



About the `nrcmd` Program

You can use the Web-based user interface (Web UI) or the **`nrcmd`** command line interface (CLI) to configure and manage your DNS, DHCP, and TFTP servers. This chapter describes how to use the **`nrcmd`** command line interface. It specifically describes:

- Invoking the command in batch and interactive modes
- Command organization and syntax
- Special keyboard navigation characters

Invoking the `nrcmd` Command

You can use the **`nrcmd`** command in batch mode by executing scripts that use the commands or by using the interactive mode in which you enter commands at the **`nrcmd`** command prompt. By default, the **`nrcmd`** command is located in `C:\Program Files\Network Registrar\Local\bin` on Windows and in `/opt/nwreg2/local/usrbin` on Solaris and Linux.



Note

In Windows, if you want to run the **`nrcmd`** program from outside the installed path, you must set the `CNR_HOME` environment variable.

On Windows, you can invoke the **`nrcmd`** command window from the Start menu:

Start > Programs > Network Registrar 6.2 > Network Registrar 6.2 CLI

This method prompts for your user name and password. On Solaris and Linux (as well as Windows alternatively), invoke the command from the command prompt using this syntax:

```
nrcmd [general-options] [command] [options]
```

[Table 1-1](#) describes the general options when invoking from the command prompt. [Chapter 2, “Using the `nrcmd` Commands,”](#) describes the commands and their specific options.

Table 1-1 General Options to `nrcmd` Command

Option	Description
<code>-C cluster</code>	Cluster (<i>cluster</i> is the name of the machine on which the Network Registrar servers are running). If not specified, the cluster name defaults to localhost.
<code>-N user</code>	Network Registrar user name (<i>user</i>).
<code>-P password</code>	Network Registrar user password (<i>password</i>).

Table 1-1 General Options to nrcmd Command (continued)

Option	Description
-h	Prints help text.
-L	Accesses the local cluster CLI.
-R	Accesses the regional cluster CLI.
-b <file.txt	Batch file (<i>file.txt</i> is the file of nrcmd commands that run in batch mode, read a line at a time and with a new line printed after the prompt).

**Note**

The cluster to which you connect determines the CLI attributes that appear and are available for the release of Network Registrar running on the cluster. This *CLI Reference* describes the attributes for the current release. For the attributes available for an earlier release, see the *CLI Reference* for that release.

Batch Mode

The program goes into batch mode if you include a functional command or the **-b** <file.txt option on the line. The text file can include any number of **nrcmd** commands, and you can include comment lines preceded by the pound sign (#). In batch mode, you return to the normal system prompt. Note that display in batch mode is intended for parsing by an external program and, therefore, includes only command attributes that have values.

**Note**

The last line of code in the input file must end with an end-of-line character. It is also a good practice to make the last line of code an explicit **exit** command.

Interactive Mode

The program goes into interactive mode if you enter just the **nrcmd** command, or include the *cluster*, *user*, or *password* options. To execute the CLI in interactive mode, enter:

```
nrcmd [-C cluster] [-N user] [-P password]
```

This syntax displays the interactive **nrcmd>** prompt, at which you enter a functional command and any optional parameters:

```
nrcmd> command [parameter,parameter, ...]  
system-response
```

To enter a series of attribute values, insert commas (,) between them. Do not add a space after the comma. If the value is a string containing one or more space characters, enclose the value in quotes:

```
nrcmd> zone example.com. set auth-servers=192.168.50.1,10.0.0.1  
100 Ok  
auth-servers=192.168.50.1,10.0.0.1
```

To terminate an interactive session, enter the **exit** command. To view the online help, enter the **help** command.

Registry and Environment Variables

If you omit the general options, Network Registrar gets them from the Registry or environment variables. If Network Registrar cannot find values for these parameters, it prompts you for them. If you omit the cluster name on a system where Network Registrar servers are installed, the **nrcmd** program assumes access to **localhost** and does not prompt you.

