
Release Notes
NetCCN™ for cPCI – MTN Version
Firmware Revision v1.9.8
Release Date: 2005 June 29
 PN# CC04804-00

Summary of Changes 1.9.8r00

Defect/Change Request	Title	Change Summary
D: ccpu00046222	Add API to get NetCCN up time.	<ul style="list-style-type: none"> Add CCNlib API commands to control uptime blurring and access uptime.
C: ccpu00057040	Terminate connect when Telnet disconnects.	<ul style="list-style-type: none"> Remove limitation on escape char. Set default back to ^]

Command Update

Connect Escape Char and Telnet

The previously cited (1.9.6r00) limitations on the custom connect escape character have been removed. The default escape character is ^].

ccnxGetScUpBlurt CCNlib API

This new CCNlib API command gets the current Uptime Blurt interval (similar to the CLI `uptime` command). It returns the current number of seconds between blurts, or zero is not blurring.

ccnxSetScUpBlurt CCNlib API

This new CCNlib API command sets the Uptime Blurt interval to the number of seconds supplied by the user (similar to the CLI `uptime #sec` command). A value of zero disables blurring.

ccnxGetMiscioUptimeStat CCNlib API

This new CCNlib API command get the currently cached uptime value. The cached value is set by the last uptime blurt. The value represents the number of seconds that the NetCCN has been up (similar to the CLI `uptime` command). A value of zero indicates that uptime blurring is disabled, and that the uptime value is not valid.

Summary of Changes 1.9.7r00

Defect/Change Request	Title	Change Summary
D: ccpu00057302	Make escape command rsh accessible.	<ul style="list-style-type: none">• Move escape command from CLI to node.
D: ccpu00055815	CCN agent does not detect temp changes	<ul style="list-style-type: none">• Fix temp monitoring

Command Update

escape Command

The `escape` command has been added to the hidden Node commands. This command lets you configure the escape character used by the CLI `connect` command. You may access this command from another node via `rsh XX escape`.

Summary of Changes 1.9.6r00

Defect/Change Request	Title	Change Summary
C: ccpu00057006	Connect responses only on first connection.	<ul style="list-style-type: none"> All connects to a given NetCCN serial port will receive the response data.
C: ccpu00057040	Terminate connect when Telnet disconnects.	<ul style="list-style-type: none"> When a telnet session to a NetCCN terminates: <ul style="list-style-type: none"> o Terminate any connect sessions o Logout of CLI

Command Update

The custom `connect` escape character set by the `escape` command should be one of the following, to ensure correct interworking when accessing a NetCCN via telnet:

`^a, ^b, ^c, ^d, ^e, ^f, ^g, ^h, ^i`

Failure to do so will result in serial connections not being terminated when the telnet session is disconnected.

The default escape character for the `connect` command is now `^a`. This will only affect pre-1.9.5 versions, updated to 1.9.6. 1.9.5 versions will have already set the escape character to `^]`, unless the user has changed it since installation.

If you only access the NetCCN CLI from a console connection (i.e not a telnet session), then any control character is allowable for the custom escape character. It is still recommended you choose one from the above list.

