

Codemat Sarl  
Innere Margarethenstr. 10  
4051 Basel  
Switzerland  
Tel: +41 (0)61 271 64 14  
Web: <http://www.netcube.ch>  
E-Mail: [info@netcube.ch](mailto:info@netcube.ch)

---



# **NetCube Updater**

## **User's Manual**

# Table of Contents

<b>1 NetCube Updater.....</b>	<b>1</b>
1.1 Availability and license protection.....	1
1.2 Overview and concepts.....	1
1.3 Discovering the user interface.....	2
1.3.1 Menus.....	3
1.3.2 Windows descriptions.....	4
1.4 Technical details.....	6
1.4.1 Authentication.....	6
1.4.2 Automatic discovery of NetCubes.....	6
1.4.3 Unconfigured NetCubes.....	7
1.5 Format of the scenario files.....	7
1.5.1 Descriptions of variables.....	7
1.5.2 Telnet commands.....	8
1.5.3 FTP commands.....	9
1.5.4 Miscellaneous commands.....	10
1.6 Important remarks about scenarios.....	10
1.6.1 Known problems.....	10
1.7 Examples of scenarios.....	11
1.7.1 Changing the root password.....	11
1.7.2 Installing a package.....	11
1.7.3 Checking the version of a software.....	11

# 1 NetCube Updater

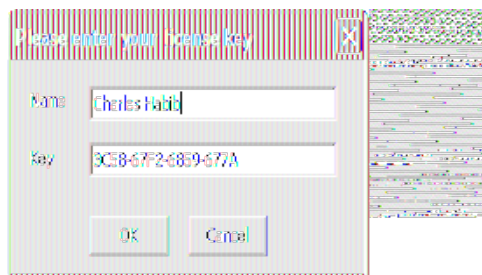
## 1.1 Availability and license protection

The NetCube Updater is distributed on standard NetCube CDs but it is sold as a separate product. You won't be able to use the executables provided on the CD unless you purchase a dedicated license key. Please get in touch with [info@netcube.ch](mailto:info@netcube.ch) to order such a license.

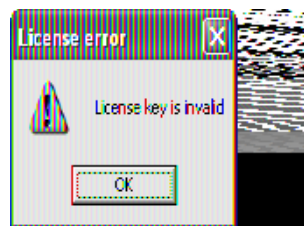
The license is always restricted to a single computer as identified by a MAC address of one of its network interfaces. When you order the license, please provide the MAC address and the name of the user of the license. You can find the MAC address by starting a command line (Start > Execute ... then type "cmd") and then executing "ipconfig /all". The MAC address looks like this "00-0C-6E-64-BB-14" (6 pairs of hexadecimal digits), it's sometimes called "physical address".

A license key is composed of 16 hexadecimal digits (grouped by four and separated by hyphens), ex: "3C5B-67F2-6859-677A".

When you start the NetCube Updater for the first time, you're presented with this screen and you have to indicate the license key that you just bought. Be sure to type the "Name" and the "Key" exactly like it has been sent to you by Codemat.



If you gave the wrong license, you'll see this error message and you'll have to try again:



## 1.2 Overview and concepts

The NetCube Updater is a program that can be used to automate the execution of commands on a large number of NetCubes. It is thus useful in the following situations:

- You want to configure/customize a large number of NetCubes.
- You want to upgrade a software on all your NetCubes.

The NetCube Updater hooks to the NetCube via the command line interface provided by the telnet service. It can also use the FTP service if the NetCube provides it.

The set of commands to execute is described in a "scenario file" (\*.netCubesScenario). You apply a scenario to a set of NetCube. You have 3 ways to create the list of NetCubes:

- Using a "scan functionality" to automatically add NetCube detected on the local network;
- Manually adding each NetCube;
- Loading a list previously saved (\*.netCubeList).

