

...the broadest narrowband money can buy



User application

MORSE Utils

version 10.0.23.0
6/4/2010

Table of Contents

1. Morse Network Management Software	5
1.1. The most important applications	5
1.2. Application Control	5
2. Setr.exe	9
2.1. Macros	9
2.2. Default macro	10
2.3. Batch files	11
2.4. Parameters for Setr.exe	12
2.5. Shortcuts for pro Setr.exe	13
3. Netcnf.exe	14
3.1. Upload configuration from the CU to the PC	14
3.2. Download configuration form PC to CU	16
4. Memcp.exe	21
4.1. Example of downloading fw module E to the CU	21
4.2. Parameters for Memcp.exe	22
5. Netlock.exe	24
5.1. Local access	24
5.2. Local access with a password	26
5.3. Remote access	26
5.4. Parameters for Netlock.exe	27

List of Figures

3.1. Upload configuration	14
3.2. Acknowledgement for local uploading of configuration	16
3.3. Acknowledgement for ignoring calibration	17
3.4. Selection of configuration	18
3.5. Selection of the configuration file	18
4.1. Transfer of a firmware module	21

1. Morse Network Management Software

For configuring and controlling CUs (Communication Units, e.g. MR400, MR25), or alternatively for configuring and managing the whole MORSE system several control applications (Utils) are available. The application runs on Linux or Windows platforms. The DOS platform is not supported in sw version 586 and higher.

1.1. The most important applications

SETR.exe	configuration, diagnosis
NETCNF.exe	download / upload (backup) of complete configuration
MEMCP.exe	download firmware
NETLOCK.exe	locking / unlocking the CU, or the whole network

Common features for all utilities:

- The connection between the application and the MORSE network can be made via the RS232 serial channel between the PC and CU or via the Ethernet channel
- If the PC is connected to the CU via the serial channel only one application can be started at a time. When using Ethernet a arbitrary number of the same or different applications can run on the PC in parallel.
- Upon making a successful connection the application is connected to the MORSE network node in the connected CU. The application then carries out all operations using the system of services - it sends service packets to the respective nodes in the network and receives service messages.

1.2. Application Control

1.2.1. Connection to a CU (e.g. MR400, MR25) via the serial port

This uses serial communication with the MARS-A protocol. If you use a service cable (DSUB9-RJ12 or DSUB9-Jack3,5) from the COM port of your PC to the service connector of the CU (recommended connection), then the connection is made automatically upon starting the application with the default parameters.

It is necessary to count on the fact that user communication in the respective port on the CU (e.g. SCC0 in the case of MR400) is disconnected immediately as soon as you connect the service connector and the COM port on the PC via the service cable.

After making a successful connection a message appears containing the address of the connected node. Next the Main menu (for SETR) or the desired application operation (NETCNF, MEMCP, NETLOCK) appears. If a "no response" message appears we recommend starting the application with **appname.exe -mh** monitoring switched on for ascertaining failures in the connection (typically it is caused by use of a different port on the PC). **Appname.exe** is used here as alternative labelling for some of the applications, use for example, **setr.exe -mh**.

1.2.2. Connecting to the CU via Ethernet

Connecting the CU via Ethernet requires at least minimum configuration settings for IP communication. The easiest method for this purpose is to use Setr connected via the serial port; see above.

Basically there are two possible methods:

1. Direct PC - CU connection

For the CU configuration purposes there it is sufficient set the IP address and mask in CU only. Then connect the PC via crossed Ethernet cable and start Setr using the command e.g.:

```
setr -pIP192.168.0.1
```

Until 9.30 version inclusive it is necessary insert the non-zero MORSE address in the node 1 also.

The configuration must be completed for the communication through MORSE network. Configure e.g. MAS according to the manual MORSE Guide 3¹.

2. LAN

The CU can naturally be connected to a LAN and communication with it from the PC will either be direct or via a router (default gw). In the case of connecting the CU to a standard hub (switch) a standard (uncrossed) Ethernet cable is used. A standard ICMP ping may be employed on Ethernet to test the correct setting of IP routing in the MORSE equipment.

1.2.3. Application Parameters

Whereas the `SETR.exe` application can be started without entering parameters the entering of parameters is a necessity for `MEMCP.exe`. An overview of parameters for the respective application can be obtained using the command:

```
setr -h
```

Using command **System> Screen to file** change the list to a file and save it using a suitable name.

The parameters can be activated in more ways than one. The method by which the parameter is loaded has differing priority. The following overview of loading parameters is ordered from the lowest to the highest priority.