Summary of Changes from 1.9.5r00

Defect/Change Request	Title	Change Summary
D: ccpu00046219	Add Watchdog Timer	<ul style="list-style-type: none"> • Enable Pentium watchdog timer. • Add code to reset the timer. • Centralize code.
D: ccpu00046952	Connection Timeout	<ul style="list-style-type: none"> • Monitor com connections. • Timeout if no activity • Only enabled if CLI password is enabled.
D: ccpu00054407	Password protect the sh command.	<ul style="list-style-type: none"> • Enable/Disable via shpassword CLI command. • Password same as CLI password. • sh CLI command asks for password. • syscmd CLI command asks for password. • ctrl-C<ret> will abort password query. • 5 unsuccessful retries will abort password query.
D: ccpu00054408	Close the 8800 Backdoor	<ul style="list-style-type: none"> • Enable/Disable via backdoor CLI command. • Removes telnet from inittab.
D: ccpu00049030	Nice -22 error	<ul style="list-style-type: none"> • Correct evaluations of nice command return. • Add ccnd, ccnc syslog started messages.
D: ccpu00054254 C: ccpu00054133 C: ccpu00054433	User data < 8 chars not written	<ul style="list-style-type: none"> • Fix to allow less than 8 char user data.
D: ccpu00054291	User-specified escape char for CLI 'connect'	<ul style="list-style-type: none"> • Add escape command to allow user to define escape character for connect sessions.
C: ccpu00052642	CAN image put/get size limit	<ul style="list-style-type: none"> • Catch SIGPIPE
D: ccpu00055815	CCN agent does not detect temp changes	<ul style="list-style-type: none"> • Fix temp blurring
C: ccpu00053957	EEPROM locking	<ul style="list-style-type: none"> • Revise kernel side to no locking • Correct physical storage of data • Route errors to stderr • Streamline usage
C: ccpu00049662	CLI commands w/o usage info	<ul style="list-style-type: none"> • add usage info for: on, off, forceon, forceoff.
C: ccpu00054380	DHCP Disable	<ul style="list-style-type: none"> • No changes, add note to User's Guide.
C: ccpu00050740	Default Gateway Dropped	<ul style="list-style-type: none"> • Reset EEPROM static gw route for each interface on enable. • Reset EEPROM default gw on enable or disable.
C: ccpu00054053	DHCP On Startup	<ul style="list-style-type: none"> • Fix rc.network script to not reset IP address if DHCP enabled for the interface.
C: ccpu00053197	Cannot enable DHCP without reset	<ul style="list-style-type: none"> • No changes, add note to User's Guide.
D: ccpu00049894 C: ccpu00049855	app watchdog timeout message	<ul style="list-style-type: none"> • Remove unneeded confusing output.
C: ccpu00055991	Add route command	<ul style="list-style-type: none"> • Add node command: route
C: ccpu00054536	CCNlib example makefile	<ul style="list-style-type: none"> • Improved the examples directory.

Defect/Change Request	Title	Change Summary
C: ccpu00054519	Move VOB and fix Builds	<ul style="list-style-type: none"> • Add CCPU specific version to linux kernel. • Place modules in correct directory. • Fix file permissions on root file systems. • Build common root file system. • Fix busybox lsmod command. • Update build-app to use new kernel modules. • Update build-app to use correct kernel includes.
D: ccpu00051170	C++ Comments in ccnlib headers	<ul style="list-style-type: none"> • Change to C comments
C: ccpu00054736	Update Copyright Notices	<ul style="list-style-type: none"> • Change to packaging files

New Command Summary

The following commands have been added to the NetCCN CLI in this release:

```
backdoor
shpasswd
escape
route
wdogs hw
```

escape Command

The `escape` command has been added to the hidden CLI commands. This command lets you configure the escape character used by the CLI `connect` command. The command with no arguments causes the display of the current escape character.

```
ccncli> escape
current escape character is ^b.
```

An argument of `^<char>` redefines the escape character. (Note that the `^` is the caret key (above the 6) not the `<ctrl>` key.)

```
ccncli> escape ^a
current escape character is ^a.
```

An argument of `'?'` will cause the display of the usage of the command.

```
ccncli> escape ?
Usage:
  escape [^<char>]
Shows / sets escape char for connect. Setting is persistent.
Please note <^> is caret key (above <6>) not <Ctrl> key.
```

This escape character is used to escape from a CLI `connect` session.

```
ccncli> connect .2
Connecting to console on port 2 of node 0e. Escape char is ^a
```

```
<Ctrl>a
Connected to console on port 2 of node 0e.
connect> .
```

```
Disconnecting from console on port 2 of node 0e.
ccncli>
```

Note that it is the responsibility of the user to choose a valid control character. Choosing an already defined control character (e.g. `^c`) may cause problems. Also note that `^-` does not work.

backdoor Command

The `backdoor` command has been added to the debug CLI commands. This command lets you enable or disable the ability to telnet to the NetCCN shell via port 8800. The command with no arguments causes the display of the current status.

```
ccncli> backdoor
backdoor is currently ENABLED.
```

An argument of `ena` enables the backdoor, and `dis` disables the backdoor.

```
ccncli> backdoor dis
backdoor is currently DISABLED.
ccncli> backdoor ena
backdoor is currently ENABLED.
```

An argument of '?' will cause the display of the usage of the command.

```
ccncli> backdoor ?
Usage:
  backdoor [ena|dis]
Enables or disables backdoor telnet port. Setting is persistent.
```

