

## 2L FIELD DEVELOPER v 11.X

# COMMUNICATION AND HANDHELD CONFIGURATION

### 1. 2L Field on the handheld computer

This applies for pda's and handhelds with Windows CE 4.2/5.0, Pocket PC 2002/2003, or Windows Mobile 5.x/6.x operating systems.

**2L Field** normally is installed on the internal flash card or the internal flash memory.

If there is no internal flash card or flash memory available, than the installation is done on a Compact Flash or (micro) SD card.

This card has at least 4 folders

1. **2L**
2. **2L** Projects
3. Drivers
- 4 SVG

The **2L** folder must contain the following runtime programs :

**2L** the main program.

There will be a link to this program on your handheld desktop or program menu. This program will allow you to select one of the existing projects (in the 2L Projects folder). Within the selected project you can select your form (or start the START.FRM application) to begin the actual data collection process.

**2LForm**

this program handles the actual data collection in forms mode.

**2LSheet**

this program handles the actual data collection in spreadsheet mode

**2LInit**

this program is only needed during (re)installation of **2L Field** on the handheld.

2L.txt

The language file of the main program

2Lform.txt

The language file of form and sheet modules

The **2L** Projects folder contains one or more project or module folders. All forms, datasets, lists, pictures etc. needed for a specific project must be present in this particular project folder.

For Projects grouped into modules, all general files (the common files for all the projects in a module), forms, lists, pictures etc can be placed in the module folder. All the project specific files (datasets, project related lists etc..) must be present in the project subfolder.

The Drivers folder will contain all standard "drivers" for barcode-scanning, weighing scale reading, transponder scanning etc.

User-developed and /or 3rd party developed drivers can also be placed in this Drivers folder.

Drivers communicate partly via the registry with the **2L** runtime programs, so if you are planning to develop your own drivers, please ask for the documentation on driver development.

The SVG folder will contain all the programs and utility files for the integrated viewer.

The program SVGinit will (re)install the viewer on the handheld



## 2. Automatic backup feature

During installation, **2L Field** can be set up with the automatic backup feature installed on a (separate) memory card (another Flash card, SD card or the handhelds internal memory card).

With this automatic backup facility installed, the risk to lose valuable data will be minimal.

The backup feature will make a daily log-file, and will log all major activities during data collection with a date and time stamp.

Logged activities are : opening and closing of forms and datasets and adding, deleting or changing records in the dataset.

The daily log-file will be named something like B2L20041024.DAT

B2L stands for Backup **2L**, 2004 denoted the year (YYYY), 10 denotes the month (MM) and 24 denotes the day (DD)

After a crash on the handheld you can recover your data by importing the appropriate backup log-file from the backup card into Excel on your PC.

When you selected "Full backup" during installation, each time you open a dataset in one of your projects, the original dataset will be copied to a backup project folder.

With this original dataset, together with the backup log-file, recovery will be must easier, because you only need to recover the changes in your dataset since the last opening.

## 3. 2L Field on the pen computer

This applies for (tablet) PC's and laptops with Windows XP, Vista, or Windows 7 operating systems.

**2L Field** normally is installed on a local harddisk in the 2L folder (e.g. C:\2L).

Normally this installation folder has at least 5 sub folders

1. Progs
2. 2L Projects
3. Drivers
4. Backup
5. SVG

The exact location of the different folders is saved as INI file (2lfieldxp.ini) in the "progs folder (C:\2L\progs) and can be adapted if you need a different configuration, like backup on a different (local) drive.

The Progs folder folder must contain the following runtime programs :

**2L\_Field** the main program (includes the form and sheet modules).

This program will allow you to select one of the existing projects (in the 2L Projects folder).

Within the selected project you can select your form (or start the START.FRM application) to begin the actual data collection process.

2Lxp.txt

The language file of the main program

2Lformxp.txt

The language file of form and sheet modules

No separate initialization program is needed. The first time you start the main program, you are prompted to register **2L Field** on your local PC.

The 2L Projects folder contains one or more project or module folders. All forms, datasets, lists, pictures etc. needed for a specific project must be present in this particular project folder.

For Projects grouped into modules, all general files (the common files for all the projects in a module), forms, lists, pictures etc can be placed in the module folder. All the project specific files (dataset, project related lists etc..) must be present in the project sub-folder

The Drivers folder will contain all standard "drivers" for e.g. barcode-scanning, weighing scale reading, transponder scanning etc.