1. Not loaded

The application uses the default value - upon starting the application a parameter is not loaded and so the default value is used, for example, the parameter for selecting the communication port has a value of -p1

2. File

Parameters can be prepared in the file `appname.par`, for example `setr.par`, which will be used upon each start of the application, e.g. the contents of file `setr.par` are as follows:

¹ <https://www.racom.eu/eng/support/morse-m3/index.html>

```
-p2
-mh
# note - port p2, monitoring on
```

Then command **SETR** starts the application via port p2 and with monitoring of service packets running. The file `appname.par` may also contain a link to another file, e.g. `morse.par` in the form:

```
+morse.par
```

in which common parameters used by more applications may be stored.

3. Command line

Parameters may be used in the command line when starting the application, for example:

```
SETR -pIP192.168.0.1
```

Starts `setr.exe` communicating with the CU via the Ethernet channel.

4. ALT+P

In the running application it is possible to enter a new parameter after command **Alt+P**.

1.2.4. Common parameters for all applications

`-pb` bit rate

desired bit rate on COM port

possible values 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200

default value 19200

example `-pb57600`

`-p` com port No

No of com port on PC, which is used

default value 1

example `-p2`

`-pIP` address

IP address connected CU in decimal form

example `-pIP192.168.0.1`

`-pI` address

IP address connected CU v hexadecimal form

example `-pIC0A80001`

`-m` monitoring

monitoring mode of communication between PC and MCU

possible values h (= hex), a (= ascii), s(=size), o(=off)

default value off

example `-mh`

`-t` set up the timezone

the difference between the local time and the GMT

default values - none - a value from operating system is used

example -t-3600

-nt net layer timeout

timeout in ms, which application waits for the service report

default value 6000

example -nt15000

-nr net layer repeats

No of repeats, which application tries to get the service report desired

default value 2

example -nr10

-cl[filename] capture

everything from console is saved into the filename

example -cltest.cap

default value off

-cg[filename] macro send

macro from the filename is executed

example -cgtest.mac

default value off

1.2.5. Close of the application

ALT+X standard close

CTRL+BREAK, CTRL+C - forced close

2. Setr.exe

This is a powerful tool for manually or semi-automatically setting parameters, diagnosis and maintenance of the MORSE network. By means of the terminal type menu (MORSE main menu) it can control all CU services. A detailed description is provided in MORSE Firmware Documentation¹. Button macros and hot keys can also be used to control SETR.

2.1. Macros

A group of Setr commands can be incorporated into a macro and then launched with Setr running using command :

!mname Enter

where name is the name of the file with the macro. For example, below is a macro which sets up the frequency and output and reads the resulting values:

```
# example of macro xfreq.mac
!l^M
Q^M
Rpe^M
T4264750^M
R4264750^M
lA^M
wY^M
IY^M
SY^M^M
```

The macro is launched using command **!mxfreq Enter**. It uses Setr commands supplemented by **^M** meaning Enter. Other symbols used:

```
^M - Enter
~ - delay 1/2 sec
^S - Net Service Call mode (default)-
      waits for confirmation of command execution
^D - Net Direct mode - does not wait for confirmation
^X - end of operation and continuation with next address
# - note
```

If the name is generated in the following manner the macro may also be launched, for example, using button **F1**:

```
#This is an example, how to create a macro for keys CTRL/ALT/SHIFT F1..F10
#Macro file names are generated as follows:
#xxx_af01.mac for key ALT-F1
#xxx_cf10.mac for key CTRL-F10
#xxx_sf03.mac for key SHIFT-F3
#xxx__f04.mac for key F4
```

The macro - example for return to the local address and into Main menu:

¹ <https://www.racom.eu/eng/support/firmware/mr400/index.html>

The macro file name xxx_f01.mac
The content of file !l^M
 Q^M
The file location in the directory, where Setr.exe is
Calling the macro from running Setr by F1 key

2.2. Default macro

The default macro is designed to set up default parameters in the CU.

Macro xfac.mac

Macro xfac.mac employs the Setr command `f(a)ctory` setting, which inscribes default parameters in S-RAM memory for all menus (in the case of CU MR25 the parameters are written to FLASH memory). Next the macro sets the middle power level 7 in the menu `Rpe` and clears `Frequency off(s)et`. The Art tables and routing tables are cleared. Finally the macro restarts the CU and in this way the default parameters are copied from S-RAM memory into computing memory. The macro is designed first of all for working with the local connection and hence it sets local mode as its first step.

The macro is named `xfac.mac` and can be found at www.racom.eu², **Support** > **Firmware** > **Configuration macros** or **Download** > **Software Manuals** > **Utils Default macro**. File `xfac.mac` is located in Setr's current directory and is launched in Setr using command `!mxfac`.