The backdoor is access to the NetCCN shell (or 'sh') by telnet'ing to port 8800. When the backdoor is disabled, no telnet access is allowed.

```
destroyer-nccn# telnet 172.17.5.7 8800
Trying 172.17.5.7...
telnet: Unable to connect to remote host: Connection refused
destroyer-nccn#
```

When the backdoor is enabled, telnet access is allowed for user root with no password.

```
destroyer-nccn# telnet 172.17.5.7 8800
Trying 172.17.5.7...
Connected to 172.17.5.7.
Escape character is '^]'.

Linux 2.2.25-nccn-cpci-1.6.1r00 (netccn) (ttyr0)

netccn login: root
# exit
Connection closed by foreign host.
destroyer-nccn#
```

shpasswd Command

The `shpasswd` command has been added to the debug CLI commands. This command lets you enable or disable password protection for shell commands. Shell commands can be executed via the CLI `syscmd` command, or by descending to the linux shell via the CLI `sh` command. The command with no arguments causes the display of the current status.

```
ccncli> shpasswd
Shell password protection is currently DISABLED.
```

An argument of `ena` enables the shell password protection, and `dis` disables the shell password protection.

```
ccncli> shpasswd ena
Shell password protection is currently ENABLED.
ccncli> shpasswd dis
Shell password protection is currently DISABLED.
```

An argument of '?' will cause the display of the usage of the command.

```
ccncli> shpasswd ?
Usage:
  shpasswd [ena|dis]
Enables or disables shell password protection. Setting is persistent.
```

This setting must also be used in conjunction with the CLI `passwd` command. If the current `ccncli` password is 'none', then even with the shell password protection enabled, no password is required.

```
ccncli> passwd
Password is now: 'none'

ccncli> shpasswd
Shell password protection is currently ENABLED.
ccncli> sh
Type 'exit' to return to CLI.
# exit
ccncli>
```

When the `ccncli` password is not 'none', then a password is required.

```
ccncli> passwd letmein
Password is now: 'letmein'

ccncli> shpasswd
Shell password protection is currently ENABLED.
ccncli> sh
ccncli password: letmein
Type 'exit' to return to CLI.
# exit
ccncli>
```

The shell protection allows 5 tries then aborts back to the CLI.

```
ccncli> sh
ccncli password: invalid
Incorrect password. Try again.

ccncli password: another
Incorrect password. Try again.

ccncli password: tryagain
Incorrect password. Try again.

ccncli password: idontknow
Incorrect password. Try again.

ccncli password: helpme
Incorrect password. Try again.

ccncli>
```

Also, the password prompt may be aborted by entering `<Ctrl>c<enter>`.

```
ccncli> sh
ccncli password: wrong
Incorrect password. Try again.

ccncli password: <Ctrl>c<enter>
Giving Up.
ccncli>
```

The shell protection also applies to the CLI `syscmd`.

```
ccncli> syscmd eeprom
ccncli password: letmein
logena=85
```

```

gateway=(172.17.0.254)
bootprotol=5
bootproto0=0
serial={CM3-01982}
ipaddr0=(172.17.5.7)
netmask0=(255.255.0.0)
loghost=(172.17.22.1)
mac0=[0002bb8900a5]
mac1=[0002bb8900a4]
ccncli>

```

route Command

The `route` command has been added to the debug CLI commands. Similar to the `ifconfig` command, this command is a pass-through to the Linux shell `route` command. This should only be used for debug purposes. All network configuration should be handled via the `netcfg` command.

```

ccncli> route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
default          *              255.255.255.255 UH    0      0      0 eth1
172.17.22.1     *              255.255.255.255 UH    0      0      0 eth0
172.17.22.10    *              255.255.255.255 UH    0      0      0 eth0
172.17.66.10    *              255.255.255.255 UH    0      0      0 eth1
172.17.0.0      *              255.255.0.0    U     0      0      0 eth0
172.17.0.0      *              255.255.0.0    U     0      0      0 eth1
127.0.0.0       *              255.0.0.0      U     0      0      0 lo

```

wdogs hw Command

See watchdog changes.

Watchdog Changes

The watchdog has been updated for the pentium processor. To enable the watchdog timer, the following steps must be followed (in order):

1. Load the NCCNpci-v1_9_5r00 image. The image must be loaded first, before enabling the hardware watchdog. Otherwise, there is never enough time to load the image!
2. Place the Pentium Jumptec module in the JTAG harness.
3. During boot press F2 to start the bios setup.
4. Go to Advance
5. Select Device I/O Configurations
6. Go down and Select watchdog
7. Set Mode to RESET, Loop 32s, Delay 32s
8. exit and save

Diag image

When a diag image is started, the watchdog must be immediately disabled before doing anything else, otherwise the system will reboot.