User-developed and /or 3rd party developed drivers can also be placed in this Drivers folder.



Drivers communicate partly via the registry with the **2L** runtime programs, so if you are planning to develop your own drivers, please ask for the documentation on driver development.

The SVG folder will contain all the programs and utility files for the integrated viewer.

The Automatic Backup functionality is the same as described in chapter 2, without the “full backup” feature.

## 4. Communication

Normally the handheld computer will be placed on a cradle, which is connected via a serial or USB-cable to a PC. With Active Sync (or Mobile Device Center) running on your PC, an automatic connection can be established.

With **2L Connect** or **2L Comms Manager** installed on the PC, files can be easily transferred to and from the handheld.

**2L Connect** (which is part of the professional version of **2L Field Developer**) uses Active Sync as the connection interface between handheld and PC. **2L Comms Manager** is a separate comms manager program on the PC which uses the Comms driver (one of the standard **2L** drivers) on the handheld. This program is not a standard feature of **2L Field Developer**.

Without **2L Connect** or **2L Comms Manager**, you can use Active Sync (or the Mobile Device Center) to transfer new projects or new forms and data sets to the handheld and collected data on the handheld to the PC,

With **2L Connect** or **2L Comms Manager**, this process can be done partly or fully automatic, combined with conversion of **2L** data sets (#-separated DAT files) to and from CSV files (;-separated Excel files), DBF files (Dbase tables), XML files (Web based data tables), XLS files (Excel spreadsheets) etc.

**2L Connect** can only handle one handheld at a time (because Active Sync does not allow multiple connections), **2L Comms Manager** however can handle up to 255 handhelds connected via a physical or virtual serial connection to the PC where the manager is running.

### 4.1 2L Connect

**2L Connect** is a user friendly tool to transfer your data to and from the handheld. To make **2L Connect** work on your handheld you only have to establish a connection between your handheld and Active Sync. Once this connection is made you can start **2L Connect**.

The program **2L Connect** can be used in manual mode and in automatic script-mode.

In manual mode you start **2L Connect**.

A window will pop up with a display of the selected projects on the PC and the handheld, and 2 tabs . The tabs indicate 'Manual' and 'Script'.

In the 'manual' tab you can manually transfer data from and to your PC. Changing the active project on the PC and /or on the handheld can be done by double-click on the appropriate project-name. The program will let you select another project or let you create a new project. In the manual tab, again there are two more tabs, 'From PC to Handheld' and 'From Handheld to PC'. On the left side, a selection list will be shown where you can select individual files with a mouse click. On the right a listbox (initially empty) is shown. Any transfer action will be listed here.

With the “>>” button you can send the selected files. With the “All” button you can select all files in the selection list. With the “None” button you can deselect all files.

Files transferred will appear on the right side with the message <sent> behind the files. To facilitate the search process for a certain file you want to transfer, you can filter files with a certain extension eg. \*.DAT files.



Via the 'Script' tab you can create your own scripts for automatic file transfer to and from the handheld. Script files will have the extension .2LS. Scripts can be useful in case you have to carry out the same file transfers regularly. An integrated Script-editor will help you to create your scripts.

In automatic script-mode, you can start a 2LS script file from the Windows Explorer (select the program **2L Connect** as default program to run 2LS files). During execution of the selected script an activity log is shown. At the end of the script the activity log can be closed, which will also close the **2L Connect** script.

