The Little Movie Processing Centre

Uwe Girlich,

uwe@half-empty.de

v3.4.4, 10/24/2009, (alpha test release)

This document is the user's guide of The Little Movie Processing Centre (LMPC). LMPC is a utility to handle DOOM, DOOM][, HERETIC, HEXEN and STRIFE LMP files. There is a rudimentary support for Duke Nukem 3D and Redneck Rampage DMO files. Quake DEM files, QuakeWorld QWD files, Quake II DM2 files (client recorded, server recorded, relay files), and Quake III Arena DM3 files are supported too. The DEM file cutter tool (DEMcut), The DEM Text File Analyser (DEMA), The Demo Broadcasting Server (DBS) and The DM2 file concatenator (DM2cat) are part of this package as well.

Table of Contents

1. Introduction

LMPC is the name of the complete package and the name of the biggest parts in it: The Little Movie Processing Centre.

LMPC is a collection of utilities to handle old (<=1.4) and new DOOM, DOOM][, HERETIC, HEXEN and STRIFE LMP files. There is support for Duke Nukem 3D and Redneck Rampage DMO files, Quake DEM files, QuakeWorld QWD files, Quake II DM2 files (client recorded, server recorded, and relay files), and Quake III Arena DM3 files too. For Quake II relay files, read the home page of the Quake II Relay project by Conor Davis (cedavis@planetquake.com (mailto:cedavis@planetquake.com)) http://relay.planetquake.gamespy.com/

You can download the package from my Demo Specs page (http://demospecs.half-empty.de).

It is based upon the information in the Unofficial LMP format description (http://demospecs.half-empty.de/lmp), the Unofficial DMO format description (http://demospecs.half-empty.de/dmo), the Unofficial DEM format description (http://demospecs.half-empty.de/dem), the Unofficial QWD format description (http://demospecs.half-empty.de/qwd) and the Unofficial DM2 format description (http://demospecs.half-empty.de/dm2).

Read the apropriate document to find out the exact version numbers supported by LMPC. I write the format description and the program at the same time; so if it is described in the specs document, it is in general supported by the LMPC program.

2. LMPC, The Little Movie Processing Centre

2.1. Introduction

With LMPC you can get information about a LMP, DMO, DEM, QWD, DM2, or DM3 file, change the version and viewing player's number, add some wait tics at the end, remove the pause tics, clear the "Save Game" tics in LMP files and (this is the main part) you can compile and decompile a LMP, DMO, DEM, QWD, DM2, or DM3 from and to a LS file. LS stands for "*L*MP *S*ource". To reduce the amount of new abbreviations I call the "DMO Source" LS too. The language is similar and will be treated in the same way. The real LS language is documented at full length in the section Section 7. The "DEM Source" (see section Section 8) and all the more the "QWD Source" (see section Section 9) is totally different but I call it LS anyway. The same holds for Quake II DM2 (see section Section 10) and Quake III Areana DM3 files.

2.2. Command line parameters

LMPC is a command line oriented program. It uses the `getopt_long()` function for the command line parsing.

The LMP file convert options (`--change-version`, `--change-player`, `--add-WT`, `--remove-pause` and `--clear-save`) can be used together.

2.2.1. Syntax description

Syntax:

```
lmpc
```

Description:

If you start LMPC without any command line parameters you receive the following syntax message:

```
Little Movie Processing Centre
LMPC (c) U. Girlich, 1994-2006, Release 3.4.4 10/24/2009 (alpha test release)
lmpc [option ...] filename [filename ...]
-h, --help                  display this help and exit.
-V, --version               output version information and exit.
-i, --info                  prints out information about the LMP-files.
-s, --to-txt                binary/text to text conversion.
-l, --to-bin                binary/text, ... to binary conversion.
-v, --change-version VERSION changes the version of a LMP-file to VERSION
                            (0 means old DOOM).
-p, --change-player PLAYER   changes the recording player of a LMP-file to
                            PLAYER (0..3).
-w, --add-WT SECONDS         adds SECONDS (int) WT game tics to the LMP file
-r, --remove-pause           removes the game tics between PS and PE.
-c, --clear-save             clears SG tics.
-f, --fix-multi              fixes multi-level DEM and DM2 files.
-t, --tic FROM(,|:)TO        converts only a part of the LMP-file
                            (in connection with -s only).
-g, --game GAME              forces game (GAME=DOOM,DOOM2,HERETIC,HEXEN,STRIFE
                            DUKE,REDNECK,QUAKE,QUAKEWORLD,QUAKE2,QUAKE3)
                            short: D2HXSNRQWqA (only with -(i|s|v|p|w|r|c)).
-m, --mark-step STEP         marks every STEPth tic in the LS file (-s only)
                            STEP=0 doesn't mark any game tic; default: STEP=1.
```

2.2.2. Information about a LMP, DMO, DEM, QWD, DM2 or DM3 file

Syntax:

```
lmpc [--info|-i] infile [infile ...]
```

Input File:

`infile`s

Description:

LMPC reads each `infile` and prints out some information like game, version number, episode, map, skill, number of players, deathmatch/altdeath, nomonsters etc. Please note: in the old DOOM format there is no version information, so if you check e.g. a 1.2 LMP file you receive only the vague information v<1.4 instead of v1.2. LMPC doesn't report any version for HERETIC and HEXEN LMP files.

This option works for DMO files too. They will be detected automatically but you can use the `--game DUKE` option as well.

This option works for DEM, QWD, DM2 and DM3 files too. They will be detected automatically but you can use the `--game QUAKE`, `--game QUAKEWORLD`, `--game QUAKE2`, and `--game QUAKE3`, option as well. The information on DEM, QWD, DM2, and DM3 files nearly zero beside the information that it is actually a DEM, QWD, DM2, or DM3 file and the number of frames in binary DEM, QWD, DM2, or DM3 files.

To get more information of a DEM file convert it into a text file and use DEMA, The DEM Text File Analyser.

2.2.3. LMP/DMO/DEM/QWD/DM2/DM3 text converter (decompiler)

Syntax:

```
lmpc (--to-txt|-s) [(--tic|-t) FROM(,|:)TO] [(--mark-step|-m) STEP] infile
txtfile
```

Input File:

```
infile
```

Output File:

```
txtfile
```

Description:

LMPC reads the file `infile`, converts it to the corresponding text format and stores the result in the text file `txtfile`. `infile` may be a LMP binary, DMO binary, DEM binary, DEM text, QWD binary, QWD text, DM2 binary, DM2 text, DM3 binary or DM3 text file. Most of the additional options do work with binary LMP files only.

If the `--tic` option is used, only the part between gametic `FROM` and `TO` (inclusive) will be stored in the LS file. A missing `FROM` means the start of the LMP file and a missing `TO` means the end of the LMP file. `FROM` and `TO` are separated by a comma (`,`) or a colon (`:`). The MS-DOS version may have difficulties with comma as a separator because `COMMAND.COM` (`4dos` as well) splits a command line at spaces and commas. The `TO` part will become a different argument and might be interpreted as a file name.

LMPC appends at the end of the LS file some useful additional information. For each player there is a statistics about his favourite movement codes and the portion of `WT`. Since you can achieve with keyboard/joystick only a few special movement codes, is is very easy to detect, if the player used a keyboard or a mouse. Some newbies don’t use strafe, so LMPC shows also this handicap. The statistic takes in account always the full LMP file.

From the favourite movement codes you can easily deduce the number behind a prospective `-turbo` parameter.

If the `--mark-step` option is used LMPC will mark every `STEP`th game tic with the tic-number and the current time. The default `STEP` value is 1. If `STEP`=0 no marks are printed at all.

Any `--mark-step` option in conjunction with a binary DEM, QWD and DM2 files is a special debug-option. It includes in the text file the frame number and the starting address of all messages. This option may be removed or changed in future releases.

The `--fix-multi` option can be used together with `--to-txt` to repair the broken multi-level DEM and DM2 files.

2.2.4. LMP/DEM/QWD/DM2/MD3 binary converter (compiler)

Syntax:

```
lmpc (--to-bin|-l) infile [infile ...] binfile
```

Input File:

`infile`s

Output File:

`binfile`

Description:

LMPC reads the first file, determines the game type and compiles a binary file out of all the files on the command line. The binary (destination) file is always the last argument.

This option works with LMP text files, DEM text, DEM binary, QWD text, QWD binary, DM2 text, DM2 binary, DM3 text, and DM3 binary files as input files only. LMPC can handle more than 1 input file with LMP text files only.

For combination of DEM files use DEMcut.

The `--fix-multi` option can be used together with `--to-bin` to repair the broken multi-level DEM and DM2 files.