The environment variables that you can set that are recognized by the **nrcmd** program are **CNR_NAME** for the name, **CNR_PASSWORD** for the password, and **CNR_CLUSTER** for the cluster name.

Command Organization

The **nrcmd** commands specify a class of objects, which you can create, delete, or list. Each of these objects in turn has attributes, which you can enable, disable, set, get, and unset, depending on data type. These objects may also have common methods, which are specific to the type of object, and that let you perform operations on groups of attributes.

When you use the **nrcmd** commands to configure Network Registrar, you manipulate:

- **Classes**—Things that you can create, delete, show, or list, such as scopes, policies, or zones.
 - **create**—Creates an entry. If the entry already exists, this command returns an error.
 - **delete**—Removes an entry.
 - **list**—Displays all the objects of a given type, including all attributes.
 - **listnames**—Displays only the names of all objects of a given type.
 - **show**—Displays the values of all the attributes.
- **Attributes**—Things that you can enable or disable, or whose value you can set or display using these common methods:
 - **enable**—Enables a Boolean type of attribute.
 - **disable**—Disables a Boolean type of attribute.
 - **set**—Sets the value of an attribute.
 - **get**—Displays the value of an explicitly defined attribute.
 - **unset**—Makes an attribute have no value. You cannot unset required attributes.



Note

You cannot use **nrcmd** to **get** the value of implicitly defined attributes, including implicitly defined default values.

There are three ways to set attributes:

- For attributes required to create an object, you need to specify their value positionally in the syntax for the **create** command. For example, to create a High-Availability (HA) DNS server pair, you can specify cluster and IP addresses for the main and backup servers during creation:

```
nrcmd> ha-dns-pair ha-pair-1 create 192.168.50.1 192.168.60.1 main=localhost
backup=backup
```

- Use the **set** or **enable** command after creating the object. For example, you can set just the cluster references to the main and backup server for the created HA DNS pair:

```
nrcmd> ha-dns-pair ha-pair-1 set main=localhost backup=backup
```

- Add *attribute=value* pairs at the end of the **create** command.

Note that if you use both the positional value and the *attribute=value* pair for the same attribute on the **create** command line, the *attribute=value* pair is the actual value used (because it comes last).

As of Network Registrar 6.0, attributes that have default values appear in interactive mode in the form *attribute = [default=value]*. In interactive mode, all the attributes appear. In batch mode, only those attributes having values appear, and no default values appear. The display in batch mode is less user-friendly, but is more easily parsable by a program. These examples show how output compares in interactive and batch modes, respectively:

```
nrcmd> zone example.com show
100 Ok
example.com. (primary):
    checkpoint-interval =
    checkpoint-min-interval =
    defttl = 12h
    dynamic = [default=true]
    dynupdate-set =
    expire = 7d
    ...
```

```
$ nrcmd -N admin -P changeme zone example.com show
100 Ok
example.com.: defttl=12h; expire=7d; minttl=10m; nameservers={{0 rr2.example.com.}};
ns=rr2.; origin=example.com.; person=rr1.; refresh=3h; retry=60m; serial=1;
update-acl="key myKey";
```

- Other custom methods—These are specific operations that you can perform on an object, beyond editing its attributes. Examples are adding a range of IP addresses to a scope, or removing hosts from a zone.

Command Usage

How you specify a series of arguments depends on the type of command you are using. The following subsections describe the differences between using the **create**, **set**, and **enable** commands.

Create Keyword

When you use the **create** keyword and there are required arguments, you must supply them. You can also supply additional arguments. You must supply the required arguments in the specified order; however, you can specify the optional arguments in any order with the syntax *attribute=value*.

For example, the syntax for creating a scope is:

```
scope name create ipaddress mask [attribute=value]
```

This means that you must supply an IP address and mask when you create a scope, and you can optionally specify other attributes of the scope.