Main Menu items:

```

H/W Watchdog enable
H/W Watchdog disable

```

ccnboot

If any processing other than NetCCN image startup is desired at the `ccnboot` prompt, the watchdog must be disabled immediately, otherwise the system will reboot. The command with no arguments causes the display of the current status.

```

ccnboot> wdogs

```

HW Watchdog timer status is currently DISABLED.

An argument of '?' will cause the display of the usage of the command.

```
ccnboot> wdogs ?
wdogs [ena|dis|kick] enable, disable or kick hw watchdog timer
```

An argument of `ena` enables the watchdog, `dis` disables the watchdog, and `kick` resets the watchdog timer.

```
ccnboot> wdogs ena
HW Watchdog timer status is currently ENABLED.
ccnboot> wdogs kick
HW Watchdog timer status is currently ENABLED.
ccnboot> wdogs dis
HW Watchdog timer status is currently DISABLED.
ccnboot>
```

NetCCN Image

The CLI `wdogs hw` command has been added as a hidden command. The command with no arguments causes the display of the current status.

```
ccncli> wdogs hw
      APP SHUT   OK   ON  OFF USER
Watchdogs:    9   0   0   0   0   0
HWdog: DISABLED.
margins min 32 max 0 last 0 in secs.
# of kicks 0 loop delay 20 timeout 32 in secs
APP watchdog timed out 2057 seconds ago.
```

The statistics display the number of kicks, last, min and max margins. Margin is the number of seconds remaining for the watchdog to reset the system (e.g. if loop time is 20 seconds and timeout is 32 seconds, margin normally should be $(32-20) = 12$ seconds. This could be little less if system happens to be too busy to kick the watchdog.). The APP watchdog is the last time communication was lost with a CCNd process (This has nothing to do with the watchdog).

An argument of '?' will cause the display of the usage of the command.

```
ccncli> wdogs hw ?
Hardware Watchdog Usage:
  wdogs hw [ena|dis|stop|loop <seconds> | timeout <seconds>]
ena:  enable hardware watchdog
dis:  disable hardware watchdog (will cause a watchdog timeout)
stop: stop hardware watchdog (will not timeout)
loop <seconds>: Delay to set thread loop time.
timeout <seconds>: Delay to set timeout value.
```

An argument of `ena` enables the watchdog, `dis` disables the watchdog.

```
ccncli> wdogs hw ena
      APP SHUT   OK   ON  OFF USER
Watchdogs:    9   0   0   0   0   0
HWdog: enabled.
margins min 32 max 0 last 0 in secs.
# of kicks 0 loop delay 20 timeout 32 in secs
APP watchdog timed out 3428 seconds ago.

ccncli> wdogs hw dis
      APP SHUT   OK   ON  OFF USER
Watchdogs:    9   0   0   0   0   0
```

```
HWdog: DISABLED.  
margins min 13 max 13 last 13 in secs.  
# of kicks 1 loop delay 20 timeout 32 in secs  
APP watchdog timed out 3456 seconds ago.
```

The argument `stop` is like `dis`. The argument `loop` determines how often to kick the timer. The argument `timeout` is the time, in seconds, within which a kick should have occurred. Normally these settings should not be changed. They are included for testing purposes only.

Notes

If DHCP is enabled, the watchdog will reset the system if the DHCP server is down (or not available).

Login Timeout

Connection timeouts are enabled whenever a password is defined via the CLI `passwd` command. This timeout occurs when there is approximately 5 minutes of no CLI activity.