2.2.5. LMP file version change

Syntax:

```
lmpc (--change-version|-v) VERSION lmpfile1 [lmpfile2]
```

Input File:

`lmpfile1`

Output File:

`lmpfile1` or `[lmpfile2]`

Description:

LMPC reads the binary LMP file `lmpfile1`, changes the version number byte in `lmpfile1` to the appropriate value or stores the result in `lmpfile2`.

The VERSION can be the version byte itself (0 for the old LMP files) or anything like 1.4beta, 1.9 or whatever.

There is no guarantee for a special LMP file to work with the new version!!

2.2.6. LMP file player's view change

Syntax:

```
lmpc (--change-player|-p) PLAYER lmpfile1 [lmpfile2]
```

Input File:

`lmpfile1`

Output File:

`lmpfile1` or `[lmpfile2]`

Description:

LMPC reads the LMP file `lmpfile1`, changes the number of the player (0..3: new DOOM, DOOM][; 0..7: STRIFE) with the main view (status bar, sound) in `lmpfile1` or stores the result in `lmpfile2`.

To change only the main player byte works only with new LMP files, since old ones are always recorded from the view of the first player (no. 0) and there is no information about this in the header.

2.2.7. LMP file "Wait" (WT) tics addition

Syntax:

```
lmpc (--add-WT|-w) SECONDS lmpfile1 [lmpfile2]
```

Input File:

`lmpfile1`

Output File:

`lmpfile1` or `[lmpfile2]`

Description:

LMPC reads the `lmpfile1` and adds at the end

```
SECONDS * 35 * player number
```

empty (`WT`) game tics. This results in a `SECONDS` seconds longer LMP file.

With `lmpfile2` the result will be written in `lmpfile2`. Without this parameter, `lmpfile1` will be changed.

This option is very useful for contest LMP files with a long visible summary screen.

2.2.8. LMP file clean-up: "Pause" (`PS`, `PE`) tics

Syntax:

```
lmpc (--remove-pause|-r) lmpfile1 [lmpfile2]
```

Input File:

```
lmpfile1
```

Output File:

```
[lmpfile2]
```

Description:

LMPC reads the `lmpfile1` and copies all game tics to `lmpfile2`. During the copy process LMPC scans the contents for Pause Start game tics (`PS`) and Pause End game tics (`PE`).

The `PS` game tic will be copied to `lmpfile2` but with an empty "use byte"; the game tics between `PS` and `PE` (inclusive `PE`) wont be copied.

Without the second filename LMPC will write to a temporary file and replaces at the end `lmpfile1`.

The game tic numbers of `PS` and `PE` are printed during the conversion.

You may have difficulties with the play back of such a patched LMP file. This is the case if you patch a LMP file of a level with a revenant (skeleton). The revenant uses a random number generator to choose between a self controlled rocket and a normal rocket. The input of this generator are the last two bits of the game tic number. This means you have to make sure that you cut a number of game tics divisible by 4. In general this is impossible so do not use pause in levels with a revenant. An arch vile makes similar difficulties but I don't even know how to prevent it.

2.2.9. LMP file clean-up: “Save Game” (SG) tics

Syntax:

```
lmpc (--clear-save|-c) lmpfile1 [lmpfile2]
```

Input File:

```
lmpfile1
```

Output File:

```
[lmpfile2]
```

Description:

LMPC reads the `lmpfile1` and copies all game tics to `lmpfile2`. During the copy process LMPC scans the contents for Save Game game tics (`SG`). Every `SG` command will be cleared. With `lmpfile2` the result will be written in `lmpfile2`. Without this parameter, `lmpfile1` will be changed.

2.2.10. DEM/DM2 file repair: multi-level files

Syntax:

```
lmpc (--fix-multi|-f) demofile
```

Input File:

```
demofile
```

Output File:

```
demofile
```

Description:

Quake and Quake II have problems with the playback of multi-level recordings. They usually stopp at the first level change.

Read more about the background in the recording FAQ (http://demospecs.half-empty.de/faq/faq-2.html#ss2.1).

If the `--fix-multi` parameter comes without any conversion option on the command-line, LMPC repairs the demo file in question "in-place". This means LMPC overwrites the bad commands with `nop` or `print`.

The `--fix-multi` option can be used together with `--to-bin` and `--to-txt`. LMPC removes the offending commands in this case.

2.2.11. Game type determination

Syntax:

```
lmpc (--info|
      --change-version|
      --change-player|
      --to-txt|
      --add-WT|
      --remove-pause) (--game|-g) GAME binfile [filename ...]
```

Input File:

`binfile`

Output File:

action dependent

Description:

LMPC determines the game type of a LMP (binary), DMO (binary), DEM (text and binary), QWD (text and binary), DM2 (text and binary), or DM3 (text and binary) file very good. But it can't good distinguish between HERETIC/HEXEN and old DOOM LMP files. Moreover it can't distinguish at all between DOOM][Map 1-9 and new DOOM Episode 1 LMP files. The latter case is not very important but the first is more severe. To force LMPC to use a particular game type use `--game`. `GAME` can be `DOOM`, `DOOM2`, `HERETIC`, `HEXEN`, `STRIFE`, `DUKE`, `REDNECK`, `QUAKE`, `QUAKEWORLD`, `QUAKE2` or `QUAKE3` or the short form `D`, `2`, `H`, `X`, `S`, `N`, `R`, `Q`, `W`, `q`, or `A`.

2.3. Bugs and limitations

To distinguish between old DOOM, HERETIC and HEXEN LMP files I use a simple turn-byte-average algorithm. This fails, if the LMP file is too short or you turn always in the same direction and never go forward. Due to the header change in HEXEN 1.1, it is nearly impossible to distinguish fast between HERETIC and HEXEN 1.1 LMP files. LMPC looks then at the file date and treats files with an creation year >=1996 as HEXEN and all others as HERETIC files. This is totally stupid but what else can I do?

LMPC supports client side recorded DM2 files, DM2 files recorded by the console command *serverrecord* (working from Quake II version 3.15) and the RLA file used by the Quake II Relay project. One single DM2 file can even contain parts from all these sides. Quake II may only playback client side recordings.

If you produced a good recording file (at least some minutes and with some action) and LMPC can not detect the game type, please send it to me. Up to a corrected version make sure to use the `--game` option. Some hints to solve the problem are welcome as well.

The `*number` repeat code in LS files really repeats this line `number` times and does not care about multi player-game tics.

The MS-DOS binary of LMPC converts all file names to lower case to simplify the name handling. The `tolower` C-function doesn't converts any kind of special characters like ö or è. This can cause some severe data lost.

I use a redundancy in DEM files to detect Quake cheaters. My simple check will fail if the player gets hit and he gets a medikit at the very same time. This can happen, if a player is standing over a medikit, when the hit comes.

Some versions of Quake use a slightly different DEM format. LMPC/DEMcut can read this "new" format but they write the old 1.06 format. SQuake 1.07 for Linux behaves similar: it writes the old format but can play back the old and the new format.

Keygrip 2 writes DM2 files with non-conform bit mask values in *sound*, *spawnbaseline* and *packetentities* messages. LMPC tries its best to decompile even such bad files too but it may well be, that some other messages with bit masks have the same problem.

Quake III Arena uses bit fields for all its game messages. A server-to-client-message (or a block in a DM3 file) uses always full bytes. It follows, that some bits in the last byte are not used by the game. LMPC set these bits always to zero but Quake III Arena does not. Therefore it is impossible for LMPC to reproduce a totally identical DM3 file. The file from LMPC will behave the same as the original but some bits are zero.

The DM3 text parser is really a joke. I implemented it in about two hours. It cannot cope with non-ordered entries in any block. Thus be sure not to shuffle the entries! If this happens, the program will certainly crash or at least not work as expected. If this is a problem, I can always extend the old parser to parse DM3 texts too but this would take at least a week of hard work.

3. DEMA, The DEM Text File Analyser

To analyse a DEM file convert it into a text file and use DEMA. DEMA is a very simple Perl script, which may give you the time per level, the level name and the average frame time (client update time).

This Perl script is total incomplete but it may be helpful yet. There are no command line switches. You have to edit the script to change the summary.

I wrote DEMA in Perl, which is very good at pattern matching and extraction. I will *never* write a C version, so write one yourself or install Perl.

The main aim of DEMA is to show a sample framework for any kind of DEM text analysing.

4. DEMcut, The DEM file cutter tool

4.1. Introduction

DEMcut is the DEM file cutter tool. I don't know anything about movie cutting whatsoever but DEMcut should help you a lot cutting parts out of a DEM file, putting different DEM files together or using some special effects like slow-motion or backward moving of particular scenes.

DEMcut does all this and much more in a very easy way: It gets the names of all DEM files to be processed from the command line and reads from “Standard In” commands to manipulate them. During the manipulation DEMcut writes one or more DEM files.