Macro xde.mac

Macro `xde.mac` is more complicated but it offers additional possibilities for modifications. It successively visits all menus and sets them with default parameters and stores them to RAM and to S-RAM. Finally using command `sgB` it resets the CU. Below is an example of part of the memory:

```
# Default Ne
#
Q^M      ...return to MORSE Main menu
NfY^M~  ...insert default parameters to RAM
NwY^M~  ...insert default parameters to S-RAM

# Default Rpe # NOT ACTIVE #
#
#Q^M
#Rpe^M~
#fY^M~
#s0^M~
#l7^M~
#wY^M~  ...frequency and power settings remain unchanged
```

The macro can be modified as required, for example, by removing the six # symbols for menu `Rpe` it can also be set to default. It is recommended, for a CU without an ETH module, to use the # symbols for blocking the work of the last paragraph of the macro, which presets the ETH module.

² http://www.racom.eu/eng/support/default_macro.html

Macro xder.mac

In the CU with firmware 9.0.0.0 and higher macro `xder.mac` can be used. This is a version of the macro `xde.mac` which uses fewer idle periods and runs much faster.

2.3. Batch files

For simplifying application control, especially upon repeated use, it is possible to prepare batch files. The following example shows the following operations:

- submission of the prepared `mac2.mac` file for control
- storage of the original CU configuration
- setting of the default parameters in the CU
- entering of new address
- starting of `setr`

Batch file `config1.bat`:

```
echo off
echo check the address prepared in the macro mac2.mac
type mac\mac2.mac
```

```
echo STOP macro using Ctrl+C
echo storing original configuration in the file
pause 1
netcnf -aU -x
```

```
echo entering default parameter
pause 1
setr -cg.\mac\mac1.mac -x
```

```
echo entering new address
pause 1
setr -cg.\mac\mac2.mac -x
```

```
echo starting Setr for further work
pause 1
setr
```

```
rem you can continue the work with SETR
```

The macros used in the batch file:

- File `.\mac\mac1.mac`

```
# entering default parameter except for tables
Q^M
ca^M
Q^M
sgB^M~~~~~
```

- File `.\mac\mac2.mac`

```
# entering new address
Q^M
Ne^M
1a690F0004^M
w^M
I^M
Q^M
```

2.4. Parameters for Setr.exe

1. Common - see section Section 1.2.4, “Common parameters for all applications”
2. Special **-cg**, **-df**, **-dp**

Calling CU group :

setr -dfnnn -cgxtest.mac

Setr successively call addresses listed in the file `.\net\nnn.txt` and at each executes commands of macro `xtest.mac`.

setr -df -cgxtest.mac

If the name of the file is not stated after parameter `-df`, setr looks for file `.\net\anynt.txt` and calls the addresses contained in it.

Example of macro `nnn.txt`, which calls addresses 02, 03, 07 :

```
690F0002
690F0003
#690F0005
690F0007
```

Example of file `xtest.mac` , which reads the version of firmware in individual modems :

```
Q^M
sv^M
```

Calling with a path packet:

setr -dp.\net\ppp.ttt -cgxtest.mac

Setr calls to the destination address determined by file `.\net\ppp.ttt` according to the rules for the path packet.

setr -dpppp.ttt -cgxtest.mac

Setr uses file `ppp.ttt` located in the setr directory.

setr -dp -cgxtest.mac

Setr uses file `path.txt` located in the `setr` directory.

Example of file `ppp.ttt` for calling from address 1101 via address 5505 to address 3303 for the case where the connection between these addresses is not configured in routing tables:

```
690F1101
00000000
690F5505
00000000
690F3303
690F3303
```

2.5. Shortcuts for pro Setr.exe

With SETR running you may call up help by pressing Alt+Z:

Terminal mode help:

```
F1...F12 ..... User defined macros
Alt-S ..... Ascii send
Alt-G ..... macro list
Alt-I ..... Initialize connection
Alt-O ..... Pull jack
ALT-A ..... Save path
Alt-R ..... Use SerCall
Alt-P ..... Parameters
Alt-B ..... Scroll back
Alt-L ..... Capture toggle
Alt-X ..... Exit
..... GUI version only .....
Alt-C ..... Scroll lock ON/OFF
Alt-M/m .... Mark position before/after current
```

The meaning of main abbreviations:

```
Alt+I  initialization of connection PC-SETR
Alt+P  put the parameter into running SETR
Alt+B  displays containing SETR operations done till now
Alt+L  start/stop capture of operations into chosen file
Alt+C  stop/start screen refresh
Alt+X  stop SETR
```

3. Netcnf.exe

Replaces the former application `Uds.exe` and is used for upload and download of CU configuration structures. It is possible to use this application locally or remotely, e.g. upload configuration structures from all CU's in the network.