## 4.2 Script commands to be used with **2L Connect**

- ALERT** <Text>  
Display alert dialog-box showing <Text>
- APPEND** <file1> <file2>  
Add contents of <File1> to <File2>
- BYE**  
Close connection with handheld
- CLOSE**  
Close connection and exit 2L Connect, if the script could be completed successfully
- CONVERT** <conversion> <infilespecs> <outfilespecs>  
Do automatic conversion using <conversion>.exe file
- COPY** <oldfilename> <newfilename>  
Copy <oldfilename> to <newfilename>, overwrite <newfilename> on the PC.
- CREATEINDEX** <datafile> <Indexname>  
Insert (as first column) an indexcolumn in <datafile>, giving <datafile> a unique indexcolumn.  
The indexvalues are taken from the record-numbers, starting from 1.
- CWD** <newdir>  
Change working directory on the PC to <newdir> Eg. 2L\data\map2
- DEL** <filespecs>  
Delete the files in <filespecs> from the PC.
- DELREG** <RegName>  
Remove <Regname> from the registry on the HH
- DIR** <filespecs>  
List the files in <filespecs> in the log.
- ECHO** <text>  
Add <text> to the log
- ERROR** <Text>  
Display error dialog-box showing <Text>
- GET** <filespecs>  
Get the files specified in <filespecs> from the handheld.
- GETNEW** <Filespecs>  
Get newer files specified in <Filespecs> from the Handheld
- GETX** <Filespecs>  
Get not existing files specified in <Filespecs> from the Handheld
- GET2LVERSION**  
Reads the installed version of 2L Field, and saves this in a file 2LVERSION.TXT
- GETHHID**  
Get the serial number from the handheld and save it as HHID.TXT on the PC, and use it as variable \_HHID



#### GETHHIDSHORT

Get the serial number from the Handheld (in variable \_HHID) and save the last 4 characters as HHIDSHORT.TXT on the PC and in variable \_HHIDSHORT

#### GETHHPROJECTNAME

Save the active project-name as HHPROJECT.TXT on the PC

#### GETPROJECTSLIST (<Projectsspecs>)

Save the list of available projects as file HHPROJECTS.LST on the PC

#### IFFILE <FileName>

Test if <Filename> exists

#### JOIN <Dataset1> <Dataset2>

Join datasets, remove empty records from <dataset1>, update <Dataset1> with records from <Dataset2>, replace existing records, add new records (based on recid)

#### LOADFROMREG <RegName> (<Varname>)

Get the value from the register-variable <Regname> and save in the local variable <Varname>, or <RegName> if <Varname> parameter was omitted

#### LOCAL

Do not use Active sync comms, but connect to the local 2L Field XP projects on the PC  
Note : this command does not have to be specified on the first line of the script.

#### MD <newdir>

Create <newdir> on the PC, or change to this <newdir> if it already existed.

#### RENAME <oldfile> <newfile>

Rename <oldfile> to <newfile> on the PC

#### RUN <Program> (<Parameters>)

Execute <Program> (with optional <Parameters>)  
The script will wait for the program to finish. If the program is canceled, the script will fail.

#### SAVE <FileName> <Text>

Create <FileName> on the PC, add <Text> to this new file

#### SAVETOREG <RegName> <Value>

Fill register-variable <Regname> with <Value> on the HH

#### SELECT <Listfile> <Varname> (<Prompt>)

Pop up a listbox to let the user select an entry from the list. The selected item is saved in <Varname>.

#### SEND <filespecs>

Send all the files as specified in <filespecs> to the handheld.

#### SENDNEW <Filespecs>

Send all newer files as specified in <Filespecs> to the handheld

#### SENDX <Filespecs>

Send all non-existing files as specified in <Filespecs> to the handheld

#### SILENT

Do not display the progress list during execution of this script

#### TYPE <filespecs>



Display the contents of the files in <filespecs> in the log.

=== REMOTE COMMANDS ===

REMOTE COPY <oldfilename> <newfilename>

Copy <oldfilename> to <newfilename>, overwrite <newfilename> on the handheld.

REMOTE CWD <newdir>

Change working directory on the handheld to <newdir> Eg. \iPAQ File store\data\map2

REMOTE 2LDIR

Change to the 2L Field program folder on the handheld

REMOTE DRIVERSDIR

Change to the Drivers folder on the handheld

REMOTE SVGDIR

Change to the SVG folder on the handheld

REMOTE BACKUPDIR

Change to the Backup-folder on the handheld

REMOTE DEL <filespecs>

Delete the files in <filespecs> on the handheld.

REMOTE DIR <filespecs>

Get a file-list specified in <filespecs> in the log from the handheld.

REMOTE IFFILE <Filename>

Test if <Filename> exists on handheld

REMOTE MD <newdir>

Create <newdir> on the handheld, or change to this <newdir> if it already existed.

REMOTE MODULE <ModuleName>

Select <Module> on handheld, module must exist

REMOTE PROJECT <Project>

Select <Project> on the handheld. (In the 2L Projects folder)

If <Project> does not exist, create this project.

REMOTE RD <Project>

Remove the empty project folder <Project> from the handheld

REMOTE RENAME <oldfile> <newfile>

Rename <oldfile> to <newfile> on the handheld.