A good knowledge of DEM files is required to use DEMcut. DEMcut doesn’t do any checks to prevent you from creating non-working files.

4.2. Batch usage

The easiest way to use DEMcut is with a cutting list batch file. This file should contain all commands to be processed and can even have some comments why you did something like you did it. Use any text editor for this and start DEMcut with an input redirection.

```
demcut demfile1.dem demfile2.dem < commands.txt
```

DEMcut starts, reads in all DEM files from the command line and reads the commands from the command file. All messages (every command produces a message after completion) from DEMcut go to the screen or may be redirected to a logging file as well.

```
demcut demfile1.dem demfile2.dem < commands.txt > output.log
```

4.3. Interactive usage

One of the best feature of DEMcut is the possibility to do an interactive analysis of DEM files. The interactive mode is not compiled in on systems without support for named pipes. It runs with Unix and Linux but not on MS-DOS, Win32 etc.

I will describe now the Linux use. First you have to create a named pipe in the directory where your Quake copy reads its DEM files from.

```
mkfifo fifo.dem
```

Now start Quake (make sure to use `xquake` and and not `xf86quake`) in a window as usual and playdemo this named pipe.

```
playdemo fifo
```

Quake tries now to read a demo out of `fifo.dem` but there is nothing to read in: it stops.

Now comes the appearance of DEMcut. Open a new window and start DEMcut with the DEM file of your interest.

```
demcut mydem.dem
```

DEMcut reads in `mydem.dem`, does some preprocessing and waits for input. The first thing to do is to direct all output to the named pipe.

```
o fifo.dem
```

Now create the start of a normal DEM file with CD track and level initialisation.

```
c -1\n
r 0 2
```

It's unbelievable but Quake starts now the level and waits for further actions. Now play around with different ranges, speed tests or backward movements. Don't forget to put the disconnect frame at the end. Quake will usually play immediately what you intend but the Quake client time stands still after playing, so all the active sound patches will be repeated over and over and the Quake doesn't redraw its window.

This is the point where the "interactive mode" comes in: DEMcut changes its behaviour with the command

```
i 1
```

This repeats the last frame sent to Quake but with an ongoing time. The interactive mode allows you to type in new commands during a range command.

Now you can fast forward wind and rewind to find out all the frame numbers of interest. In another window you can now edit your cutting list batch file and run this file for cutting the final product.

4.4. Commands

There are not too many commands but they are very powerful. The command language is line oriented: one line means one command. Comments start with '#', empty lines or lines with wrong commands will be ignored as well. DEMcut prints a short response after finishing a command to `stdout`.

4.4.1. Help

Syntax:

```
h
```

Description:

DEMcut prints a *very* short online help. If '`i`' (interactive mode) isn't listed, your operating system doesn't support named pipes and the interactive mode is left out.

4.4.2. Output file

Syntax:

```
o filename
```

Description:

Writes further output to this file. The old output file (if there was one) will be closed. If the output file can't be written (permission denied), DEMcut bombs back to the OS shell without any warning. I may do a check in the future.

Query syntax:

```
o
```

Description:

Prints the current output file name.

Default value:

```
demcut.dem
```

4.4.3. DEM file

Syntax:

```
d filename
```

Description

Selects this DEM file as source.

Query syntax:

```
d
```

Description:

Prints the current input DEM file name.

Value range:

Any DEM file name. DEMcut generates an error, if the file is not a good DEM file.

Default value:

First DEM file from the command line. If there was no valid DEM file on the command line the d query command generates an error.

4.4.4. Speed

Syntax:

```
s speed
```

Description:

Speeds up all further output by this factor. `speed`>1 increases the play-back speed, `speed`<1 slows the play-back down.

Value range:

`speed` may be any positive float value.

Query syntax:

```
s
```

Description:

Prints the current speed.

Default value:

1.0

4.4.5. Time

Syntax:

```
t time
```

Description:

Sets the current time.

Value range:

`time` may be any positive float value.

Query syntax:

```
t
```

Description:

Prints the current time.

Default value:

0.0

4.4.6. Loop

Syntax:

```
l loop-count
```

Description:

Sets the loop count. Every further frame will be written `loop-count` times to the output. The time goes forward as usual.

Value range:

`loop-count` may be any positive integer value.

Query syntax:

```
l
```

Description:

Prints the current loop count.

Default value:

1

4.4.7. CD track

Syntax:

```
c track
```

Description:

Write the CD track string to the current output. If the CD track string doesn't end with whitespace ('\t', '\r', '\n', '\f'), an '\n' will be silently appended. Don't forget this command at the beginning of every DEM file (like it happened with finesc5.dem of the Eschaton movie).

Query syntax

```
c
```

Description:

Prints the current CD track string.

Default value:

-1\n

4.4.8. Frame

Syntax:

```
f frame-number
```

Description:

Writes the selected frame to the output. This frame gets the current time. The time will be increased afterwards by the frame duration divided by the current speed. The whole process will be repeated as much as the current loop count states.

The frame command stops in interactive mode a running range command.

Value range:

`frame-number` can be any integer number between 0 and the maximum frame number in the currently selected DEM input file.

Query syntax:

```
f
```

Description:

Prints the current frame number.

Default value:

0

4.4.9. Range

Syntax:

```
r range-start range-end
```

Description:

Writes the selected range of frames to the output. Every frame gets the current time. The time will be increased after each frame by the frame duration divided by the current speed. Every frame will be repeated as much as the current loop count states. In interactive mode you can use any commands during a running range command (i.e. speed change). The range command can be stopped in interactive mode by a new range command or a frame command.

Value Range:

`range-start` and `range-end` may be any integer number between 0 and the maximum frame number in the currently selected DEM input file. `range-end` may even be smaller than `range-start`. If they are equal the result is the same as with the frame command.

Query syntax:

```
r
```

Description:

Prints the current range.

Default value:

0, last frame number of the first DEM file from the command line.

4.4.10. Multi-player preparing

Syntax:

```
m multi-start multi-end
```

Description:

Grabs out of the defined frame range all *updatename* and *updatecolors* messages and writes all these messages together with the camera angles of the last written frame to the output. It doesn't write anything if there was no *updatename* or *updatecolors* message in the defined frame range.

Everyone, who ever tried to concatenate different multi-player DEM files knows the problem: The "actors" log in to the server in a different order and so they get every time different player numbers. If one tries to assemble all these files together one will end up with the wrong colour scheme because all players stick to the colour, number and name from the very first recording. The only solution is to insert the right name and colour definition of each part in the final product. And this does the "`m`" command.

Value Range:

`multi-start` and `multi-end` may be any integer number between 0 and the maximum frame number in the currently selected DEM input file. `multi-end` may even be smaller than `multi-start`.

Query syntax:

```
m
```

Description:

Prints the current multi-player range.

Default value:

0, last frame number of the first DEM file from the command line.

4.4.11. Interactive mode

Syntax:

```
i inter
```

Description:

Toggles the interactive mode. The frame and the range command will be affected by the interactive mode. This command is not implemented on all systems. Use the help command (‘h’) to find it out.

DEMcut continues to write (under certain conditions) the last written frame with an ongoing time. This feature is useful for an interactive use only (as the command name suggests). DEMcut remembers the (real) time, when the last frame written to Quake is finished. If up to this time came no new command (this is the normal case in interactive use) and the interactive mode is on, DEMcut writes the last frame again. But the time stamp of this frame goes on and on. So Quake gets always new frames and doesn’t stop. This is extremely important for users with many overlapping windows (like me): Quake refreshes its window in an normal working state only. It is very interesting to see what happens in the Quake window, if it gets always the same frame but the time goes on. This shows us the different sources of movement: particles (blood traces) are totally moved by the client (they continue to move), gibs are frozen in the air (they are real entities) and the weapon moves up and down as usual (client driven). DEMcut does not allow to repeat frames which change the `signon` number (frame 0, 1 or 2 of a normal DEM file). Neither does it repeat the *disconnect* frame.

The range command may be interrupted by other commands. You need this for looking at the current frame number (frame command without new frame number), for changing the current playback-speed etc. A new frame or range output command stops the running range command.

Value Range:

0 (interactive mode off) or 1 (interactive mode on).

Query syntax

```
i
```

Description:

Prints the current interactive mode.

Default value:

0

4.4.12. Quit

Syntax:

```
q
```

Description:

Closes the output file and quits DEMcut.

4.5. Example

A very easy task is to create a DEM file, which shows with double speed the reversed actions of the well-known demo “Ranger Gone Bad][: The Assault” (created by United Rangers Films).