This example creates the scope testScope with an IP address of 192.168.50.0 and a mask of 255.255.255.0:

```
nrcmd> scope testScope create 192.168.50.0 255.255.255.0
100 Ok
testScope:
    addr = 192.168.50.0
    bootp = disabled
    deactivated =
    ...
```

You can also include attribute definitions on the same line. This example creates the same scope, but also specifies the name of the DNS zone to which a DHCP client's host name should be added:

```
nrcmd> scope testScope create 192.168.50.0 255.255.255.0 dns-zone-name=example.com.
100 Ok
dns-zone-name=example.com.
testScope:
  addr = 192.168.50.0
  bootp = disabled
  deactivated =
  ...
```

After the **create** keyword creates and assigns all specified parameters to the object, it checks that all the required attributes have values (either default or user-specified). If you omit the required attributes, Network Registrar returns an error.

Set Keyword

You use the **set** keyword to set the value of an attribute that is already created. If you want to set a list of values, such as DNS servers or IP addresses, you can separate them with commas. You can also use the **set** keyword to set several attributes on a single line—just specify the attribute and its value followed by a space and the next attribute and value pair.

This example specifies the name of the DNS zone to which a DHCP client's host name should be added:

```
nrcmd> scope testScope set dns-zone-name=example.com.
100 Ok
dns-zone-name=example.com.
```

This example specifies the list of IP addresses for zone transfers for a zone:

```
nrcmd> zone example.com. set auth-servers=192.168.50.1,10.0.0.1
100 Ok
auth-servers=192.168.50.1,10.0.0.1
```

This example sets a client's client-class and domain name:

```
nrcmd> client 00:d0:ba:d3:bd:3b set client-class-name=internal
domain-name=example.com.
100 Ok
client-class-name=internal
domain-name=example.com.
```

The **unset** keyword places an attribute in the undefined state. The **get** keyword displays the value for an attribute.

Enable Keyword

You use the **enable** keyword to enable a Boolean attribute. After you enable one Boolean attribute, you may need to set its associated attributes. Use the **disable** keyword to disable a Boolean attribute. You can use the **unset** keyword to remove the enabled or disabled state of the Boolean attribute.

This example enables incremental transfer processing for the DNS server:

```
nrcmd> dns enable ixfr-enable
100 Ok
ixfr-enable=enabled
```

Once incremental transfer is enabled, this example changes its expiration interval:

```
nrcmd> dns set ixfr-expire-interval=10d
100 Ok
ixfr-expire-interval=1w3d
```



Note

You cannot use **set** and **enable** on the same command line.

Attribute Flags

Command attributes are described as:

- **Required**—The attribute is required for the object, and usually syntactically positional on the **create** command line. You must set the attribute or accept its default, and you can modify the value. You cannot use the **unset** keyword to set a required attribute to undefined. Trying to do so returns the error message 386 - Required attribute cannot be deleted.
- **Optional**—The attribute is optional and does not require a value. You can set and reset the attribute, and you can use the **unset** keyword to make it undefined.
- **Read-only**—The attribute is immutable and read-only. You can use the **get** keyword with the attribute, but you cannot set or unset it. Trying to set or unset a read-only attribute returns the error message 385 - Read-only attribute cannot be modified.

Saving Your Changes

With new commands introduced in Network Registrar 6.2, **nrcmd** applies the changes you make immediately. (Commands introduced in Network Registrar 6.2 are listed in Release Notes). With the commands from prior releases, the CLI waits for one of these events to occur before it saves your changes to the database:

- Invoking the **save** command
- Exiting from **nrcmd**
- Reloading a server
- Adding a resource record or host to a zone

Refreshing and Clearing the CLI Cache

The CLI caches many configuration objects that it reads. If multiple users are making changes simultaneously, one CLI instance might have cached an out of date version of an object. The **session cache refresh** command causes the CLI to clear its local cache of all unmodified objects, forcing it to reread objects from the configuration database. The **session cache clear** command forces the CLI to clear all cached data, whether or not unsaved changes were made.

Navigation Keys

Table 1-2 lists keyboard navigation key combinations that are useful when entering **nrcmd** commands.

