `/mcd` subdirectory of the `recover` directory.
- c. Optionally, for additional confidence, run the `cnrdb_archive` utility program from the `.../recover` directory:
 - `cnrdb_archive -l`—lists all log files
 - `cnrdb_archive -s`—lists the database file
- d. If there are any indications that an error occurred, proceed to restore the database from a backup, as described in the [“Recovering CNRDB Data from Backups” section](#).
- e. Replace the operational database file and log files in the Network Registrar installation tree with the files in the `recover` directory. Be careful not to mix database files processed with `cnrdb_recover` with those that were not. To avoid this problem, delete the operational CNRDB file and log files from the Network Registrar installation. In copying the recovered files, ensure that the database file and log files are copied to the appropriate and separate directories in the Network Registrar installation tree.

Step 5 Delete the files in the `.../data/dhceventstore` directory.

Step 6 Restart Network Registrar.

Recovering CNRDB Data from Backups

If there are any indications, such as server log messages or missing data, that database recovery was unsuccessful, you may need to base a recovery attempt on the current shadow backup (in the Network Registrar installation tree). Move the operational databases to a separate temporary location, then move the files in the `.../data/name.bak` directory to the `.../data/name` directory; for example, move the contents of `.../data/mcd.bak` to `.../data/mcd`.

The CNRDB database maintains centrally managed configuration data that is synchronized with the server configuration databases. If you restore the CNRDB files from backup, also restore the MCD database from the same backup.



Note

If the recovery fails, perhaps because the current shadow backup is simply a copy of corrupted files, use the most recent previous shadow backup. (This illustrates the need to regularly archive shadow backups.) You cannot add operational log files to older shadow backup files. All data added to the database since the shadow backup was made will be lost.

After a successful database recovery, using the `mcshadow` utility, initiate an immediate shadow backup and archive the files (see the “[Performing Manual Backups](#)” section on page 7-2). This backup is the earliest, oldest backup from which you can do a future recovery. You cannot use older shadow backups.

Using Session Assert Commands for Data Management

You can use the CLI `session assert` commands to simplify interactions with external data management processes. It also helps in writing multicommand batch scripts that stop processing if an asserted precondition fails. You generally use these commands with the default session format of `script`. If the assertion passes, you get a `100 Ok` message. If it fails, you get a `107 Assertion Failed xxx.dbsn (minor-serial-number) = value` message and the CLI exits.

The `session assert locked` command exits the CLI if it cannot lock the session. This sample command file performs batch operations requiring a lock. Note that the session default format is normally set to user format; you set it to script format here:

```
session set default-format=script
session assert locked
commands-that-require-a-lock
```

The `session assert dhcp.dbsn` command exits the CLI session if the minor serial number of the DHCP server does not match (`==`) or does not exceed (`!>=`) the value given. The minor serial number is incremented with each configuration change. Get its value using `dhcp get dbsn`. This sample script modifies a DHCP server based on configuration version 1234:

```
session set default-format=script
dhcp get dbsn
session assert dhcp.dbsn == 1234
scope scope1 create 192.168.1.0 255.255.255.0
scope scope1 addRange 192.168.1.10 192.168.1.200
```

This sample script lists DHCP configuration changes made since version 1234:

```
session set default-format=script
session assert dhcp.dbsn != 1234
scope list
policy list
client-class list
```

Virus Scanning While Running Network Registrar

If you have virus scanning enabled on your system, it is best to configure it to exclude certain Network Registrar directories from being scanned. Including these directories might impede Network Registrar operation. Exclude from scanning the `.../data`, `.../logs`, and `.../temp` directories and their subdirectories.

Troubleshooting Databases

The following sections describe troubleshooting the Network Registrar databases.

Using the `cnr_exim` Data Import and Export Tool

Because Network Registrar extends the data repositories to serve the Web UI, the current Network Registrar data import and export tool, `mcdadmin`, is no longer adequate (see the [“Using the `mcdadmin` Tool” section on page 7-10](#)). The `cnr_exim` data import and export tool now serves to import data to and export data from Network Registrar servers. The `cnr_exim` tool overcomes the `mcdadmin` limitations of not being able to export dynamic resource record information.