To do this start DEMcut with

```
demcut rgb2.dem
```

and type in (or much simpler use a text file and redirection)

```
# this is a comment, I love comments # select the output o rgb2rev.dem #
select the input (not really necessary) d rgb2.dem # speed things up s 2 #
write the CD track c -1\n # write the starting frames (level init is in 0,1
and 2) r 0 2 # write all frames in reversed order (actions are in 3..7330) r
7330 3 # write the disconnect frame (is the last one) f 7331 # quit q
```

This creates the new DEM file `rgb2rev.dem` with the desired content.

4.6. Problems, ideas and everything

The current implementation of DEMcut is full of deficiencies but anyway, it works.

- The synchronisation between DEM output and the “real” time in the interactive mode is made using polling technique. It should be `alarm(2)` controlled.
- I’m thinking about the possibility to incorporate the fake Quake server of DBS in DEMcut to remove the FIFO dependency. Interactive work could then mean: fiddling at this server and a Quake client of any architecture serves as a monitoring screen (after connect to this server at the same or any other machine).

5. DBS, The Demo Broadcasting Server

5.1. Introduction

DBS is a fake Quake server, which shows every connected client a DEM file. This server is absolute alpha code. It prints a hell of debug messages and doesn’t know the full Quake network protocol. I think the name is crap and I will change it. Please send me ideas.

The source of DBS relies on some low-level Unix network routines. I may get it to work in Win32 some day with the Cygwin API but don't expect a plain Win32 version.

5.2. Command line parameters

DBS is a command line oriented program. It uses the `getopt_long` library for the command line parsing. DBS doesn't stop for itself. Use `Control+C` to stop it.

5.2.1. Syntax description

Syntax:

```
dbs [--help|-h]
```

Description:

If you start DBS without any command line parameters or the help parameters you receive the following syntax message:

```
Demo Broadcasting Server
DBS (c) U. Girlich, 1997-98, Release 3.4.4 10/24/2009 (alpha test release)
dbs [option ...] demfile
-h, --help                  display this help and exit.
-v, --version               output version information and exit.
-p, --port                  defines an UDP port for the server.
-m, --max-player            defines the max. number of players.
```

5.2.2. Version

Syntax:

```
dbs (--version|-v)
```

Description:

DBS prints a version message and quits.

5.2.3. Port

Syntax:

```
dbs (--port|-p) PORT ...
```

Description:

DBS uses by default the UDP port 26000 for the communication. This value can be overwritten with this option.

5.2.4. Maximum number of players

Syntax:

```
dbs (--max-players|-m) MAX_PLAYERS ...
```

Description:

DBS allows by default 16 players to connect and view the DEM file. This value can be overwritten with this option. The internal limit for the number of players is 255 because the Quake Network protocol reserves only an `unsigned char` for the number of players.

5.3. What DBS does

Nowadays are so many fake clients out there but not a single fake server. A proxy is not a real server because it is a client for another server. So I decided to analyse the network protocol, learnt something about UDP programming and wrote my own server. I hope to include some parts of this server into DEMcut, a feature which lacks some GUI DEM file editors up to now.

DBS mimics the behaviour of a normal Quake server. This means you can check the server with your favourite spy tool. The next step is to log in. This works as usual (you get your private UDP port and so on) but DBS ignores all client movement packets and sends to the client only the converted DEM packets of a prepared DEM file.

A small problem are the `camera` messages in the DEM packets. They come into the DEM file at the client side and not from the server over the network. Therefore I included in every packet an additional `setangle` message with the current camera angles.

This means, that you can’t get the original DEM file by client (or proxy) recording.

I hope to stabilise the server much more and think of a selection menu at the beginning where a client may select the DEM file (“channel”) of interest. I call it DoD (Demo on Demand).

6. DM2cat, The DM2 file concatenator

DM2cat is a simple Perl script to concat multiple binary DM2 files together to a single multi-level DM2 file.

It does this by simpy removing the last 4 bytes (the *endblock*) from each DM2 file.

The main aim of DM2cat is to show a sample framework for any kind of binary DM2 file manipulation.

7. LS format

This chapter describes the ASCII text based source of a LMP or DMO file. I created this language for myself. This means neither id Software nor 3D Realms use this for game development or the like. It is simply a way to show what is in these miraculous recording files.

7.1. Introduction

The LMP Source format (LS) is a (ASCII text based) readable description of the contents of a (binary) LMP or DMO file.

LMPC produces (with `--to-txt`) a well formatted LS file. So you can learn by example. I hope, that the text parser in LMPC (with `--to-bin`) can parse most LMP source. It can’t parse DMO source files.

- The LS language is a line oriented language.
- Empty lines are ignored.
- The keywords aren’t case sensitive.
- The characters between `#` and “new line” will be ignored.
- “new line” can be 0x0A or 0x0D0A.

At the very beginning of a LS file (produced by `lmpc --to-txt`) there are some comment lines with derived information:

LMP

```
DOOM (][) LMP file:  filename of the LMP file
Number of players:   derived from the number of used colours
Total play time:     derived from the number of game tics
Number of game tics: derived from the file length
```

DMO

```
Duke Nukem 3D DMO file:  filename of the DMO file
Total play time:         derived from the number of game tics
Number of game tics:     taken from the header, it stays better in a comment
```

7.2. Header Area

The Header Area starts with the keyword `HeaderStart` and ends with the keyword `HeaderEnd`.

Between `HeaderStart` and `HeaderEnd` there are lines with general information about the LMP or DMO file:

Game:

> game type (DOOM, DOOM][, HERETIC, HEXEN, STRIFE, Duke Nukem 3D, Redneck Rampage)

Version:

> version number or version byte (new DOOM/DOOM][, STRIFE) or 0 (old DOOM/HERETIC/HEXEN), the real version for Duke Nukem 3D and Redneck Rampage

Skill:

> player skill (LMP: 1-5, DMO: 1-4)

Episode:

> episode (DOOM/HERETIC/Duke Nukem 3D: 1-4, DOOM][/HEXEN: 1, not in STRIFE), Redneck Rampage: 1 (I don't know it better yet, should be 1-4)

Map:

> starting map (DOOM/HERETIC/Duke Nukem 3D: 1-9, DOOM][: 1-32, HEXEN 1-40, STRIFE: 32-??), Redneck Rampage: 1-11

In multi-player LMP files comes now a table with the colours (0-3) of the players (0-3) (0-7 in STRIFE and HEXEN 1.1)

```
Player0:      Colour of player 0
Player1:      Colour of Player 1
```

etc.

Player:

> Number of players (does exist in DMO source only)

The following lines are special for new DOOM, DOOM][and STRIFE LMP files.

MultiRule:

> Multi-Player Playing Rule (Cooperative, DeathMatch or Altdeath)

Respawn:

> number (This line appears only if the LMP was recorded with `-respawn`. The number is the number of the command line parameter `-respawn`.)

Fast:

> number (This line appears only if the LMP was recorded with `-fast`. The number is the number of the command line parameter `-fast`.)

NoMonsters:

> number (This line appears only if the LMP was recorded with `-nomonsters`. The number is the number of the command line parameter `-nomonsters`.)

ViewOfPlayer:

> colour (The LMP is recorded from the view of the player with this colour, (0<=colour<=3))

The following lines are special for new Duke Nukem 3D and Redneck Rampage DMO files.

MultiRule:

> Multi-Player Playing Rule (DukeMatch (RedneckMatch) (with spawn), Coop, Dukematch (Redneckmatch) (without spawn))

Respawn Monsters:

> number (This line appears only if the DMO file was recorded with `/t1`.)

Respawn Items:

> number (This line appears only if the DMO file was recorded with `/t2`.)

Respawn Inventory:

> number (This line appears only if the DMO file was recorded with `/t3`.)

NoMonsters:

> number (This line appears only if the DMO file was recorded with `/m`.)

The following lines are special for Redneck Rampage DMO files.

FakePlayerAI:

> number (This line appears only if the DMO was recorded with `/a`.)

Name:

> "string" (The standard name is "LEONARD" but it can be changed with the `-name` option.)

Sometimes `lmpc --to-txt` can't distinguish between new DOOM and DOOM][LMP files: In the Game line appears "DOOM or DOOM][".

The version byte can easily be derived from the game version.

Table 1. LMP version bytes

0 :	DOOM 1.1, 1.2
101:	STRIFE 1.0, 1.1

104:	DOOM 1.4beta
105:	DOOM 1.5beta
106:	DOOM 1.6beta, DOOM 1.666, DOOM][1.666
107:	DOOM][1.7, DOOM][1.7a
108:	DOOM 1.8, DOOM][1.8
109:	(Ultimate) DOOM 1.9, DOOM][1.9
110:	published source code

In HEXEN LMP files appears the player class (0: fighter, 1: cleric, 2: mage) too:

```
Class:         single player class (0-2)
```

or in multi-player LMP files:

```
Class0:        player 0 class
Class1:        player 1 class
```

etc.

