

m5cmd.pl - v 0.2

David Lapsley
dlapsley@haystack.mit.edu

March 26, 2004

Contents

1	Introduction	3
2	Platform Requirements	4
3	Installation	4
4	Usage	4
5	Examples	5
5.1	m2m	5
5.2	h2m	6
5.3	m2h	7
5.4	sc	7
5.5	dc	8
6	Troubleshooting	8
7	Copyright	8

1 Introduction

This script automates the transfer of scans between a pair of Mark5's and between a Mark5 and a UNIX host.

The tool runs in three different modes:

- *m2m* In this case the script is executed on a host (it could be either of the Mark5's or a completely independent host). The script creates a TCP/IP connection to the m5drive control port of each mark5 and transfers the scans listed in a "scan_file" from one of the Mark5's (the source) to the other. The script automatically invokes the "disc2net" and "net2disc" commands on each machine, with the appropriate parameters at the appropriate time.
- *h2m* In this mode the script is executed on the host on which the scans are stored (on a regular operating system disc - as opposed to a Mark5 disc pack). It creates a TCP/IP connection to the m5drive control port of the recipient Mark5 and uses this to control the execution of the "net2disc" command. The script automatically invokes the "File2net" command on the host and the "net2disc" command on the Mark5, with the appropriate parameters and at the appropriate time in order to transfer files from the host to the Mark5.
- *m2h* In this mode, the script is executed on the host which is to receive the scans and store them on a regular operating system disc (as opposed to a Mark5 disc pack). It creates a TCP/IP connection to the m5drive control port of the source Mark5 and uses this to control the execution of the "disc2net" command. The script automatically invokes the "Net2file" command on the host and the "disc2net" command on the Mark5, with the appropriate parameters at the appropriate time.
- *sc* In this mode, the script will perform a scan_check of the list of scans passed into it.
- *dc* In this mode, the script will perform a sequence of data_check's of the list of scans passed into it. The "steps" parameter specifies how the size in bytes of the steps to take between successive data checks of the same scan.

2 Platform Requirements

- Hosts must have PERL installed and in their PATH environment variable
- Hosts must be running UNIX (preferably Linux)
- “File2net” and “Net2file” commands must have been installed
- Mark5’s must be running release “## 2004 March 23 (day 083)” of the Mark5 software and associated drivers (including “File2net” and “Net2file”).

3 Installation

Installation is in two parts:

1. copy the libraries and scripts to a Linux/UNIX server.
2. set the environment variables specified below

Table 1: Environment Variables

Environment Variable	Description
PERL5LIB	location of m5tk perl modules
EXEC_DIR	location of perl scripts
M5CMD	location of m5cmd.pl script
FILE2NET_EXEC	location of File2net executable
FILE2NET_PROTOCOL	protocol to use for File2net

A sample file is included in the distribution (“init.sh”).

4 Usage

```
m5cmd.pl -help
         -version
         -src=<src_ip>
         -dst=<dst_ip>
         -srcport=<m5driveport>
```

```

-dstport=<m5driveport>
-scans=<scan_file>
-prefix=<results prefix>
-step=<step size in bytes>
-file2net_exec=<path of File2net>
-mode=[m2m|m2h|h2m|sc|dc]
-protocol=[tcp|udp]

```

- **-version**: prints version information
- **-help** : prints help information
- **-src**=< *src_ip* >: in “m2m” mode the IP address or name of the source Mark5.
- **-dst**=< *dst_ip* >: in “m2m” mode the IP address or name of the destination Mark5.
- **-srcport**=< *src_port* >: in “m2m” mode the TCP port of the control port of the source Mark5.
- **-dstport**=< *dst_port* >: in “m2m” mode the TCP port of the control port of the destination Mark5.
- **-prefix**=< *results prefix* >: not currently used.
- **-scans**=< *scan_file* >: a file containing the list of scans to be transferred.
- **-mode**=[*m2m|m2h|h2m|sc|dc*]: the transfer mode: “m2m”=Mark5 to Mark5, “m2h”=Mark5 to Host, “h2m”=Host to Mark5, “sc”=scan_check, “dc”=data_check.

5 Examples

5.1 m2m

Here is an example of a transfer between two Mark5s.

The following command:

```

./m5cmd.pl -src=192.168.1.1 -srcport=2620 -dst=192.168.1.2
          -dstport=2620 -scans=scans.txt -mode=m2m

```

Will create control connections to the source and destination Mark5's and then issue commands to the Mark5's to transfer the scans listed in the file "scans.txt" from the source Mark5 to the destination Mark5.