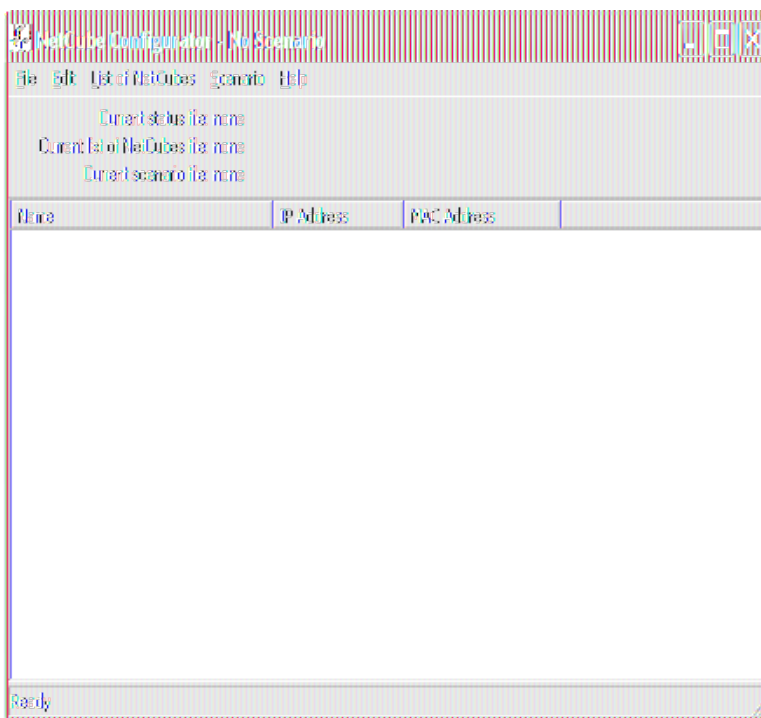
Each NetCube has a status in the process:

- Not updated: the scenario has not yet been run on this NetCube
- Updated: the scenario has been successfully run
- Failed: an error appeared during the scenario execution
- Scheduled: the scenario is running on another NetCube but this one is planned for later.
- Updating: the scenario is running on this NetCube.

You have the possibility to store the status of the whole process in a file (\*.netCubesStatus). In that way you can store the result of a first run and come later again to rerun it on NetCube which couldn't be updated in the first run (for example if they were unreachable on the network).

The software has an embedded scenario file that can be used to configure the network settings of new unconfigured NetCubes connected on the local network.

## 1.3 Discovering the user interface



The main window contains a menu bar, 3 summary fields and an empty list of NetCubes.

## 1.3.1 Menus

### 1.3.1.1 File menu

- **Save Update Status as...**  
Creates a file storing everything (List of NetCubes, actual scenario, current status of each NetCube).
- **Load Update Status...**  
Loads a file created with the previous menu entry.
- **Forget Status of NetCubes**  
Bring back all NetCubes to status "Not updated".
- **Exit**  
Quit the application.

### 1.3.1.2 Edit menu

This menu contains commands to select and deselect items based on several different criteria.

### 1.3.1.3 Menu "List of NetCubes"

- **Save List of NetCubes as...**  
Stores the list of NetCubes in a textual file. The file is mainly a list of IP address but it may also contain MAC address for unconfigured NetCubes and login/password information when that information has been supplied.
- **Load List of NetCubes...**  
Loads a complete list of NetCubes from a previously saved file.
- **Scan for unconfigured NetCubes**  
Detect unconfigured NetCubes (they share the IP address 192.168.0.100) by sending a broadcast request.
- **Scan for configured NetCubes**  
Detect configured NetCubes (they have an IP address which is not 192.168.0.100) by sending a broadcast request.
- **Add NetCube...**  
Adds a new NetCube to the list. You must supply its name, its IP address and maybe its MAC address (for unconfigured NetCubes only).
- **Delete select NetCube(s)**  
Removes the currently selected NetCube(s).
- **Change authentication information...**  
Changes the login/password used to connect to the selected NetCube. This information overrides the default login/password.
- **Change default authentication information...**  
Changes the default login/password used to connect to the NetCubes.