`Netcnf.exe` (when upload) saves configuration structures from FLASH or S-RAM memory of CU into `address.cnf` file, where `address` is the address of the connected node (locally or remotely), e.g. `690F0001.cnf`.

`Netcnf.exe` (when download) saves configuration structures directly into permanent memory in CU (FLASH, S-RAM). To use these parameters immediately it is necessary to put them into RAM too:

- by the command `(s)ervice (g)o go MORSE (B) thus sgB`
- or by the hw reset (off/on)

File `address.cnf`, which contains the CU configuration can be archived and downloaded to the same or another CU. It is important to make sure that more CUs with the same MORSE address are not created in the network!

If we need to make a change in the stored configuration, we can use from fw version 9.0.13.0 the `Setr` with `-af` parameter, e.g. `Setr -afcncf/anynet/690f0001.cnf`. In older versions we can download it to an auxiliary CU, use `Setr` to modify the configuration and again upload it to the file in the PC.

Since fw version 10.0.27.0 the CNI monitoring setting include `Split` is contained in the configuration file. This parameters had to be set using macro or manually before.

3.1. Upload configuration from the CU to the PC

3.1.1. Locally

- Start program **Netcnf.exe** with the CU connected by the service cable :

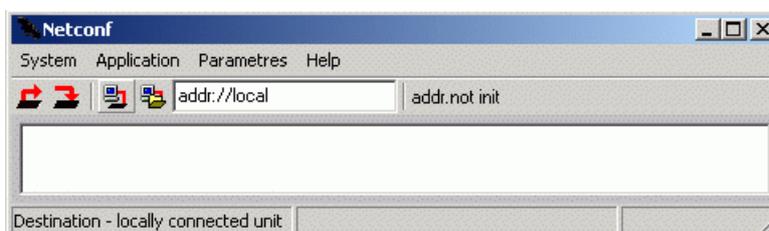


Fig. 3.1: Upload configuration

By clicking on icon *Configuration upload*  or selecting **Application** > **Configuration upload** we begin transmission of the configuration from the CU to the file in the PC. Upload ends by displaying a message containing the name and location of the generated configuration file:

```
File .\cnf\anynet\69000006.cnf created.
UPLOAD FINISHED
```

Press **Alt+X** to end program `Netcnf`.

- If we start the program with parameter *application Upload -aU* , it starts running without pressing icon:

Netcnf -aU

3.1.2. Remotely for 1 CU

For downloading the configuration from a remote CU we need to be locally connected to a CU which is connected to the remote CU via the MORSE network. The `Netcnf` application can be started in one of the following ways:

- Start **Netcnf**, click on icon *Destination local* , `addr://local` will appear in the window, replace word `local` by the destination address and press **Enter**, e.g. `addr://690F0003` and press **Enter**

Click on icon  to start downloading the configuration, which is stored in directory `.\cnf\anyinet\`

- Use parameter **-da** with the destination address, e.g.:

Netcnf.exe -da69000007

Click on icon  to start downloading the configuration.

- Use parameter **-aU** with the destination address (application Upload), e.g.:

Netcnf.exe -aU69000006

`Netcnf` starts running without clicking on the mouse. This is suitable method when generating batch files.

3.1.3. Remotely for a group of CUs

For downloading configurations from more CUs their addresses first need to be added to a text file, e.g. `aaa.txt`:

```
69000007
69000008
6900001A
```

Then we can use some of the following methods:

- **Netcnf** - start the application.

Click on icon *Destination file*  to open the dialogue window. Select a file, e.g. `aaa.txt`, which contains one or more destination addresses.

Click on icon  to start downloading configurations.

Configurations will be successively downloaded from these CUs and uploaded to the new directory `.\cnf\aaa\`.

- Start `Netcnf` with parameter **-df** and file, e.g. **bbb**:

Netcnf -dfbbb

File `bbb.txt` contains the addresses and is located in directory `.\net\`, i.e. `.\net\bbb.txt`.

Click on icon  to start downloading configurations.