Table 1-2 *nrcmd* Navigation Key Combinations

Key Combination	Action
Control-a	Go to the beginning of the line
Control-b	Back one character
Control-d	Delete one character
Control-e	Go to the end of the line
Control-f	Forward one character
Control-k	Kill to the end of the line
Control-l	Redraw the line
Control-n	Next line in the history
Control-p	Previous line in the history
Control-t	Shift an individual character left
Control-u	Delete the line and move the cursor to the beginning of the line
Control-w	Delete one word backwards
Esc-b	Back one word
Esc-f	Forward one word

Command List

Table 1-3 lists the **nrcmd** commands, alphabetically. You can use these commands on the command line or insert them into scripts.

Table 1-3 *nrcmd* Commands

Command	Description
acl	Creates access control lists (ACLs)
addr-trap	Creates traps for free-address monitoring of the DHCP server
address-block	Creates and sets properties for CCM address blocks
admin	Creates administrators for the cluster
ccm	Manages the CCM server
client	Creates clients and assigns them to client-classes
client-class	Creates client-classes
client-class policy	Sets embedded client-class policies
client-policy	Sets embedded client policies
cluster	Configures a regional or local cluster
dhcp	Manages the DHCP server
dhcp-address-block	Creates and sets properties for DHCP address blocks
dhcp-address-block-policy	Configures the policy embedded in a DHCP address block
dhcp-dns-update	Configures the DHCP server updates to DNS

Table 1-3 *nrcmd* Commands (continued)

Command	Description
dhcp-interface	Configures the DHCP server's network interfaces
dhcp-link	Creates DHCPv6 links
dhcp-link-policy	Configures the policy embedded in a DHCPv6 link
dhcp-prefix	Creates DHCPv6 prefixes
dhcp-prefix-policy	Configures the policy embedded in a DHCPv6 prefix
dhcp-subnet	Shows the list of possible DHCP subnets
dns	Specifies the DNS server attributes
dns-interface	Configures the DNS server's network interfaces
dns-update-map	Manages DNS update configurations
exit or quit	Exits the nrcmd command and saves the current configuration changes
export	Exports configurations
extension	Configures DHCP extension modules
failover-pair	Manages DHCP failover server configurations
force-lock	Obtains an exclusive lock for the nrcmd command session
group	Configures groupings of administrators
ha-dns-pair	Defines the high availability relationship between a DNS main and a DNS backup server
help	Provides online help
import	Imports configurations
key	Creates transaction signature (TSIG) keys for DNS updates
ldap	Configures Lightweight Data Access Protocol (LDAP) for DHCP
lease	Retrieves information about DHCP leases
lease6	Retrieves information about DHCPv6 leases
lease-notification	Notifies you when you run out of available leases in a scope
license	Views and updates license information
option	Manages DHCP option definitions
option-set	Manages DHCP option definition sets
owner	Manages owners
policy	Configures DHCP policies
region	Manages regions
remote-dns	Configures remote DNS server data
report	Creates summary usage reports
role	Manages administrator roles
router	Manages routers
router-interface	Manages router network interfaces
router-type	Shows router types

Table 1-3 *nrcmd Commands (continued)*

Command	Description
save	Saves the current configuration changes
scope	Configures DHCP scopes
scope-policy	Configures the policy embedded in a DHCP scope
scope-selection-tag	Creates DHCP scope selection tags
scope-template	Configures templates for DHCP scopes
scope-template-policy	Configures the policy embedded in a DHCP scope template
server	Controls servers
session	Sets the current CLI session parameters
snmp	Configures the SNMP server
snmp-interface	Configures the SNMP server's network interfaces
subnet	Configures CCM subnets
tftp	Configures the Trivial File Transport Protocol (TFTP) server
trap-recipient	Configures recipients for SNMP traps
update-policy	Configures DNS update policies
vendor-option	Configures vendor-specific DHCP options
vpn	Configures DHCP virtual private networks (VPNs)
zone	Configures DNS zones
zone-dist	Manages zone distribution maps
zone-template	Configures templates for DNS zones