### 1.3.1.4 Scenario menu

- **Load update scenario...**  
Loads a new "update scenario" to use. During this operation new fields may be added to the list of NetCubes. You can double click on those fields to change the value. Each NetCube can thus be configured differently if needed.

- **Load standard scenario**  
Loads the standard scenario used to update the network settings of NetCubes.
- **Play scenario on selected NetCubes**  
Runs the scenario on all selected NetCubes. NetCube which are marked as "updated" are skipped.
- **Play scenario on all NetCubes**  
Runs the scenario on all NetCubes. NetCube which are marked as "updated" are skipped.
- **Show current scenario**  
Displays the actual scenario.
- **Check logs of selected NetCubes...**  
Displays the logs of the various "telnet sessions" managed during a play of the scenario. You can copy the logs in a text file.

## 1.3.2 Windows descriptions

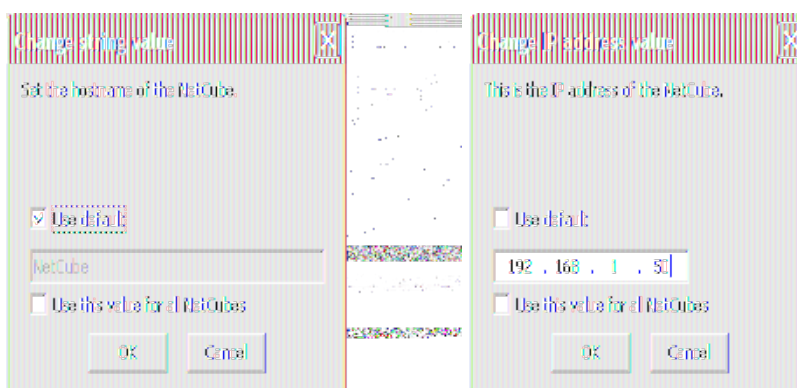
### 1.3.2.1 Main window



This screenshot shows the NetCube Updater with a list of two NetCubes. New scenario-specific fields have been added to the list (Address, DHCP, Domain, ...). You can double-click on the content of each cell to change a value. You can thus customize the behaviour of the scenario for each NetCube.

The "Status" field indicates that the scenario has not yet been run. You can click on the "Name" column header to sort the list by name of NetCube.

### 1.3.2.2 "Change ... value" window

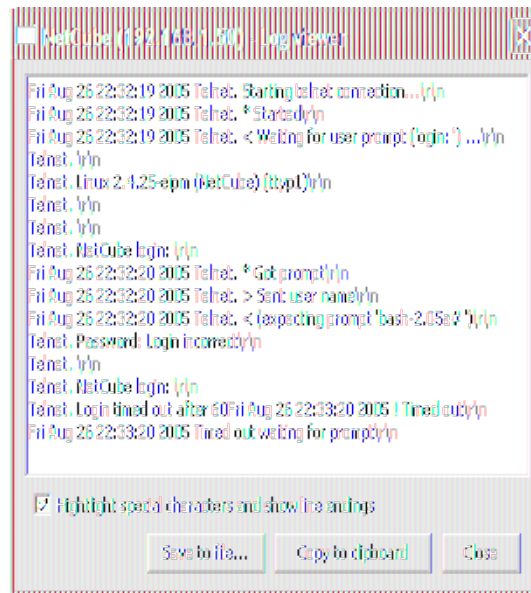


When you double-click on a value in the main list of NetCubes, the program will open a little "Change ... value" window. The appearance of the window depends on the type of the variable that you're changing.

By default, the program uses the default value supplied in the scenario file. If you want to use another value, you have to uncheck the "Use default" checkbox and to type the new value in the field below.

If you want to use the new value for all NetCubes (and not only for the one you clicked on), you can check "Use this value for all NetCubes".