Files with configurations are uploaded to directory `.\net\bbb\`

- Use parameter *application Upload -aU*:

Netcnf -dfbbb -aU

Upload begins without clicking the mouse, files are uploaded to `.\net\bbb\`

3.1.4. Table of options for Netcnf Upload

Local Upload			Result saved it:
Netcnf		click Upload	<code>.\cnf\anynet\xxxxxxx.cnf</code>
Netcnf		-aU	<code>.\cnf\anynet\</code>
Remota Upload for 1 CU			
Netcnf	click Dest local > addr	click Upload	<code>.\cnf\anynet\</code>
Netcnf	-da address	click Upload	<code>.\cnf\anynet\</code>
Netcnf	-da address	-aU	<code>.\cnf\anynet\</code>
Remote Upload for more CUs			
Netcnf	click Dest file > aaa.txt	click Upload	<code>.\cnf\aaa\</code>
Netcnf	-df bbb	click Upload	<code>.\cnf\bbb\</code>
Netcnf	-df bbb	-aU	<code>.\cnf\bbb\</code>

3.2. Download configuration form PC to CU

3.2.1. Locally

- Start program **Netcnf.exe** with CU connected via the service cable. Click on icon *Configuration download*  to open the dialogue window. Select the configuration file, which should be sent to the CU. **Netcnf** makes connection with the CU and a message appears:



Fig. 3.2: Acknowledgement for local uploading of configuration

By acknowledging **YES** downloading begins. Then the following question appears:

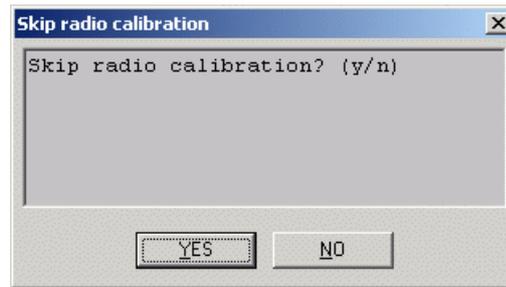


Fig. 3.3: Acknowledgement for ignoring calibration

By acknowledging **YES** we jump the part of the configuration file which contains the calibration constants of the radio part. These constants are individually set up in each CU during manufacture.

After downloading has finished the following message appears: `DOWNLOAD FINISHED`. By pressing **Alt+X** we exit `Netcnf`.

The configuration has now been uploaded to the S-RAM (previously FLASH) in the CU. It is uploaded into RAM when restarting (switching the CU on and off). It is possible to restart from `Setr` using command `sgB`.

- If we start the program with the parameter *application Download -aD* with the address of the file which contains the configuration then it begins running without clicking the icon:

Netcnf -aDcnf/anynet/690F0001.cnf

Acknowledge download with **YES** and ignore calibration with **YES**.

3.2.2. Remotely for 1 CU

For transfer the configuration to the CU we need to determine the address of the destination CU and the file with the configuration which needs to be written:

- Start `Netcnf`, click on icon *Destination local* , `addr://local` will appear in the window, replace word `local` with the destination address and press **Enter**, e.g. `addr://690F0003` and press **Enter**

Continue by clicking icon *Configuration download*  and select the file with the configuration. Acknowledge the control question by clicking **YES** and ignore calibration by clicking **YES**.

Note: until a hw or sw reset the remote station continues to work with the old configuration. Using `Setr` we can communicate with it and even modify the configuration (e.g. the address). Only after a sw reset `sgB` does the new configuration appear.

- Start `Netcnf` using the command with the destination address in parameter:

Netcnf -da690F0003

Continue by clicking icon *Configuration download*  and select the file with the configuration. Acknowledge download by clicking **YES** and ignore calibration by clicking **YES**.

- It is possible to designate the source file with the configuration by parameter **-aD**:

Netcnf -da690F0003 -aDcnf/anynet/690F0003.cnf

Netcnf starts running and we only acknowledge download by clicking **YES** and ignore calibration by clicking **YES**.

3.2.3. Remotely for a group of CUs

Add addresses to a text file, e.g. `aaa.txt` :

```
690F0002
690F0003
```

Continue in any of the following ways:

- Start **Netcnf** and using icon *Destination file*  select file `aaa.txt`. Continue by clicking icon *Configuration download* . A question appears:



Fig. 3.4: Selection of configuration

After selecting **YES** select the configuration file for the first node. Its address is given in the header of the table for selection of the configuration file:



Fig. 3.5: Selection of the configuration file

Acknowledge download by clicking **YES** and ignore calibration by clicking **YES**.

Once downloading has finished for the first node the selection table once again appears for the next node in sequence.

If we answered **NO** to the previous question `Do you want to upload a different configuration to each node ? (yes/no)`, then the configuration selected as the first configuration is uploaded to each node according to `aaa.txt`.

- Add file `aaa.txt` to directory `.\net\`, thus `.\net\aaa.txt`

Start Netcnf with parameter **-df** and file **aaa**:

Netcnf -dfaaa

Continue by clicking icon *Configuration download* . Acknowledge individual configuration by clicking **YES**, select `cnf_file.cnf`, acknowledge download by clicking **YES** and ignore calibration by clicking **YES**. Repeat selection of the file for other addresses from the list `aaa.txt`.