Before using the `cnr_exim` tool, exit from the CLI. Then, find the tool on:

- Windows—`.../bin/cnr_exim.exe`
- Solaris and Linux—`.../usrbin/cnr_exim`

You must reload the server for the imported data to become active.

Note that text exports are for reading purposes only. You cannot reimport them.

The data export syntax is:

```
> cnr_exim -e exportfile [-N username -P password -C cluster]
```

(The username and password are prompted for if omitted. The cluster defaults to the local cluster.)

To export raw data, use the `-x` option:

```
> cnr_exim -e exportfile -x
```

To get a complete configuration export you should also export dynamic resource records by including the `-a dynRR` option; to export them alone without any other components, add the `-c "none"` option:

```
> cnr_exim -e exportfile -a dynRR -c "none"
```

To export DNS server and zone components as binary data in raw format, use the `-x` and `-c` options:

```
> cnr_exim -e exportfile -x -c "dnserver,zone"
```

The data import syntax is:

```
> cnr_exim -i importfile [-N username -P password -C cluster]
```

The import file must be in raw format.

You can also overwrite existing data with the `-o` option:

```
> cnr_exim -i importfile -o
```

When you import a configuration including dynamic resource records (either into a new system or overwriting data in the current system), you must import in two steps to get the complete configuration:

1. Do a standard import using **cnr_exim -i**. This imports all but the dynamic resource records. Then, reload the DNS server so that server picks up zone configuration data and allows it to accept dynamic resource records.
2. Import just the dynamic resource records with the command:

```
> cnr_exim -i importfile -a dynRR -c "none"
```

Table 7-2 describes all the qualifying options for the **cnr_exim** tool.

Table 7-2 *cnr_exim Options*

Option	Description
-a dynRR	Imports or exports dynamic resource records only. It assumes that the DNS server was reloaded and zone data exists. (You can combine this option with the -c "none" option to import or export dynamic resource records only.)
-c "components"	Imports or exports Network Registrar components, as a quoted, comma-delimited string. Use -c help to view the supported components. User names are not exported by default; you must explicitly export them using this option, and they are always grouped with their defined groups and roles. Secrets are never exported. Note After you import administrator names, you must set new passwords for them. If you export groups and roles separately from user names (which are not exported by default), their relationship to user names is lost.
-C cluster	Imports from or exports to the specified cluster. Defaults to localhost .
-e exportfile	Exports the configuration to the specified file.
-h	Displays help text for the supported options.
-i importfile	Imports the configuration to the specified file. The import file must be in raw format.
-N username	Imports or exports using the specified username.
-o	When used with the -i (import) option, overwrites existing data.
-P password	Imports or exports using the specified password.
-x	When used with the -e (export) option, exports binary data in raw format.

Using the mcdadmin Tool

The **mcdadmin** tool is a utility with which you can import and export the MCD database and diagnose Network Registrar conditions under Cisco guidance. This tool is relevant for versions of Network Registrar before Release 6.0 only.



Caution

Use this tool only under the guidance of the Cisco Technical Assistance Center. Casual use can inflict catastrophic damage to the Network Registrar configuration database. This tool is only relevant for versions of Network Registrar before Release 6.0. For Network Registrar Release 6.0 data imports and exports, see the “Using the **cnr_exim** Data Import and Export Tool” section on page 7-9.

Before using the **mcdadmin** tool, exit from the CLI. Then, find the tool on:

- Windows—...\bin\mcdadmin.exe
- Solaris and Linux—.../usrbin/mcdadmin

Run the tool from the command shell. This example displays the application version number:

```
> mcdadmin -v
```

This command overwrites the current configuration database with the installation defaults:

```
> mcdadmin -i cnrconfig.txt -z m=3
```

This command exports the scopePC scope configuration data to the cnrconfig.txt file:

```
> mcdadmin -e cnrconfig.txt -p /servers/name/dhcp/1/scopes/scopePC/ -x -z m=3
```

[Table 7-3](#) describes the qualifying options. The absolute path to an object in the MCD tree, *dbpath*, always begins and ends with a forward slash (/).