### 1.3.2.3 "Log viewer" window



```

NetCube (192.168.1.50) Log Viewer
Fri Aug 26 22:32:19 2005 Telnet: Starting telnet connection.../n
Fri Aug 26 22:32:19 2005 Telnet: * Started/n
Fri Aug 26 22:32:19 2005 Telnet: < Waiting for user prompt (login: ...) /n
Telnet: /n
Telnet: Linux 2.4.25-epm (NetCube) (tty) /n
Telnet: /n
Telnet: /n
Telnet: NetCube login: /n
Fri Aug 26 22:32:20 2005 Telnet: * Got prompt/n
Fri Aug 26 22:32:20 2005 Telnet: > Sent username/n
Fri Aug 26 22:32:20 2005 Telnet: < (expecting prompt 'bash-2.05$') /n
Telnet: Password: Login incorrect/n
Telnet: /n
Telnet: NetCube login: /n
Telnet: Login timed out after 60Fri Aug 26 22:33:20 2005 ! Timed out/n
Fri Aug 26 22:33:20 2005 Timed out waiting for prompt/n

 Highlight special characters and show line endings
Save to file... Copy to clipboard Close

```

Once a scenario has been run on a NetCube, you can check the logs of what's happened. Special characters are encoded:

- Carriage return is "\r" and is in light grey
- New line is "\n" and is in light grey
- Other special characters are <XX> and are in blue. XX is a hexadecimal value of the character.

In this example we have a connection error, the NetCube is expecting a password but the NetCube Updater hasn't been configured to supply any.

You can store the logs in a file: the file will have "raw" data so that you can analyze it with an adapted editor.

### 1.3.2.4 Change (default) authentication



The same window is used for "Change default authentication" and "Change authentication". For each field checked, the NetCube Updater expects a corresponding prompt done by the NetCube. In the example above, we supplied a username and a password for all NetCubes (image on the left) except for one NetCube (image on the right) which only needs the username (this NetCube is probably not yet configured, it has no password set).

The only username that makes sense in the default setup is "root". You will never need to use the "Domain" field, this one is only here for completeness because some telnet implementations expect a domain (but the telnet on the NetCube doesn't require that information).

## 1.4 Technical details

### 1.4.1 Authentication

The scenario file may embed authentication information but that's not advised. The NetCube Updater will use the data from the first source where any authentication information is available:

- NetCube specific authentication information
- Login/password embedded in the scenario file
- Default authentication

The FTP authentication always requires a login and a password. Thus an unchecked "password" is translated into an empty password.

### 1.4.2 Automatic discovery of NetCubes

The "scan functionality" really sends an UDP broadcast request on port 1150. All NetCube which have the "udp-discover" program installed will respond by sending the informations required to be added to the list. The response also goes over UDP port 1150. This means that you can only discover NetCubes which are on the local network and which are receiving the broadcasted request.

You can check if a NetCube has udp-discover installed by checking if the file "/usr/sbin/udp-discover" exists. Try running this command in telnet:

```
# test -e /usr/sbin/udp-discover & echo "udp-discover is installed"
```

Note: You can install udp-discover on NetCubes by using the NetCube Updater and the corresponding "udeb" package. You just have to upload the package file in the NetCube and to run "udpkg -i udp-discover\_1.0-1\_i386.udeb". Ask the package to Codemat if you don't have it. The service will only be

active after a reboot.

### 1.4.3 Unconfigured NetCubes

NetCubes shipped by Codemat are all configured to use IP address 192.168.0.100 and they have no root password set. If you connect several unconfigured NetCubes on the same network, you can't dissociate them by their IP address (since they all share the same one). That's why you have to give the MAC address (example: "44-4D-50-80-02-E3"). The NetCube Updater will use "arp -s" to force the IP to MAC address resolution; that's how he communicates with individual unconfigured NetCubes.