Other Output Changes

ccninit

The CLI `ccninit` command now shows the cause of last reset in text. It is logged until a power cycle or `ccninit clear` command is issued. This is very useful for debugging.

```
ccncli> ccninit  
NetCCN was RESET 176896 seconds ago.  
Reset status register = 0x01. Reason POWERGOOD DETECT  
  
ccncli> ccninit clear  
NetCCN was RESET 177001 seconds ago.  
Reset status register = 0x01. Reason POWERGOOD DETECT  
  
ccncli> ccninit  
NetCCN was RESET 177011 seconds ago.  
Reset status register = 0x00. Reason
```

CCNd and NCd Syslog Startup Messages

The CCNd and NCd syslog startup messages have been changed. The following will appear in the system logs if `daemon.debug` is enabled:

```
Mar 16 09:57:57 destroyer-nccn ccnd[462]: [ID 306928 daemon.notice] CCNd starting  
solaris2.7 target, rev 2.1.3r00, compiled Mon Feb 28 10:55:48 PST 2005)  
Mar 16 09:57:58 destroyer-nccn ccnc[463]: [ID 514085 daemon.notice] NCd starting  
(solaris2.7 target, rev 2.1.3r00, compiled Mon Feb 28 10:56:04 PST 2005)  
Mar 16 09:57:58 destroyer-nccn ccnc[463]: [ID 736025 daemon.notice] NCD OUT OF SYNC WITH  
LOCAL CCN, NODE STATE INVALID  
Mar 16 09:57:59 destroyer-nccn ccnc[463]: [ID 789315 daemon.notice] NCD IN SYNC WITH  
LOCAL CCN  
Mar 16 09:57:59 destroyer-nccn ccnc[463]: [ID 653732 daemon.notice] CHG NODESTATE = 0xb  
Mar 16 09:57:59 destroyer-nccn ccnc[463]: [ID 892206 daemon.notice] NCD NORMAL  
Mar 16 09:57:59 destroyer-nccn ccnc[463]: [ID 903687 daemon.notice] NODESTATE READ  
Mar 16 09:57:59 [172.17.22.10.4.0] NCCN[112]: ct_nodest_trans: node state changed from  
0x0b to 0x09  
Mar 16 09:57:59 [172.17.22.10.4.0] NCCN[282]: generate_alarms: FAULTNOW changed from  
0x20820080 to 0x820080  
Mar 16 09:57:59 destroyer-nccn ccnc[463]: [ID 653691 daemon.notice] CHG NODESTATE = 0x9  
Mar 16 09:57:59 destroyer-nccn ccnc[463]: [ID 903687 daemon.notice] NODESTATE READ
```

OS Version

The Linux OS version name has changed. This affects `uname` and `telnet` output:

```
ccncli> sh
Type 'exit' to return to CLI.
# uname -a
Linux netccn 2.2.25-nccn-cpci-1.6.1r00 #1 Wed Feb 9 13:22:29 PST 2005 i486 unknown

destroyer-nccn# telnet 172.17.5.7 8800
Trying 172.17.5.7...
Connected to 172.17.5.7.
Escape character is '^]'.

Linux 2.2.25-nccn-cpci-1.6.1r00 (netccn) (ttyr0)
```

Known Issues

- A modem connected to the front console port **MUST** be pre-configured for the correct baud rate of that port before connecting it. Also, local command echo should be disabled on the modem. This configuration is not recommended. There is a port on the transition board that can specifically be configured for use with a modem (see NetCCN User's Manual).
- Certain serial terminals (VT510/520) operating with software flow control (XON/XOFF) may cause the NetCCN to hang at the boot prompt during startup. This occurs because these terminals send a <Ctrl>q upon detection of a connection. Recommend using no flow control if possible.
- The previously released image cannot transfer the new image over CAN bus using the 'image get' and 'image put' commands. This behavior has been corrected in this release and will cause no issues in future releases.
- The 486 (not pentium) hardware watchdog can infrequently cause the NetCCN to unexpectedly reset when it is initializing.
- There is a known issue with Sun CP2140 not asserting its board healthy signal per the PICMG 2.1 hotswap specification when it is initially powering up. Specifically, the CP2140 will not assert its board healthy signal until after it is taken out of reset and has performed its POST. The NetCCN uses a hotswap state machine as defined in the PICMG 2.1 specification which requires the reset signal to remain asserted by the NetCCN after the assertion of the board select signal until the healthy signal is asserted by the CPCI board. Since the CP2140 does not conform to this requirement, the NetCCN keeps it powered off and in reset. There are several workarounds to this issue. One workaround is to set the healthy mask ('hlthymask' CLI command) for the CP2140's slot. This will allow the NetCCN to automatically power the board up. Another workaround is to disable the automatic hotswapping feature altogether on the NetCCN (using the 'hotswap disable' CLI command) and manually turn the CP2140 on and off as desired.