7.3. Data Area

The Data Area starts with the keyword `DataStart` and ends with the keyword `DataEnd`.

All actions of one game tic have to be in one text line. In a multi-player LMP there is one line for each player per game tic.

LMPC may append (with the `--mark-step` option) a comment with the game tic and the time passed until this game tic at the line of the first player.

7.3.1. Movements

A movement consists of a 2 letter word and an amount `n`.

Table 2. Movement commands

code	action	game	range
`GFn`	Go Forward `n`	LMP	1<=`n`<=127
`GBn`	Go Backward `n`	LMP	1<=`n`<=127
`SLn`	Strafe Left `n`	LMP	1<=`n`<=127
`SRn`	Strafe Right `n`	LMP	1<=`n`<=127
`GXn`	Go X-direction `n`	DMO	-32768<=`n`<=32767
`GYn`	Go Y-direction `n`	DMO	-32768<=`n`<=32767
`TLn`	Turn Left `n`	LMP	1<=`n`<=127
`TLn`	Turn Left `n`	DMO	1<=`n`<=32767

TRn	Turn Right n	LMP	1<=n<=127
TRn	Turn Right n	DMO	1<=n<=32767

7.3.2. Use (LMP only)

Table 3. Use commands

UT	Use Thing (press button, open door etc.)
NWn	New Weapon (1<=n<=8) (change the weapon manually)
FW	Fire Weapon
PS	Pause Start
PE	Pause End
SGn	Save Game in slot n (0<=n<=7)

You can not combine PS, PE and SG with other Use commands. But you can combine the other Use commands (for instance UT and FW at the same time).

7.3.3. Fly and Look (HERETIC and HEXEN only)

A fly and look command consists of a 2 letter word and an amount (1<=n<=15).

Duke Nukem 3D supports from 1.3D onwards also Look Up and Look Down. Here the allowed range is (1<=n<=127).

Table 4. Fly and Look commands

FUn	Fly Up n
FDn	Fly Down n
LUn	Look Up n
LDn	Look Down n
FC	Fly Centre (landing, drop)
LC	Look Centre (look forward)

7.3.4. Use Artifact (HERETIC and HEXEN only)

The Use Artifact command consist of the code `AT` and a letter (`a..j`) equivalent to the `gimme`-cheat of HERETIC. In HEXEN such a cheat code does not exist but the command remains the same:

HERETIC:

Table 5. HERETIC artifacts

`ATa`	ring of invincibility
`ATb`	shadowsphere
`ATc`	quartz flask
`ATd`	chaos device
`ATe`	tome of power
`ATf`	torch
`ATg`	time bomb of the ancients
`ATh`	morph ovum
`ATi`	wings of wrath
`ATj`	mystic urn

HEXEN:

Table 6. HEXEN artifacts

`ATa`	icon of the defender
`ATb`	quartz flask
`ATc`	mystic urn
`ATd`	clerical healing key
`ATe`	dark servant
`ATf`	torch
`ATg`	porkalator
`ATh`	wings of wrath
`ATi`	chaos device
`ATj`	flechette
`ATk`	banishment device
`ATl`	boots of speed
`ATm`	krater of might
`ATn`	dragonskin bracers
`ATo`	disc of repulsion
`UA`	use all (panic button)

7.3.5. STRIFE Artifacts

You can only the use STRIFE artifacts with the `IN` or `DR` actions.

The Use Artifact command consist of the code `A` and an artifact number. Up to now I know only 12 artifacts.

Table 7. STRIFE artifacts

code	name
`0x74`	toughness
`0x75`	accuracy
`0x76`	full health
`0x7B`	teleportor beacon
`0x7C`	metal armor
`0x7D`	leather armor
`0xA1`	med patch
`0xA2`	medical kit
`0xA3`	coin
`0xA7`	shadow armor
`0xA8`	environmental suit
`0xB7`	offering chalice

7.3.6. Jump (HEXEN and STRIFE only)

In HEXEN and STRIFE you can jump up.

Table 8. Jump command

`JU`	jump

7.3.7. STRIFE actions

There are some more STRIFE actions. Remember: no talks are stored in the LMP file.

Table 9. STRIFE actions

`LU`	Look Up

LD	Look Down
RU	Run
IN	Use Inventory
DR	Drop Inventory
JU	Jump Up
U6	Unknown Action Number 6
HE	Use Health

7.3.8. Use (DMO only)

There are 28 actions in DMO files (and 4 bits for weapon change). As far as it works I use the same codes for the same or similar thing as in LMP sources. To describe the actions I use the terms for key definitions in DUKE3D.CFG.

Table 10. DMO actions

JU	Jump
CR	Crouch
FW	Fire
AU	Aim_Up
AD	Aim_Down
RU	Run
LL	Look_Left
LR	Look_Right
LU	Look_Up
LD	Look_Down
VI	NightVision
MK	MedKit
AR	AutoRun
LC	Center_View
HW	Holster_Weapon
IL	Inventory_Left
PA	Pause
QK	Quick_Kick
AM	AimMode
HD	Holo_Duke
JP	Jetpack
GQ	GameQuit

`IR`	Inventory_Right
`TA`	TurnAround
`UT`	Open (Use Thing)
`IN`	Inventory
`EC`	Esc
`NWn`	New Weapon (1<=`n`<=10) (change the weapon manually)

7.3.9. Special Codes

Table 11. Special commands

`WT`	Wait a Tic
`*n`	Repeats the current line `n` times

8. DEM source format

The DEM format is very complicated. Therefore I don't include a documentation of the text format here. Please look in the "Unofficial DEM format description". The current version is available at http://demospecs.half-empty.de/dem.

You will understand my textual representation of the DEM format by inspecting a sample file and reading a bit about the structure in the mentioned document.

If someone is willing to write a documentation of my DEM text format please inform me and your document will be included right here. For the die-hard hacker: the lex file `udeml.l` and the yacc file `udemy.y` contain the complete language grammar documentation.

9. QWD source format

The QWD format is even harder to understand than the DEM format. Please look in the "Unofficial QWD format description". The current version is http://demospecs.half-empty.de/qwd.

You will understand my textual representation of the QWD format by inspecting a sample file and reading a bit about the structure in the mentioned document.

If someone is willing to write a documentation of my QWD text format please inform me and your document will be included right here.

10. DM2 source format

The DM2 format is a nice and compact file format. Please look in the “Unofficial DM2 format description”. The current version is http://demospecs.half-empty.de/dm2.

You will understand my textual representation of the DM2 format by inspecting a sample file and reading a bit about the structure in the mentioned document.

If someone is willing to write a documentation of my DM2 text format please inform me and your document will be included right here.

11. DM3 source format

The DM3 format is very sophisticated file format to save every thinkable bit. It was developed for internet multiplayer action at a time, when highspeed internet access was rare. There is (not yet) an “Unofficial DM3 format description” but you can easily learn it by looking through the Quake III Arena source code or the LMPC source code.

You will understand my textual representation of the DM3 format by inspecting a sample file.

If someone is willing to write a documentation of my DM3 text format please inform me and your document will be included right here.

12. Some links and further information

For further information on LMP files look in the “Unofficial LMP format description”. The current version is http://demospecs.half-empty.de/lmp.

For a huge collection of some great LMP files, look at the COMPET-N home page http://www.hszk.bme.hu/~ha211/compet-n/.

For further information on DMO files look in the “Unofficial DMO format description”. The current version is http://demospecs.half-empty.de/dmo.

For a small collection of DMO files, look at the http://www.acs.oakland.edu/~klmatero/duke3d/demos.html.

For further information on DEM files look in the “Unofficial DEM format description”. The current version is http://demospecs.half-empty.de/dem.

All info you’ll ever need on DEM files can be found in the published Quake source at ftp://ftp.idsoftware.com/idstuff/source/q1source.zip

A simple DEM file editing guide can be found at http://www.Z-Studios.com/archives/filmat11.html.

For further information on QWD files look in the “Unofficial QWD format description”. The current version is http://demospecs.half-empty.de/qwd.

All info you’ll ever need on QWD files can be found in the published QuakeWorld source at ftp://ftp.idsoftware.com/idstuff/source/q1source.zip

For further information on DM2 files look in the “Unofficial DM2 format description”. The current version is http://demospecs.half-empty.de/dm2.

DM2 file editing tutorials can be found at http://www.Z-Studios.com/tutorials/dm2/tut_demo.html.

For further information on Quake II Relay project look at http://www.planetquake.com/relay/.

13. History and future of LMPC

LMPC started in the good old DOOM days as a very small Turbo Pascal 6.0 program to show me what's in these LMP files. It is now a "fat" C application and has way too many command-line switches. DEMcut is not as widely used as LMPC but it started the DEM editing hype. DBS (I really have to change this name) gave the Keygrip authors (http://www.planetquake.com/keygrip) the starting point for their internal fake Quake server.