REMOTE RUN <ProgramName>

Start <ProgramName> on the handheld

=== DEBUG AND OTHER COMMANDS ===

DEBUG ON

Run the commands step by step and show the result of each step.

DEBUG OFF

Continue running the commands in normal mode.

DEBUG FILE

Run the commands and log the results to the file debug2ls.txt in the \_2L\_Connect folder

ONFAIL :<Label>  
continue from line :<Label> if one of the next commands fails  
Note a label MUST start with a colon (:)

GOTO :<Label>  
continue from line :<Label>

:LABEL  
:linelabel

--- COMMENT ----  
;comment  
lines beginning with a semicolon (;) will be regarded as comments  
execution of the script will continue with the next line

## 4.3 Using scripts to handle exceptions

A Script will be executed starting from the first line until:

- \* the script is finished
- \* an empty line is encountered
- \* a command failed

In case a command fails you can use ONFAIL, GOTO and labels in your script to handle different possibilities. See next example

```
ONFAIL :NOTEXISTING
REMOTE IFFILE TEST.DAT
GOTO :EXISTING
```

```
:NOTEXISTING
ALERT File does not exist
GOTO :NEXT
```

```
:EXISTING
GETNEW TEST.DAT
GOTO :NEXT
```

```
:NEXT
; continue with the script from here
```

## 5 Using 2L Comms Manager

The comms manager is available in 3 versions.

- The "normal" comms manager: \_2L\_COMMS\_MANGER.EXE
- The "normal" comms manager extended with modem communication:  
\_2L\_COMMS\_MANAGER\_MDM.EXE
- The Multi comms manager: \_2L\_MCM.EXE

### 5. 1. Basic functionalities for all versions.

### 5.1.1 Registration

The comms manager needs to be registered on the PC before it can be used.

If this registration has not been done correctly, a license code will be shown and the program will prompt you to register.

Using this license code, you can get a registration code at Linx mobile solutions.

Note: multiple versions of the comms manager programs can be installed on the same PC, but each installation needs its own registration.

### 5.1.2 Configuration with CM.INI

CM.INI is the configuration file, where you can specify which ports need to be handled by the comms manager.

A sample CM.INI with modem extension could look like

```
Portnr#Conntype#Comport#Baudrate#Retries#A#R
1###115200#3#-#-
2###9600#5#v#-
4#MODEM#2#19200#7#-#v
12###9600#5#v#-
```

\*Portnr (Portnumber) is mandatory

Portnumbers not mentioned (like 3 and 5 till 11) will be shown as:

Com <P>: Not used

\*Conntype (Connectiontype), only used with modem extension

MODEM : you must specify (under "Comport") the physical comport connected to your modem

GPRS : you must specify (under "Comport") a virtual comport installed that connects to the IP-gateway for this handheld

<not specified> : the HH is connected to the physical serial port <Portnr>

\*Comport

See Conntype MODEM/GPRS

\*Baudrate

If not specified:

115200 for direct cable connection

9600 for modem/gprs connection

\*Retries

Maximum number of retries allowed for a script to complete.

If the script could not be completed within these retries, a mail can be sent automatically (see 5.1.3 MAIL.INI)

Retries default to 5

\* A (Active)

v : this port is included in the loop

- : this port must be excluded from the loop

\* R (Repeat retry)

v : use retries for this port

- : do not use retries for this port

Note Conntype + Comport are only valid in the extended modem-version of the comms manager

### 5.1.3 MAIL.INI

If a script could not be completed (within the specified number of retries) a mail can be sent automatically to a manager.



A mail will also be sent if a script contains an error (unknown command, ERROR command). Via the configuration file MAIL.INI the mailing can be specified.

A sample mail.ini:

```
SMTP=linxmobile.nl
From=jan@linxmobile.com
To=jan@linxmobile.com
CC=jan@linxmobile.com
BCC=jan@linxmobile.com
Subject=SMTP mail test
Body=test.txt
```

Note: if no MAIL.INI exists, the mail-functionality will not be used.  
You can change the MAIL.INI at runtime, the comms manager will reload the specifications when a mail needs to be sent.