The scans.txt file has the same format as the output of the "DirList" command on a Mark5. For example:

1	M5.03010001.dat	0	2884932152
2	M5.03010003.dat	2884932152	10505451400
3	M5.03010005.dat	10505451400	14315711208
4	M5.03010008.dat	14315711208	16474858592
5	M5.03010010.dat	16474858592	19922226016
6	M5.03010011.dat	19922226016	29357133456

Reading from left to right, the fields are: scan number, scan name, starting byte and ending byte.

5.2 h2m

Here is an example of a transfer from a host to a Mark5.

The following command:

```
./m5cmd.pl -dst=192.168.1.2 -dstport=2620 -scans=scans.txt
-mode=h2m
```

Will create a control connection to the source and destination Mark5's and then issue commands to the Mark5 and to the local host to transfer the scans listed in the file "scans.txt" from the host to the destination Mark5. Note that the script must be executed on the host on which the scan files reside. The "File2net" program must be in the PATH of the shell executing m5cmd.pl on the host.

The scans.txt file is a list of filenames stored on the host. For example:

```
1 M5.03010001.dat
2 M5.03010003.dat
3 M5.03010005.dat
4 M5.03010008.dat
5 M5.03010010.dat
6 M5.03010011.dat
```

5.3 m2h

Here is an example of a transfer from a Mark5 to a host.

The following command:

```
./m5cmd.pl -src=192.168.1.1 -srcport=2620 -scans=scans.txt  
-mode=m2h
```

Will create a control connection to the source Mark5 and then issue commands to the Mark5 and to the local host to transfer the scans listed in the file “scans.txt” from the source Mark5 to the destination host. Note that the script must be executed on the host on which the scan files will reside after the transfer.

The scans.txt file has the same format as the output of the “DirList” command on a Mark5. For example:

1	M5.03010001.dat	0	2884932152
2	M5.03010003.dat	2884932152	10505451400
3	M5.03010005.dat	10505451400	14315711208
4	M5.03010008.dat	14315711208	16474858592
5	M5.03010010.dat	16474858592	19922226016
6	M5.03010011.dat	19922226016	29357133456

Reading from left to right, the fields are: scan number, scan name, starting byte and ending byte.

5.4 sc

Scan check mode.

The following command:

```
./m5cmd.pl -src=192.168.1.2 -srcport=2620 -scans=scans.txt  
-mode=sc
```

Will create a control connection to the source Mark5’s and then issue commands to the Mark5 and to the local host to scan_check the scans listed in the file “scans.txt”. This can be done in one command using the included “ScanList” shell script.

The scans.txt file has the same format as the output of the “DirList” command on a Mark5. For example:

1	M5.03010001.dat	0	2884932152
2	M5.03010003.dat	2884932152	10505451400
3	M5.03010005.dat	10505451400	14315711208
4	M5.03010008.dat	14315711208	16474858592
5	M5.03010010.dat	16474858592	19922226016
6	M5.03010011.dat	19922226016	29357133456

5.5 dc

Data check mode.

The following command:

```
./m5cmd.pl -src=192.168.1.2 -srcport=2620 -scans=scans.txt
          -step=1000000000 -mode=sc
```

Will create a control connection to the source Mark5's and then issue commands to the Mark5 and to the local host to data_check the scans listed in the file "scans.txt". Each scan will have a sequence of data_checks executed on it with the step (in bytes) between consecutive data_check's given by the '-step' parameter.

The scans.txt file has the same format as the output of the "DirList" command on a Mark5. For example:

1	M5.03010001.dat	0	2884932152
2	M5.03010003.dat	2884932152	10505451400
3	M5.03010005.dat	10505451400	14315711208
4	M5.03010008.dat	14315711208	16474858592
5	M5.03010010.dat	16474858592	19922226016
6	M5.03010011.dat	19922226016	29357133456

6 Troubleshooting

If you have any problems, please feel free to contact me: dlapsley@haystack.mit.edu.

7 Copyright

The software is released under the MIT license (please take the time to read it, as you will find that is one of the least restrictive software licenses around). Please feel free to distribute, modify, etc. if you find this software useful - subject to the license below. I would really appreciate it if you could

let me know of any modifications you make to the software, so that I can include these in the main branch.

Copyright (c) 2003 MIT Haystack Observatory

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.