Table 7-3 *mcdadmin* Options

Option	Description
-a <i>area</i>	Specifies the area of the database to use: config (the default), state , altconfig , or altstate .
-c	Creates the MCD database, and deletes your existing one. Note Use this option only under Cisco Technical Assistance Center guidance.
-d <i>dbname</i>	Specifies the database name or the default database.
-e <i>exportfile</i>	Exports the configuration to the specified file, or a dash (-) for standard output. <pre>> mcdadmin -N admin -P changeme -e mcdconfig.txt > mcdadmin -N admin -P changeme -e -</pre>
-G <i>gen</i>	When used with the -e (export) option, specifies the generation number for the resource record table export.
-H	Dumps the full table generational history.
-i <i>importfile</i>	Imports the configuration from the specified file, or a dash (-) for standard input. <pre>> mcdadmin -N admin -P changeme -o -i mcdconfig.txt -z m=3</pre> Note If used with -o , -i overwrites the current database. If not used with -o , creates new the entries introduced by the file.
-k	When used just by itself, kills the lock manager.
-l	When used with the -i (import) option, hold an exclusive lock while importing.
-o	Overwrites the contents of the current database with entries of the same name.
-O	Merges the data into the existing database, while ignoring unique keys and the resource record tables.
-p <i>dbpath</i>	When used with the -e (export) option, specifies the subset of the configuration database to export, as an absolute path. <pre>> mcdadmin -N admin -P changeme -e scopepc.txt -p /servers/name/dhcp/1/scopes/scopePC/ -x -z m=3</pre>

Table 7-3 *mcdadmin* Options (continued)

Option	Description
<code>-r dbpath</code>	Removes the specified path in the MCD database, not including subentries. <pre>> mcdadmin -r /servers/name/dhcp/1/scopes/scopePC/</pre>
<code>-R dbpath</code>	Recursively removes all entries below and including the specified path. Note Using this option can destroy large portions of the MCD database that may require partial or total reconstruction
<code>dbpath</code>	The use of <code>dbpath</code> in <code>-p</code> , <code>-r</code> , and <code>-R</code> is either an absolute path name starting with <code>/</code> , when it refers to that part of the MCD tree, or a <i>server zone name</i> tuple, starting with a colon (<code>:</code>). The fields after the colon are whitespace-separated and the <i>name</i> , or <i>zone</i> and <i>name</i> , can be left unspecified. For example, <code>-p ":myserv example.com. mypath"</code> .
<code>-t dir</code>	Specifies the directory from which to get the database templates.
<code>-T</code>	Recopies the database templates when creating the database.
<code>-v</code> or <code>-V</code>	Prints version information.
<code>-x</code>	When used with the <code>-e</code> (export) option, exports binary data in raw format. <pre>> mcdadmin -N admin -P changeme -e mcdconfig.txt -x</pre>
<code>-z class=n</code>	Sets the debugging class or classes to level <i>n</i> . <pre>> mcdadmin -N admin -P changeme -c -o -i mcdconfig.txt -z m=3</pre>

Using the `cnrdb_recover` Utility

The `cnrdb_recover` utility is useful in restoring the Network Registrar databases to a consistent state after a system failure. You would typically use the `-c` and `-v` options with this command (Table 7-4 describes all of the qualifying options). The utility is located in the installation bin directory.

Table 7-4 *cnrdb_recover* Options

Option	Description
<code>-c</code>	Performs a catastrophic recovery instead of a normal recovery. It not only examines all the log files present, but also recreates the <code>.ndb</code> (or <code>.db</code>) file in the current or specified directory if the file is missing, or updates it if is present.
<code>-e</code>	Retains the environment after running recovery, rarely used unless there is a <code>db_config</code> file in the home directory.
<code>-h dir</code>	Specifies a home directory for the database environment. By default, the current working directory is used.
<code>-t</code>	Recovers to the time specified rather than to the most current possible date. The time format is <code>[[CC]YY]MMDDhhmm[.ss]</code> (the brackets indicating optional entries, with the omitted year defaulting to the current year).
<code>-v</code>	Runs in verbose mode.
<code>-V</code>	Writes the library version number to the standard output, and exits.