## 5.1.4 Commands list

Each script will be carried out line by line until:

- \* the last line
  - if this last line was carried out correctly, the script will be finished, unless an ALERT line was encountered
- \* an empty line is encountered
  - if the previous command was carried out correctly, the script will be finished unless an ALERT line was encountered
- \* the command could not be executed (e.g. the HH did not react) and no ONFAIL was specified
  - the script will stop and will be restarted at the next retry
- \* the ERROR command is encountered
  - the script will stop (without retry) showing the error message
- \* the RETRY command is encountered
  - the script will stop now and will be restarted at the next retry

===== Remote-commands (REMOTE or REM) =====

REMOTE 2LDIR

Select the 2L-folder as the active folder on the HH (e.g. to update 2L and start 2linit)

REMOTE BACKUPDIR

Select the Backup-folder as the active folder on the HH

REMOTE COPY <filemask1> <filemask2>

f.e. REMOTE COPY \*.DAT \BACKUP\\*.DAT

REMOTE CWD <newfolder>

f.e. REMOTE CWD ..  
or REMOTE CWD \TMP  
or REMOTE CWD .

this last command will activate the original active project folder on the HH

REMOTE RENAME <Filespec1> <Filespec2>

Rename <Filespec1> to <Filespec2>

REMOTE DIR (<filespec>)

list remote directory (filter on <filespec>)

REMOTE DEL <File>/<Filespec>

Remove <File> or <Filespecs> on the HH

REMOTE DRIVERSDIR

Select the Drivers-folder as the active folder on the HH (e.g. to update the drivers)

REMOTE MD <newfolder>

<newfolder> will be created (if not existing) and selected

REMOTE MODULE <modulename>

Select module <modulename> on the HH



The module MUST exist  
REMOTE PROJECT <projectname>  
    Select project <projectname> on the HH  
    The project will be created if it did not exist  
REMOTE RUN <Filename>  
    Execute <Filename> on the HH  
REMOTE SVGDIR  
    Select the SVG-folder as the active folder on the HH

===== local PC based commands =====

;Comment  
    To use comments in a script file  
:<Label>  
    This is not a real command, but must be used in combination with ONFAIL and GOTO  
ALERT <Alert-message>  
    Show alert status (in red), and add ==><Alert message><== to log  
    Fail at end of the script file  
APPEND(FILE) <File1> <File2>  
    Add contents of <File1> to <File2>  
    Fail if <File2> could not be appended to <File1>  
BYE  
    close remote comms  
    Fail if HH not connected  
BYE CLOSE  
    close and exit remote comms  
    Fail if HH not connected  
CLOSE  
    close and exit remote comms  
    Fail if HH not connected  
CONVERT <Conversion> <InFilespecs> <Outfilespecs> <Template>  
    do automatic conversion by running <Conversion> with parameters and  
    wait for conversion program to exit  
    (same as RUN command, but compatible with 2L Connect scripts)  
    Fail if conversion could not be completed  
CD / CWD <newdir>  
    change to new directory on the PC  
    Fail if <newdir> not existing or accessible  
COPY <OldFileName> <NewFileName>  
    Copy <OldFileName> to <NewFileName, overwrite <NewFileName>  
    (wildcards are allowed)  
    Fail if <NewFileName> could not be created  
DEBUG ON  
    Turn debugging on  
DEBUG OFF  
    Turn debugging off  
DEL(ETE) <File>/<Filespec>  
    Delete <File> / <Filespecs> on PC  
    Fail if delete fails  
DELREG <RegName>  
    Delete register variable <RegName> on the HH  
DIR <filespecs>  
    show PC directory  
    Never fails  
ECHO <Message>  
    show <Message>  
    Never fails  
ERROR <error message>  
    Show <error message> and halt script



A mail will be generated (see MAIL.INI)

GETHHID  
get unit id/serial-number, saved as HHID.TXT  
Fail if HH not connected, or the HH does not have a unit id/serial-number

GETHHPROJECTNAME  
Get name of active project from HH and save as HHPROJECT.TXT  
Fail if HH not connected, or no (active) project exists

GETHHPROJECTSLIST (<Projectsspecs>)  
Get a list of existing projects (filter on <Projectsspecs> if specified) from HH and save as  
HHPROJECTS.LST  
Fail if HH not connected

GET <File>/<Filespec>  
Copy <File> /<Filespecs> from HH to PC  
Fail if HH not connected, or GET not succeeded

GETNEW <File> / <Filespecs>  
Copy only newer file(s) from HH to PC, other files will be skipped