1.0 (8/30/1994) `lmpc10.zip` (8/31/1994 posted in alt.binaries.doom (news:alt.binaries.doom), alt.games.doom (news:alt.games.doom))

- General: first public version.

1.01 (8/31/1994) `lmpc101.arj` (9/1/1994 posted in alt.binaries.doom (news:alt.binaries.doom), alt.games.doom (news:alt.games.doom))

- LMP: bug fixed in `WT` statistics.
- Docu: LS description / LMPC program reference splitted in LS.DOC / LMPC.DOC.
- LMP: New heuristic algorithm to discover former version changes.
- Docu: History added in LMPC.DOC.
- General: Terry (terry@usis.com (mailto:terry@usis.com)) wrote also a DOOM utility (I don't know it) with the archive name `lmpc10.zip`. Therefore I changed to the ARJ archive.

2.0 (10/11/1994) `lmpc20.arj` (internal beta test release)

- LMP: the LMP compiler is ready after all!

2.01 (10/12/1994) `lmpc201.arj` (internal beta test release)

- LMP: decompiler bug fixed (`UT`/`FW` and `NW` in the same game tic).
- LMP: compiler parser improved (token length test).
- LMP: `NW8` (8 gives you the chain saw directly) added.

2.1 (10/24/1994) `lmpc21.arj` (10/25/1994 posted in alt.binaries.doom (news:alt.binaries.doom))

- General: after a error free beta test phase nothing changed, new public version.

2.20 ... 2.26 (never published)

- General: rewritten from scratch (Turbo Pascal 6.0 -> ANSI C).

2.27 (2/25/1995) `lmpc-2.27.tar.gz` (internal beta test release)

- LMP: LS format improved (new HERETIC movements).
- LMP: can handle DOOM][and HERETIC LMP files.
- LMP: `GAME`-token in the LS Header.
- LMP: version change expects now the real version byte.
- LMP: 1.5 times faster (2.27/2.1, checked with a >300K LMP).
- General: includes now an executable for MS-DOS and Linux.

2.3 (4/29/1995) `lmpc-2.3.tar.gz`/`lmpc23.zip`

- General: full source code included.
- LMP: version change expects a real version string or the version byte.
- General: zip and tar.gz release.
- General: includes now go32.exe and stub.exe too.

2.4 (8/7/1995) `lmpc-2.4.tar.gz`/`lmpc24.zip` (public release)

- General: MS-DOS binary cross compiled with go32crs.
- Docu: documentation in HTML and plain text (created with Mosaic from the HTML version).
- LMP: Ultimate DOOM with 4th episode tested (nothing changed).

2.5 (8/23/1995) `lmpc-2.5.tar.gz`/`lmpc25.zip` (internal beta test release)

- General: command line parsing now with `getopt_long` and long option names are allowed too.
- General: to compile lmpc.exe (MS-DOS binary) I included the getopt-package from the GNU C library (my copy of djgpp doesn't include getopt).
- General: `--info` can handle many files on the command line.
- LMP: `--lmp-to-ls` can decompile a specific part of the LMP file (the new `--tic` option).

- LMP: the buggy version number string function repaired.
- LMP: the new option `--game` to force the game type of the LMP file.

2.6 (8/26/1995) `lmpc-2.6.tar.gz`/`lmpc26.zip` (internal beta test release)

- LMP: `--ls-to-lmp` converts now several LS files to a LMP file.
- LMP: new option `--add-WT` to add `WT` at the end of a LMP file.

2.6.1 (8/28/1995) `lmpc-2.6.1.tar.gz`/`lmpc261.zip` (internal beta release)

- General: 3 parts of the version number.
- LMP: `--gametic FROM(,|:)TO`: colon is allowed as a separator too.
- LMP: `--ls-to-lmp` keeps track of `PS` and `PE`.
- LMP: `--remove-pause` to remove the Pause game tics.

2.7.0 (9/3/1995) `lmpc-2.7.0.tar.gz` (internal alpha release)

- LMP: `--gametic` changed to `--tic`
- LMP: total new code to convert LMP files (all in one).
- LMP: do multiple LMP file changes at the same time.
- LMP: convert LMP files with the help of a temporary file (if necessary).

2.7.1 (9/5/1995) `lmpc-2.7.1.tar.gz`/`lmpc271.zip` (internal beta release)

- LMP: small bug fixes in the new code.
- LMP: MS-DOS binary converts now all filenames to lower case.

2.7.2 (9/6/1995) `lmpc-2.7.2.tar.gz`/`lmpc272.zip` (internal beta release)

- LMP: small bug in the MS-DOS binary corrected.

2.8.0 (10/5/1995) `lmpc-2.8.0.tar.gz`/`lmpc280.zip` (public release)

- General: Nothing important changed after one month of intensive beta testing. The code is stable.

2.9.1 (10/9/1995) `lmpc-2.9.1.tar.gz`/`lmpc291.zip` (internal beta release)

- General: from now on I use a version number scheme similar to Linux: second number odd means test.
- LMP: HEXEN support (rudimentary).

2.9.2 (10/9/1995) `lmpc-2.9.2.tar.gz`/`lmpc292.zip` (internal beta release)

- LMP: correct names of all HEXEN artifacts.

2.9.3 (11/19/1995) `lmpc-2.9.3.tar.gz`/`lmpc293.zip` (internal beta release)

- LMP: new option `--clear-save` to clear the `SG` tics.
- Docu: documentation in Linux-SGML.
- LMP: small multi-player bug corrected, thanks to Rob McCartney (argon@netcom.com (mailto:argon@netcom.com)) for his bug report.

2.9.4 (12/6/1995) `lmpc-2.9.4.tar.gz`/`lmpc294.zip` (internal beta release)

- new option `--mark-step` to mark only every `STEP`th game tic
- General: MS-DOS version uses `CR/LF` instead of `LF` for LS files.

2.9.5 (2/10/1996) `lmpc-2.9.5.tar.gz`/`lmpc295.zip` (internal alpha release)

- DMO: rudimentary support for Duke Nukem 3D DMO files (alpha code!).

2.9.6 (2/12/1996) `lmpc-2.9.6.tar.gz`/`lmpc296.zip` (internal alpha release)

- DMO: reliable version of `lzw.c`/`unlzw.c` included.
- DMO: all DMO header bytes decoded.
- General: new WWW directory structure.

2.9.7 (2/13/1996) `lmpc-2.9.7.tar.gz`/`lmpc297.zip` (internal beta release)

- DMO: version 1.1 of `lzw.c`/`unlzw.c` included.
- DMO: DMO support in `-s` and `-i`.
- Docu: documentation structure changed.

2.9.8 (2/19/1996) `lmpc-2.9.8.tar.gz`/`lmpc298.zip` (internal beta release)

- DMO: Chunk header removed from DS format.
- DMO: multi-player size/tics/time corrected.
- DMO: Pause is AC21.

2.9.9 (3/6/1996) `lmpc-2.9.9.tar.gz`/`lmpc299.zip` (internal beta release)

- LMP: STRIFE support included.

2.9.10 (4/7/1996) `lmpc-2.9.10.tar.gz`/`lmpc2910.zip` (internal beta release)

- LMP: HEXEN 1.1 (8 player) support included.
- LMP: considerable changes in the `-g` code.

2.9.11 (4/14/1996) `lmpc-2.9.11.tar.gz`/`lmpc2911.zip` (internal bug fix release)

- LMP: version string scan routine improved (buggy since 2.9.9).
- LMP: game type determine routine corrected (buggy since 2.9.10).
- LMP: HEXEN version change (1.0 <-> 1.1) now possible.

2.9.12 (4/21/1996) `lmpc-2.9.12.tar.gz`/`lmpc2912.zip` (internal beta release)

- DMO: macro-block error fixed.
- LMP: STRIFE player colour names included.

2.9.13 (7/1/1996), never released (internal incomplete alpha release)

- DEM: coding started for Quake DEM files, nothing completed.

2.9.14 (7/2/1996), never released (internal incomplete alpha release)

- Quake DEM files: coding continued, something completed.

2.9.15 (7/3/1996), never released (internal incomplete alpha release)

- DEM: coding continued, structure mostly completed.
- General: new recursive Makefiles, MS-DOS compiling not supported anymore.

- General: new Linuxdoc-SGML 1.5 options.
- General: Linux binary is ELF finally.
- General: MS-DOS binary is djgpp V2.0 compatible (and runs in dosemu).

2.9.16 (7/4/1996), never released (internal incomplete alpha release)

- DEM: CD track identified.
- DEM: player movement identified.