- Add file `aaa.txt` to directory `.\net\`, thus `.\net\aaa.txt`

Files with configurations can be written, for example, to directory `net1`:

`.\cnf\net1\690F0002.cnf` and `.\cnf\net1\690F0003.cnf`.

Start `Necnf` with parameters:

`Necnf -dfaaa -aD.\cnf\net1\690F0002.cnf`

Continue without clicking the mouse. If configurations for addresses listed in `aaa.txt` are prepared in directory `.\cnf\net1\` then simply acknowledge the steps by pressing Enter.

3.2.4. Table of options for Necnf Download

Local Download			Source of config. data		Down	Skip
Necnf		click Download	choice file.cnf		YES	YES
Necnf		-aDsoubor.cnf			YES	YES
Remote Download for 1 CU						
Necnf	klik Dest local > adresa	click Download	choice file.cnf		YES	YES
Necnf	-daaddress	click Download	choice file.cnf		YES	YES
Necnf	-daaddress	-aDfile.cnf			YES	YES
Remote Download for more CU			Diff conf			
Necnf	click Dest file > aaa.txt	click Download	YES	choice file.cnf	YES	YES
Necnf	-df_bbb	click Download		choice file.cnf	YES	YES
Necnf	-df_bbb	-aDfile.cnf	NO	choice file.cnf	YES	YES
					YES	YES

3.2.5. Parameters for Necnf.exe

Parameters can be used at work from command line.

- Common – see chapter Section 1.2.4, “Common parameters for all applications”
- Special
 - **-dl** local access
 - Upload (download) configuration from locally connected CU.
 - Example: `-dl`
 - Default value: `on`

- Configuration is saved into: `.\cnf\anynet\address.cnf`
- Example of command from command line with application of default value `-dl: netcnf`
- **-da[address]** remote access
 - Upload (download) configuration from remote CU of given address.
 - Example: `-da69DF1234`
 - Configuration is saved into: `.\cnf\anynet\69DF1234.cnf`
 - Example of command from command line: **netcnf -da69DF1234**
- **-df[file]** communication with CUs according to file
 - Netcnf.exe uploads configurations for all addresses (one by one) from `abc.txt` file. The file `abc.txt` must be put in `.\net` directory. If you do not specify filename, Netcnf looks for `.\net\anynet.txt`
 - Example: `-dfabc`
 - Configurations are saved into: `.\cnf\abc*.cnf` (* – for every address from `abc.txt` one file).
 - Example of command from command line: **netcnf -dfabc**
- **-aU** application Upload
 - Accomplishes the configuration upload without mouse click, suitable for the batch files.
 - Example: `-aU`
 - Example of command from command line: **netcnf -aU**
- **-aD[addr.cnf]** application Download local
 - Accomplishes the configuration download of file `addr.cnf` into locally connected CU without mouse click.
 - Example: `-aD69DF1234.cnf`
 - Netcnf.exe looks for file `address.cnf` in working directory. It is possible to specify complete path, e.g. `-aDc:\com\morse\cnf\anynet\69DF8606.cnf` or remaining part of the path only (e.g. from morse directory): `-aDcnf\anynet\69DF8606.cnf`
 - Example of command from command line: **netcnf -aDcnf\anynet\69DF8606.cnf**
- **-aD[address1.cnf] -da[address2]** application Download remote
 - Download the file `address1.cnf` into remote CU `address2`
 - Example: `-da69DF1245 -aDcnf\anynet\69DF1234.cnf`
 - Example of command from command line: **Netcnf -da69DF1245 -aDcnf\anynet\69DF1234.cnf**

4. Memcp.exe

Serves for downloading firmware, i.e. for recording program modules into the CU (e.g. into the MR400) via the service cable, Ethernet channel, IP network or through the MORSE network. Work with the MEMCP.exe application is described in more detail in the MORSE Guide documentation in the chapter Download firmware to the CU - introduction and in next chapters. Here we provide a short example.

4.1. Example of downloading fw module E to the CU

Start the application with the batch file which is prepared in the MORSE package together with the file memcp.exe:

```
mr900_cbl_le.bat
```

The same result is achieved by starting utility memcp.exe with the respective parameters:

```
memcp -nt5000 -nr20 -as8 -pS -pb115200 -af.\fkl\ce10.fkl %1 %2 %3
```

For it to function it is important that files memcp.exe, starter.exe and sub-directory fkl containing file ce10.fkl are stored in the current directory where the macro mr900_cbl_le.bat is located. If program setr.exe is running switch it off first.