In the case of a catastrophic failure, restore a snapshot of all database files, along with all log files written since the snapshot. If not catastrophic, all you need are the system files at the time of failure. If any log files are missing, **cnrdb_recover -c** identifies the missing ones and fails, in which case you need to restore them and perform the recovery again.

This example shows possible output from this utility:

```
C:\temp\recover\dns> "C:\Program Files\Network Registrar\Local\bin\cnrdb_recover" -c -v
db_recover:Finding last valid log LSN:file:1 offset 83482
db_recover:Recovery starting from [0][0]
db_recover:Recovery complete at Thu Jul 25 21:18:58 2002
db_recover:Maximum transaction ID 80000013 Recovery checkpoint [1][83770]
db_recover:Recovery complete at Thu Jul 25 21:18:58 2002
db_recover:Maximum transaction id 80000000 Recovery checkpoint [1][83770]
```

If you forget to copy the log files into the directory, the output might look like this (note that the maximum transaction ID was not incremented):

```
C:\temp\recover\dns> "C:\Program Files\Network Registrar\Local\bin\cnrdb_recover" -c -v
db_recover:Recovery complete at Thu Jul 25 21:19:20 2002
db_recover:Maximum transaction id 80000000 Recovery checkpoint [0][0]
```

Using the cnrdb_verify Utility

The **cnrdb_verify** utility is useful for verifying the structure of the Network Registrar databases. The command requires a file parameter. Use this utility only if you are certain that there are no programs running that are modifying the file. [Table 7-5](#) describes all its qualifying options. The utility is located in the installation bin directory. The syntax is described in the usage information when you run the command:

```
C:\Program Files\Network Registrar\Local\bin>cnrdb_verify
usage: db_verify [-NoqV] [-h dir] [-P password] file
```

Table 7-5 *cnrdb_verify* Options

Option	Description
-h dir	Specifies a home directory for the database environment. By default, the current working directory is used.
-N	Prevents acquiring shared region locks while running, intended for debugging errors only, and should not be used under any other circumstances.
-o	Ignores database sort or hash ordering and allows cnrdb_verify to be used on nondefault comparison or hashing configurations.
-P password	User password, if the file is protected.
-q	Suppresses printing any error descriptions other than exit success or failure.
-V	Writes the library version number to the standard output, and exits.

Using the cnrdb_checkpoint Utility

The **cnrdb_checkpoint** utility is useful in setting a checkpoint for the database files so as to keep them current. The utility is located in the installation bin directory. The syntax is described in the usage information when you run the command:

```
C:\Program Files\Network Registrar\Local\bin>cnrdb_checkpoint ?
usage: db_checkpoint [-lVv] [-h home] [-k kbytes] [-L file] [-p min]
```

Using the keybuild Tool

The **keybuild** tool is a utility you can use to rebuild the database key files, which contain redundant data with the MCD files:

- On Windows—Click **Start > Settings > Control Panel > Administrative Tools > Services**, highlight Network Registrar Local Server Agent or Network Registrar Regional Server Agent, then click **Stop**. You need system administrator privileges. Go to the *install-location\data\db* folder and run the keybuild.exe file:

```
keybuild mcddb
```

- On Solaris—Stop the Server Agent. Go to the *.../data/db* folder. You need root privileges:

```
/etc/init.d/nwreglocal stop
/etc/init.d/nwregregion stop
keybuild -a mcddb
```

Using the dbcheck Tool

The **dbcheck** tool is a utility you can use to check the MCD integrity. You must have system administrator or root privileges to run the **dbcheck** tool:

- On Windows—Click **Start > Settings > Control Panel > Administrative Tools > Services**, highlight Network Registrar Local Server Agent or Network Registrar Regional Server Agent, click **Start**, then go to the *...data\db* folder and run the dbcheck.exe file:

```
dbcheck mcddb
```

- On Solaris and Linux—Stop the server agent, then go to the *.../data/db* folder and run the dbcheck file:

```
/etc/init.d/nwreglocal stop
/etc/init.d/nwregregion stop
dbcheck -a mcddb
```



Caution

Do not run the **dbcheck** utility while the servers are running.