## 1.5 Format of the scenario files

A scenario is a simple textual file with one command on each line. Some characters have a special meaning and thus need to be escaped/encoded:

- Carriage return: `\r`
- New line: `\n`
- Percent sign: `\%`
- A double-quote inside a double-quoted parameter: `\"`
- A space in a space delimited parameter: `\<space>`

Each command takes one or more parameters. If the command takes only one parameter, then everything after the command is taken as the value of the parameter (except trailing and leading spaces). By default multiple parameters are separated by spaces, but you can start a parameter with a double quote (") and then all characters including spaces are read until the next double-quote. If you need to include a double-quote in the parameter, you need to escape it by adding a backslash before it. To define an empty parameter, you must use the following syntax: ""

A line starting with a sharp (#) is a comment. The line is ignored by the parser but it's useful for you to describe the purpose of the various commands that you've written in a scenario file.

### 1.5.1 Descriptions of variables

Variable `<name>` `<type>` `<default>` `<long-description>`

Defines a new variable named `<name>` of type `<type>`. `<type>` can be string, integer, real, ip. `<default>` is the default value given to this variable, the user can then edit that value.

You can later use the value of the variable in the scenario by embedding `%VariableName%` in any parameter. That's why a real percent sign needs to be escaped by a backslash.

Four variables always exists, they are predefined:

- `%Name%`: the name of the NetCube
- `%IP%`: the IP address of the NetCube
- `%MAC%`: the MAC address of the NetCube in Unix format, ex: "AA:BB:CC:DD:EE:FF"
- `%MAC2%`: the MAC address in Windows format, ex: "AA-BB-CC-DD-EE-FF"

## 1.5.2 Telnet commands

TelnetPort <port>

Changes Telnet connection port. Default value: 21.

TelnetUserPrompt <prompt>

Defines a string, which is expected to be received before sending user name. Default value: "ogin: ".

TelnetPasswordPrompt <prompt>

Defines a string, which is expected to be received before sending password. Default value: "assword: ".

TelnetDomainPrompt <prompt>

Defines a string, which is expected to be received before sending domain. Default value: "omain: ".

TelnetLoginTimeout [<timeout>]

Defines maximum time that could be spent on establishment of telnet connection and sending of authentication information. Upon exceeding this time connection establishment is considered to be failed. Default value: 10 seconds. If <timeout> value is not specified, then it is considered to be equal to the default (10 seconds).

TelnetReadTimeout [<timeout>]

Default maximum time that could be spent waiting for a response according to a TelnetRead command. Upon exceeding this time command is considered to be failed. Default value: 1 minute. If <timeout> value is not specified, then it is considered to be equal to the default (1 minute).

TelnetStart [<user> [<password> [<domain>]]]

Starts the telnet session. Absent values are considered to be unneeded in an authentication procedure. In other words, if, for example, <domain> value is not specified, then domain prompt is not expected and domain value is not sent.

TelnetPrompt <prompt>

Indicates the prompt expected (by default "bash-2.05a# "). Note that recent NetCube have "netcube# " as prompt. Thus you will have to put such a statement in your scenario.

TelnetWrite <command>

Write data in the telnet session. If you want to execute a command, don't forget to send "\r\n" at the end of the command! Note: if you write data while data is available in the input, you'll have a warning in the logs. This probably means that you have a prompt badly configured or that you forgot to wait for the prompt.

TelnetRead <data>

Expects `<data>` as answer in the telnet session. If it isn't read then considers that the scenario failed and stops any further processing.

```
TelnetWaitPrompt [<timeout>]
```

Orders to wait until it receives the command line prompt. Times out automatically after `<timeout>` (default to 60) seconds if the prompt hasn't been received.

This command automatically skips any data received until it finds the prompt. If you want to check the output of the previous command, use "TelnetRead" before using "TelnetWaitPrompt".

```
TelnetStop
```

Closes the telnet connection. Note: usually you should only close the connection after reception of a prompt. This is needed to make sure that the previous command completed. Be sure to add a "TelnetWaitPrompt" before "TelnetStop".

```
TelnetEcho {on|off}
```

Enables or disables the echo sent by the telnet server. By default the echo is disabled. Enabling the echo means that any command that you send is written back to you.