Wait approx. 10 sec before communication is made and then monitor the transfer of packets indicated by symbols ###. Once transfer is complete end Memcp using command Alt+X. During the next 10s module ce10.fkl is copied within the CU and is prepared for operation.

```

Memory copy
System Application Parametres Help
local addr: FFFFFFFF
Morse API, ver. 7.45, (c) 2000, RACOM s.r.o., Czech republic.
Fri Oct 13 07:28:16 2006
address seek...: FFFFFFFF
MEMCP
module:3000001,size A4500; addr 100000; cpu ColdFire MCF5272
Sending seed request...OK.
begin:300014, end:3A4514
##### 10%
##### 20%
##### 30%
##### 40%
##### 50%
##### 60%
##### 70%
##### 80%
##### 90%
#####
Time elapsed 65103ms, transfer rate 82 kBit/s
DOWNLOAD FINISHED
Sending modprobe request with checksum...
MF_CODE_OK
addr: FFFFFFFF COM 1

```

Fig. 4.1: Transfer of a firmware module

Other macros `mr900_cbl_1...` are used in the same way :

```
mr900_cbl_ld.bat
mr900_cbl_le.bat
mr900_cbl_lg.bat
mr900_cbl_lh.bat
mr900_cbl_la.bat
mr900_cbl_lb.bat
```

For MR400, MC100, MR900 use macro `mr900_cbl_le.bat` and files `ce10.fkl ...`

For MR25, MCM302, MR25ET used macro `mr25_cbl_le.bat` and files `me10.fkl...`



Important

After fw download it is necessary to save the initial configuration into FLASH memory using Setr and command `(c)nf (b)ackup`.

Below is a general description of the macro. For a more detailed explanation see section Download firmware to CU - introduction. Macros for downloading individual modules:

```
starter memcp -nt5000 -nr20 -as8 -pS -af.\fkl\cd10.fkl -pb115200 %1 %2 %3 %4 %5
starter memcp -nt5000 -nr20 -as8 -pS -af.\fkl\ce10.fkl -pb115200 %1 %2 %3 %4 %5
starter memcp -nt5000 -nr20 -as8 -pS -af.\fkl\cg10.fkl -pb115200 %1 %2 %3 %4 %5
starter memcp -nt5000 -nr20 -as8 -pS -af.\fkl\ch10.fkl -pb115200 %1 %2 %3 %4 %5
starter memcp -nt5000 -nr20 -as4 -pS -af.\fkl\cb10.fkl -pb19200 %1 %2 %3 %4 %5
starter memcp -nt5000 -nr20 -as4 -af.\fkl\cb10.fkl -pb19200 %1 %2 %3 %4 %5
```

<code>starter</code>	start auxiliary program starter
<code>memcp</code>	calling the memcp program
<code>-nt5000 -nr20</code>	timeout and repeat for transmission through the network
<code>-as8</code>	length of transmitted packets
<code>-pS</code>	B-saver module active in the destination CU
<code>-pb115200</code>	signalling rate over service cable
<code>-af.\fkl\ce10.fkl</code>	calling source file <code>ce10.fkl</code>
<code>%1 %2 %3</code>	place for other possible parameters

When downloading various modules only the labelling of the source file `ce10.fkl` changes. For increased security when transmitting module B the length of the packet is shortened to `-as4` and the signalling rate is reduced to `-pb19200`.

4.2. Parameters for Memcp.exe

- Common – see chapter Section 1.2.4, “Common parameters for all applications”
- Special - example:
 - `-af[filename]`

- calling the source file
- example `-af.\fkl\ce10.fkl` for MR400 and MR900
- example `-af.\fkl\me10.fkl` for MR25 and MCM302
- **-pS**
 - communication with the Saver module

Other parameters are given in the example and in the detailed description.

5. Netlock.exe

Program `Netlock.exe` serves for locking the CU. Configuration data can be remotely read from a locked unit, however it is not possible to remotely execute the write or Init functions or start the monitoring or tests which would interrupt communication in the network. `Netlock.exe` is used by starting once with the respective parameters. There are 3 levels for locking and unlocking the CU:

```
0 root
1 racom
2 nobody
```

The level of access is defined by the name and password used (the original names are root, racom and nobody). The levels differ in their rights to change names and passwords, but the possibilities for locking and unlocking are the same. For local access where the CU is connected to the PC via the service cable it is possible to overcome locking using parameter **-aL**.

5.1. Local access

Unlocking the CU via the service cable:

```
netlock -aL -au
```

After acknowledging by clicking **y Enter** we receive the message `Successfully unlocked`

Key to parameters:

- aL local access, means no need to use a name or password. If the parameter is used for work in the distant CU then next PC must be connected to it using the service cable.
- au unlock - unlock station
- x exit - program netlock terminated after the given operation. If the parameter is not used the final messages can be seen. Exit the program using **Alt+X**.
- v0 If redundant messages appear when the application is running use this parameter to remove them.