2.9.17 (7/7/1996), `lmpc-2.9.17.tar.gz`/`lmpc2917.zip` (beta release)

- DEM: cross reference to DEM specs.
- DEM: clientdata with more information.
- DEM: entities: ModNum with reference to the model names.

2.9.18 (7/8/1996), `lmpc-2.9.18.tar.gz`/`lmpc2918.zip` (beta release)

- DEM: stupid spawnstatic error corrected.
- DEM: code beautified.

2.9.19 (7/14/1996), `lmpc-2.9.19.tar.gz`/`lmpc2919.zip` (beta release)

- DEM: many new values explained.
- DEM: strings can contain unprintable characters.

2.9.20 (7/17/1996), `lmpc-2.9.20.tar.gz`/`lmpc2920.zip` (beta release)

- DEM: some new values explained.
- DEM: code rearranged.
- DEM: corresponds (more or less) to dem-0.0.5.

2.9.21 (7/23/1996), `lmpc-2.9.21.tar.gz`/`lmpc2921.zip` (beta release)

- DEM: some new values explained.
- DEM: massive code rearrangements.

2.9.22 (8/7/1996), `lmpc-2.9.22.tar.gz`/`lmpc2922.zip` (beta release)

- DEM: almost all explained.
- DEM: text file parser is ready (lex, yacc), first check.
- DEM: -s may now decompile a DEM txt file to a DEM txt file.
- DEM: massive code rearrangements.

2.9.23 (8/14/1996), `lmpc-2.9.23.tar.gz`/`lmpc2923.zip` (beta release)

- DEM: the "to bin converter" is ready. Not very stable.
- General: all messages (not getopt) go to stderr.
- DEM: write to - means stdout.
- General: new command line options: `--to-txt` and `--to-bin`
- Docu: some terms changed.

2.9.24 (10/6/1996), `lmpc-2.9.24.tar.gz`/`lmpc2924.zip` (beta release)

- DEM: all float output (time inclusive) with g-format (9 digits).
- DEM: more internal errors corrected.

2.9.25 (10/13/1996), `lmpc-2.9.25.tar.gz`/`lmpc2925.zip` (beta release)

- DEM: memory problem solved: LMPC needs now only 400-500K RAM.
- Docu: version table reorganised, README, INSTALL slightly changed.
- Makefiles: new targets, easier to make packages and my web-page.
- DEMA: first working version.

2.9.26 (11/17/1996), `lmpc-2.9.26.tar.gz`/`lmpc2926.zip` (bug fix release)

- DEM: MS-DOS binary can read in DEM text files.
- DEM: `attack_state`->`effects`, `ang_ofs_1`->`punchangle_x`.
- LMP: MS-DOS binary prints now the correct statistics.
- General: new E-mail address.

2.9.27 (11/20/1996), `lmpc-2.9.27.tar.gz`/`lmpc2927.zip` (public beta release)

- General: many thanks to Eric Gordon Fowler (efowler@jove.acs.unt.edu (mailto:efowler@jove.acs.unt.edu)) and Paolo Perrotta (nan0744@iperbole.bologna.it (mailto:nan0744@iperbole.bologna.it)) for all their bug reports.
- DEM: string in and out handles special characters better (not perfect).
- DEM: text parser allows digits after an octal escape sequence.
- DEM: \n, \r, \\ and \" are allowed character escape sequences in DEM text files.
- DEM: binary file read is byte-order independent (unchecked).
- DEM: binary file write is byte-order independent (unchecked).
- DEM: float numbers may contain exponents.
- DEM: `updateentity` with an entity > 255 works.
- General: my code survives `-Wall -pedantic` without warnings.
- Docu: Old LMPC history a bit reformatted.

2.9.28 (11/30/1996), `lmpc-2.9.28.tar.gz`/`lmpc2928.zip` (public beta release)

- DEM: put an `items` entry in binary DEM files even if it is 0. This modification enables an error-free decompile/compile of CAMPER3.DEM by “United Rangers Films” after all.

2.9.29 (12/15/1996), `lmpc-2.9.29.tar.gz`/`lmpc2929.zip` (public beta release)

- DEM: `--info` detects DEM text files.
- DEM: binary file read reorganised (coding just started).
- DEMcut: coding started. First working version.

2.9.30 (1/5/1997), `lmpc-2.9.30.tar.gz`/`lmpc2930.zip` (alpha test release)

- DEM: binary file read totally rewritten. I use now for binary files the same internal node tree representation as for text files. The text output part is a bit slower now but the whole project is much easier to maintain.
- DEM: `--to-bin` can handle binary DEM files too. It should create an identically copy of the input file (if I didn’t make any mistakes).
- DEM: binary write: last (I hope so) endian-dependence removed.
- DEM: binary write: suppresses “camera only” blocks.
- DEM: text write: suppresses “camera only” blocks.
- DEM: text write: stops at file write errors (e.g. device full).

- DEMcut: almost all new coded. The interface remains identical. I can now include very easy new commands.
- DEMcut: the time message can be anywhere in a block.
- DEMcut: new “m” command to prepare multi-player DEM file concatenations.
- DEMcut: new “i” command to improve the interactive work: range may be interrupted, output is time-synchronised.
- Docu: Formatted with SGML-Tools 0.99.0.
- General: Orthographical check with `ispell` (British English).

2.9.31 (1/7/1997), `lmpc-2.9.31.tar.gz`/`lmpc2931.zip` (bug fix release)

- DEM: text write: CD track was never printed: totally stupid bug.

2.9.32 (2/9/1997) (alpha test release)

- DEMcut: more info on running ranges in interactive mode.
- DEMcut: “d” command don’t need files from the command line.
- General: includes executables for Win32 (untested).
- General: new packaging system.
- General: all messages go to stderr.

2.9.33 (2/16/1997) (alpha test release)

- DMO: Redneck Rampage support started. Info works, decompression not.
- DMO: Duke Nukem 3D 1.3D DMO files supported. This incompatible version exists since April 1996 but nobody said it to me.
- DMO: endian-dependence should be removed. Never checked this.

2.9.34 (3/23/1997) (beta test release)

- DEM: `--info` prints the number of game tics (or frames) for binary DEM files.
- DEM: new option `--fix-multi` for in-place editing.
- DEM: new option `--fix-multi` as additional option for `--to-bin` and `--to-txt`.
- DEM: bin read: type check was endian-dependent (no longer).
- DEM: bin read: CD track at the beginning can be anything, that Quake itself likes. Read the DEM specs (>=1.0.4) for details. The full CD track string will be included in a DEM text file but LMPC can cope with the old format as well.

- DEMcut: CD track commands expects the '\n' character as well. The old syntax continues to work.
- Docu: Demolition and DemoStat mentioned.

3.0.0 (3/31/1997) (public beta release)

- DEM: text read: hours in time stamps didn't work (bug report by James Stephenson (james@BETAus.com (mailto:james@BETAus.com))).
- DEMcut: CD track command works now correctly.
- `--game` option repaired.
- Docu: general clean-up.

3.0.1 (5/15/1997) (public release)

- General: text write: Win32 versions make MS-DOS style line breaks (hint by David Wright (wgwright@mnsinc.com (mailto:wgwright@mnsinc.com))).
- General: Minimalist GNU Win32 (Mingw32) version ready (untested).
- Win32: The `syserror` function was always broken. Why didn't somebody give me a hint?
- DEM: bin read: 'new' (version > 1.06) DEM format.
- DEM: bin write: stupid error in *clientdata* removed. It was there since 2.9.23 (hint by Yonatan Donner (ruthd@post.tau.ac.il (mailto:ruthd@post.tau.ac.il))).
- DEM: entry clientdata->weapon may contain invalid values (more than 1 bit set).
- Packaging: Win32 versions with standard file names. Only the 'MS-DOS Binary Package' use the short 8.3 name scheme.

3.1.1 (7/20/1997) (internal alpha test release)

- QWD: bin read: coding started. Nothing completed.
- Docu: all tables now floating in TeX.
- Conf: all is now Autoconf configured.

3.1.2 (7/28/1997) (alpha test release)

- DEMcut: interactive mode is removed in systems without named pipes.
- DBS: alpha code.
- Conf: Autoconf rules newly written. Really everything is Autoconf configured.
- Docu: DBS mentioned.
- Docu: rename INSTALL in COMPILE.

- Docu: new file: Autoconf generic INSTALL.

3.1.3 (8/17/1997) (alpha test release)

- General: package renamed: LMP/DMO/DEM/QWD Control Centre -> Little Movie Processing Centre
- QWD: QuakeWorld 2.00 support included.
- Docu: back again to SGML-Tools 0.99.0.
- Docu: chapters restructured.