GETX <File> / <Filespecs>  
Copy only non-existing file(s) from HH to PC, existing files will be skipped

GOTO :<Label>  
Continue execution of script-file from :<Label>

IFFILE <FileName>  
Check if <FileName> exists on PC

LOAD <filename>  
Load first line of contents from <filename> into \_TMP variable  
You can use this variable in subsequent commands (specify as %\_TMP%)

LOADFROMREG <RegName> (<VarName>)  
Get a user-defined register variable from the HH and save in Memory variable <VarName>  
If <VarName> was omitted, <RegName> will be used as <VarName>

LOCAL  
Command without action, just for compatibility with 2L Connect scripts

MD <Newfoldername>  
Create new folder <Newfoldername> on de PC

MODEM <Modemcommand>  
See MODEM commands and phone-list

ONFAIL :<Label>  
Continue execution from :<Label>, if one of the following commands fails

RENAME <OldFileName> <NewFileName>  
No wildcards allowed, <NewFileName> can have a different path from <OldFileName>,  
so you can use RENAME also to MOVE files

RETRY  
Stop executing this script now and restart at next retry

RUN <Filename> (+ parameters)  
Execute <FileName> (+ parameters) on the PC, wait until execution stops  
Use "Filename" if filename contains spaces

SAVETOREG <RegName> <Value>  
Fill HH register-variable <RegName> with <Value>

SEN(D) <File>/<Filespec>  
Send file(s) from PC to HH

SENDNEW <File> / <Filespecs>  
Send only newer file(s) from PC to HH, other files will be skipped

SENDX <File> / <Filespecs>  
Send only non-existing file(s) from PC to HH, existing files will be skipped

SILENT  
Do now show or log the individual script lines, only the results

TYPE <File>/<Filespec>  
display contents of file(s)

===== MODEM commands and phone-book =====  
(This only applies to the extended modem version of the comms manager)



The general MODEM command is given as  
MODEM <Modemcommand>  
e.g. MODEM ATZ

MODEM commands are NOT meant for the HH, but to initialize a modem connection with a HH.  
With MODEM commands we assume that the Modem ECHO is on, and the modem answers with text-strings (OK, ERROR etc..)

Two modem commands will be treated separately:

MODEM CALL (<tel.number>)

If <tel.number> is omitted the tel.number to call will be searched for in the phone-book (PHONELIST.DAT file)  
See sample phone-book

MODEM HANGUP

This will try to get the modem to listen and drop the line.

===== Sample phone-book =====

Portnr#Telnr

1#0,123456789

10#0,003 98765432

## 5.1.5 Comms-driver on the HH

To be able to communicate with a HH, the HH must be installed with a Comms driver, compatible with the communications manager.

For all CE based HH's (Pocket PC, Windows Mobile, Windows CE) a general Comms driver is available.

Special comms drivers are available for

Kestrel communication via cradle

Eagle communication (both cradle and direct cable)

The Eagle comms driver does not recognize the following commands:

REMOTE "specialdir" (2LDIR, BACKUPDIR, DRIVERSDIR, SVGDIR)

Registry commands (LOADFROMREG, SAVETOREG, DELREG)

Both Kestrel and Eagle comms driver do NOT implement modem-initialization.

The general comms driver will use a file named MODEM.INI to set up a modem connection.

## 5.2 Special functionalities for the different versions

### 5.2.1 "Normal" comms manager

Each active port needs a subdirectory named HH(0)<Portnr> (HH01 for Portnr 1, HH12 for Portnr 12 etc.):

the HH-folder for this port.

For each active port the communication session will start with checking for a get script (GET.SCR) located in the HH-folder.

If this get script is available, it will be executed.

When this get script is executed successfully, it will wait for the a send script (SEN.SCR) located in the same HH-folder.

When this send script is executed successfully, it will display "Comms. Finished".

After 10 wait-loops, it will restart waiting for the get script again.



MODEM commands will be treated as unknown commands.



## **5.2.2 "Normal" comms manager with modem extension**

The same functionality as 5.2.1, but with MODEM commands implemented.

## **5.2.3 Multi comms manager (MCM)**

MCM must be started with the name of a script-file to be executed.

It starts with checking available and connected HH's.

Now for each connected HH it will try to execute the script file.  
If no script file was given, it will only check for connected HH's.

MODEM commands will be treated as unknown commands.