Locking the CU via the service cable:

```
netlock -aL -al
```

- al lock - locks the stations

The status of the CU can be verified from MORSE Main menu, for example, using command **Ne Enter**, **Init Enter** - a locked station replies with the message `!Unit locked` and does not execute Init.

Inserting a new name:

The name for level 0 is `root`. The names for levels 1 and 2 can be set up using command:

```
netlock -aL -an2
```

- an1 name 1 - change level 1 name

-an2 name 2 - change level 2 name

Upon a query `Type new name` insert the name and acknowledge by pressing **y** . It is necessary to insert a new password after a name change.

Inserting a new password:

The password can be inserted for levels 0, 1, 2.

```
netlock -aL -ap2
```

-ap0 password 0 - change level 0 password

-ap1 password 1 - change level 1 password

-ap2 password 2 - change level 2 password

After inserting the same new password twice and pressing **y** the result is acknowledged with the message `Password changed.`

Simultaneously inserting a new name and password:

A name and password can be inserted for levels 1, 2.

```
netlock -aL -ab2
```

-ab1 both 1 - change level 1 name and password

-ab2 both 2 - change level 2 name and password

Viewing inserted names:

```
netlock -as
```

-as status - status of CU

The message obtained contains information about the locked/unlocked status and about the names assigned to levels 1 and 2. For local access only.

Testing a new name and password in local access mode:

Passwords saved to the CU cannot be read directly. It is possible to check the functionality of a password and where necessary insert a new password from a higher level or via local access. Checking a password in local access mode:

```
netlock
```

- Parameter **-au** for unlocking is a default setting and therefore does not need to be mentioned.
- Parameter **-aL** was not used and therefore the name and password are even requested in the case of local access.

A correct result is acknowledged with the message `Successfully unlocked.`

5.2. Local access with a password

If in the above commands we leave out parameter **-aL** then we need to use the name and password for level 0, 1 or 2.

Unlocking and locking the CU

These functions can be executed at each level.

Inserting passwords

- If we log on using the name and password for level 0, we can change the password for level 0, 1, 2.
- If we log on using the name and password for level 1, we can change the password for level 1, 2.
- If we log on using the name and password for level 2, we can change the password for level 2.

Inserting names, inserting names and passwords

- From level 0 it is possible to change the name for level 1, 2.
- From level 1 it is possible to change the name for level 2.

After inserting the name it is necessary to reinsert the password. Use parameter **-ab** to simultaneously change the name and password for the selected level.

When using parameter **-aMname** with the stated name we are not queried for the name by `Netlock` again. Similarly it is possible to use the parameter for inserting a new name **-aNname** when changing a name using parameters **-an**, **-ab**.

5.3. Remote access

This differs from local access in the use of parameter **-daADDRESS** (watch the character order!), which contains the address of the destination CU, and also the need to use a name and password.

Unlocking a remote station:

```
netlock -da69112233 -au
```

to a query `Login:` insert name,

to a query `Password:` insert password,

acknowledge **y**

The result is acknowledged with the message `Successfully unlocked`.

Other operations can also be performed remotely in a similar manner.

Reading names from a remote CU is not possible. However, if remote CU is connected to the PC via a service cable it then behaves as in the case of local access. This means that it is possible to perform the same operations with parameters **-da** and **-aL** as in the CU connected via the service cable.

5.4. Parameters for Netlock.exe

- Common - see chapter Section 1.2.4, "Common parameters for all applications"
- Special - parameters for Netlock application:

Application commands:

Locker options:

```
-as -al -au -aMnnn -aNnnn -ap[X] -an[X] -ab[X]
```

```
-al: lock
```

```
-au: unlock
```

```
-aMnnn: set my name
```

```
-aNnnn: set new name
```

```
-ap: change my password
```

```
-apX: change password for user level X
```

```
-an : change my name *
```

```
-anX: change name for user level X
```

```
-ab : change both my name and my password *
```

```
-abX: change both name and password for user level X
```

```
-as : locker status
```

```
-aL : local mode - in remote modem is jack
```

Lines marked by * make sense in local mode only.

My password and New password will be requested from the prompt.

- -al locking CU
- -au unlocking CU
- -aMnnn preset of user name
- -aNnnn inserting a new name
- -ap0 -ap1 -ap2 inserting a new password for level 0,1,2
- -an1 -an2 inserting a new name for level 1,2
- -ab1 -ab2 inserting a new name and password for level 1,2
- -as state CU
- -aL local mode