3.1.4 (12/29/1997) (alpha test release)

- Conf: `Makefile.in` created from `Makefile.am` by GNU Automake. Much easier to maintain!
- General: Hexen II support just started. Not complete. Can’t even decompile correctly a single file.
- Compile: some general routines in a library.
- General: Quake II support just started. Not complete. Can’t even decompile deltapacketentities.

3.1.5 (1/1/1998) (alpha test release)

- DM2: text output: many comments included.
- Docu: SGML-Tools 1.0.2 ready after all.
- Conf: Automake 1.2 compliant. Some yacc/lex changes were needed.

3.1.6 (3/12/1998) (alpha test release)

- DM2: binary read: nasty malloc/free bug removed.
- DM2: some new commands got better names.
- DEM: binary read: CD track totally rewritten.
- LMP: can handle -v1.10 (for the released DOOM source code distribution)
- Web: PlanetQuake is the new home.
- Docu: SGML-Tools 1.0.5 used.

3.1.7 (6/17/1998) (alpha test release)

- DBS: subtle sync problem solved. Thanks to Paul Baker (pbaker@idle.com (mailto:pbaker@idle.com)) for his bug report.
- DM2: text in: parser implemented.

- DM2: some command names changed.
- DM2: binary out: implementation started.

3.1.8 (7/15/1998) (intermediate beta release)

- DM2: server record files implemented.
- Conf: dist target for source package is working now.
- Conf: more checks for more platforms.
- Docu: SGML-Tools 1.0.7 used.
- QWD: binary in: up to QuakeWorld 2.29BETA.

3.1.9 (8/16/1998) (beta release)

- QWD: text in: written.
- QWD: binary out: written. Due to the ongoing demand by many users I implemented now the QWD binary write routine. This gives together with the text read routine the first really working QWD editor.
- DM2: Visual Weapon (VWep) support.
- DM2: some really bad errors corrected.
- Conf: After many fights with the Automake system, ‘make distcheck’ works finally without problems.

3.2.0 (9/6/1998) (stable release)

- DM2: binary read and binary write: long vwep corrected. Now works `massive1.dm2`.
- DM2: binary write: `timeofs` in *sound* wasn’t even implemented.
- DM2: text read: allows now empty *packetentities*. Many users reported this problem in reease 3.1.9. Thanks to all of you.
- QWD: text read: allows now empty *packetentities* like in DM2 text read.
- text read: useless terminals removed.
- DM2: unused bits in bit mask (*sound*, *spawnbaseline*, *packetentities*) detected. It’s a real mess that Keygrip 2 set unused bits in DM2 files but I implemeted an easy way to cope with it. Now works `latenite.dm2`.
- DM2: text read: allows now empty *playerinfo*. Now works `grenbfg.dm2`.
- DEM: Hexen II support (it was always only a small subset) totally removed.

3.2.1 (9/8/1998) (stable release)

- DEM: text write: didn't write the CD track. A stupid integration error. I know, I actually have to make a software quality test with all my test demos before the upload.
- DEM: to text: problems with `--game QUAKE` switch. Not really solved. Just a kludge inserted, which even slows the program down a bit.

3.2.2 (1/9/1999) (public release)

- QWD: bin read: protocol version 27 (game version 2.29BETA and up) corrected.
- QWD: read text: many small but important parsing problems removed. This was possible because Joel Baxter (jbaxter@lemur.stanford.edu (mailto:jbaxter@lemur.stanford.edu)) provided me with a problematic QWD file (recorded with game version 2.21) and many good hints.
- QWD: *updatepl* better.
- DM2: updated up to game version 3.20.
- DM2: bin read: *temp_entity* parse code directly calculated from the DM2 specs.
- Docu: SGML-Tools 1.0.9 used.
- Compile: Cygwin and Mingw32 binaries are cross-compiled with the Cygwin toolset version B20.1.
- DM2cat: The DM2 file concatenator included.

3.2.3 (1/12/1999) (public release)

- Compile: udm2-te[123].c are created directly in the source directory.
- Package: udm2-te[123].c are included in the source and full package. I forgot them in the last release. Thanks to Stefan Schwoon (schwoon@informatik.tu-muenchen.de (mailto:schwoon@informatik.tu-muenchen.de)) for mention this error.
- The script to extract the C-Source out of the specs file is included too.
- WWW: upload script uploads now the packages to the right place to deliver them to all http://www.cdrom.com mirrors. In the last release this was totally wrong.

3.2.4 (1/19/1999) (public release)

- QWD: bin read: new *serverinfo* message wrong handled. Thanks to Michale Hofele (michael@i603.hadiko.de (mailto:michael@i603.hadiko.de)) for the bug-report.

3.2.5 (4/6/1999) (public release)

- Conf: Autoconf 2.13 and Automake 1.4 used.

- DEMcut: interactive mode corrected. Thanks to Vasily Kuznetsov (VKuznetsov@microtest.ru (mailto:VKuznetsov@microtest.ru)) for the bug-report and patch.
- QWD: bin read: *setpause* corrected. Thanks to Christer Sandin (czsuch@ocag.ch (mailto:czsuch@ocag.ch)) for his bug report.
- DM2: multi-level repair corrected. Thanks to Zerox (zerox@prism.ne.mediaone.net (mailto:zerox@prism.ne.mediaone.net)) for the bug-report and a short demo file.

3.2.6 (4/30/1999) (public release)

- QWD: bin read: initialization of the precache lists for game version 2.20 and higher corrected.
- QWD: *maxspeed* and *entgravity* corrected.
- Thanks to R. Quaak (jaqua00@direct.a2000.nl (mailto:jaqua00@direct.a2000.nl)) for the *useful* bug report and a short demo file. Remember: with a good bug report and a short demo file, I can correct a bug really fast.

3.2.7 (5/24/1999) (bug fix release)

- Compile: source can be compiled with C++ compilers too. No C++ specific feature used up to now. Many new C++ related warnings corrected.
- General: severe buffer overflow problems corrected. Thanks to Nathan Whitaker (naif@3dcoder.freeserve.co.uk (mailto:naif@3dcoder.freeserve.co.uk)) for the bug report and a demo file.
- Compile: Linux binaries are glibc finally. I updated my main development system from SuSE Linux 4.2 (about 4 years old) to SuSE Linux 6.1 to do this.

3.2.8 (6/3/1999) (bug fix release)

- QWD: *modellist* and *soundlist* corrected. Thanks to Hoffy (ripple@powerup.com.au (mailto:ripple@powerup.com.au)) for the bug report and a demo file.

3.2.9 (8/18/1999) (public release)

- DM2: multi-level repair method improved. Thanks to Mike Dowell (mdowell@Mike-D.com (mailto:mdowell@Mike-D.com)) for the bug report and a demo file.
- DM3: some code fragments prepared but nothing compiled in.
- Compile: MS-DOS binaries cross-compiled with egcs-2.91.66.

3.3.0 (1/23/2000) (public release)

- DM2: RLA (Quake II Relay) support.

- Docu: Some links updated.

3.4.0 (8/29/2004) (alpha test release)

- Compile: Compiled everything on a current platform.
- Docu: Used new tools to generate the documentation.

3.4.1 (3/4/2006) (alpha test release)

- DM3: Support added.
- Docu: DM3 mentioned.
- Compile: Compiled everything on SuSE Linux 10.0.

3.4.2 (8/5/2006) (alpha release)

- DMO: Duke Nukem 3D 1.4 and 1.5 support added.
- Compile: Compiled everything on SuSE Linux 10.1.

3.4.3 (4/21/2007) (alpha release)

- DMO: Duke Nukem 3D: game tic length corrected.
- Compile: Compiled everything on SuSE Linux 10.2.

3.4.4 (10/24/2009) (alpha release)

- WWW: Move everything from PlanetQuake to Half-Empty.
- Compile: Compiled everything on openSUSE Linux 11.1.

Plan for 4.0.0 in order of importance

1. Block-oriented rewrite of the text parse routines.
2. Object-oriented demo handling. This is really necesary for further recording formats.
3. More code generation out of the XML specs source.
4. Strict separation of the different recording types and the main program. Maybe something like `librecord.so`.
5. `--split-multi` option to split multi-level recordings.
6. Half-Life (DEM), Heretic II (DM2) and Sin (DM2) support.

7. DEMcut and DBS with QWD and DM2 support.
8. File-IO from `stdin` and to `stdout`, really difficult is the file type detection.
9. Redneck Rampage decompression - might be solved with the enhanced Duke Nukem 3D support but I can't run Redneck Rampage any longer to check it out.

Ideas finally dropped out:

1. Hexen II and HexenWorld support.

A new regular version of LMPC (like 3.5.0) will be announced on BluesNews (http://www.bluesnews.com) and on PlanetQuake (http://www.planetquake.com).

You can download LMPC from my Demo Specs page (http://demospecs.half-empty.de).

There you will find always the latest release with new features and brand new bugs.