## 1.5.3 FTP commands

```
FtpStart [<user> [<password>]]
```

Starts a FTP session with the given login/password if another login/password hasn't been provided by the user.

```
FtpPort <port>
```

Change the FTP connection port.

```
FtpModeActive
```

Select active FTP mode (with dynamic ports).

```
FtpModePassive
```

Select passive FTP mode.

```
FtpCd <directory>
```

Change the current directory on the server.

```
FtpPut <local> [<remote>]
```

Uploads a local file (`<local>`) to the NetCube (and names it `<remote>` if requested).

```
FtpGet <remote> [<local>]
```

Downloads a remote file (<remote>) and stores it locally. <local> can be a directory or a complete filename. If the destination file already exists, it's overwritten. If <local> is omitted, then the file is stored in the current directory.

You can use %IP%, %Name% or %MAC2% in <local> to customize the filename with a unique identifier of each NetCube. This lets you download the same file from many NetCubes and still store the files in a single directory.

```
FtpDel <file>
```

Deletes a file on the NetCube.

```
FtpStop
```

Closes the FTP connection.

## 1.5.4 Miscellaneous commands

```
Sleep <sec>
```

Sleeps (does nothing) during <sec> seconds.

## 1.6 Important remarks about scenarios

- Telnet always mangles single new-lines ("\n") in CR/LF ("\r\n"). Thus if you know that a program outputs "a\nb", you will have to read "a\r\nb".
- Variables are substituted without any escaping. Thus if you want to use a backslash you have to escape it. This can be useful to know if you want to indicate a path in a variable: you'll have to type "c:\mydir\myfile.txt".
- Default values of variables need to be escaped twice! Taking again the example above, a path needs to be written as "c:\\mydir\\myfile.txt". That's because the parser decodes the data once when it reads the variable line and a second time when it substitutes the variable by the value in the scenario.
- Don't forget to send "\r\n" at the end of a TelnetWrite command.

### 1.6.1 Known problems

Sometimes the commands that you send to the server are sent back (that's called the "echo"). Instead of receiving the output of your command, you're first receiving the command that you sent. This shouldn't happen since the NetCube Updater is explicitly disabling the "echo". But it happens nevertheless. If you encounter this problem you can work-around it by adding those 3 lines after your "telnetStart" line:

```
telnetEcho on
sleep 1
telnetEcho off
```

Another solution is to disable the echo from the command line with the "stty" program. However that program is not available on old NetCubes:

```
telnetWaitPrompt
telnetWrite stty -echo
```

```
telnetWaitPrompt
```

## 1.7 Examples of scenarios

In this section, you'll find some useful scenarios that you can use as basis to elaborate your own one.

### 1.7.1 Changing the root password

```
Variable NewPassword string "" Please define the new password to set to
  this NetCube.
TelnetStart
TelnetPrompt "netcube# "
TelnetWaitPrompt
TelnetWrite passwd\r\n
TelnetPrompt "assword: "
TelnetWaitPrompt
TelnetWrite %NewPassword%\r\n
TelnetWaitPrompt
TelnetWrite %NewPassword%\r\n
TelnetRead \r\nPassword changed.
TelnetPrompt "netcube# "
TelnetWaitPrompt
TelnetStop
```

### 1.7.2 Installing a package

```
Variable PackageFile string "c:\\\\netcube\\\\udp-discover_1.0-1_i386.udeb"
  The full path of the packages file.
FtpStart
FtpPut "%PackageFile%" package.udeb
FtpStop
TelnetStart
TelnetPrompt "netcube# "
TelnetWaitPrompt
TelnetWrite udpkg -i package.udeb\r\n
TelnetWaitPrompt
TelnetWrite rm -f package.udeb\r\n
TelnetWaitPrompt
TelnetStop
```

### 1.7.3 Checking the version of a software

```
TelnetStart
TelnetPrompt "netcube# "
TelnetWaitPrompt
TelnetWrite redirector -v\r\n
TelnetRead 3.0
TelnetWaitPrompt
TelnetStop
```

This scenario will fail on NetCubes which do not have the version 3.0 of the redirector. That's an easy way to check if all your NetCubes are up-to-date!