

# Mupen64 user guide

Hacktarux - Malcolm

July 3, 2004

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Technical Overview</b>	<b>3</b>
2.1	Pure interpreter . . . . .	3
2.2	Interpreter . . . . .	3
2.3	Dynamic recompiler . . . . .	3
<b>3</b>	<b>Windows</b>	<b>4</b>
3.1	Requirements . . . . .	4
3.2	Compatibility . . . . .	4
3.3	GUI . . . . .	4
3.3.1	Shortcut keys . . . . .	5
3.3.2	Command Line Support . . . . .	6
3.3.3	Other features . . . . .	6
3.3.4	Configuration . . . . .	7
3.4	Compiling (Source) . . . . .	9
<b>4</b>	<b>Linux</b>	<b>9</b>
4.1	Requirements . . . . .	9
4.2	Compiling (Source) . . . . .	10
4.3	Compatibility . . . . .	10
4.4	GUI . . . . .	10
4.5	Plugins . . . . .	10
<b>5</b>	<b>Thanks</b>	<b>11</b>
<b>6</b>	<b>Contact Information</b>	<b>11</b>

## 1 Introduction

Mupen64 is a n64 emulator desgined to be multi-OS. It has been developped on/for Linux originally but the emulator has already been ported succesfully on Windows and MacOSX for example. Actually the program can be easily ported on all OS supported by the SDL library.

In its current state, the emulator is highly compatible and use a plugin system. With the correct plugins (“correct” can be computer dependent), it can achieve nearly perfect graphics and sound in many games.

The biggest problem of this emulator until now was its speed but now it’s progressively changing with a totally new core and there’s still a huge room for improvements. Anyway if you have a fast machine, you should already be able to get many games running fullspeed.

## 2 Technical Overview

When you configure Mupen64, you’ll have to choose between three CPU emulation cores. Here we’ll introduce the technology used by those three cores so that you can understand what you are doing.

### 2.1 Pure interpreter

In the pure interpreter core, opcodes have to be decoded each time they are executed. So if a loop is executed one million times, each opcode of the loop has to be be decoded one million times. Needless to say it’s a slow core, but it has theoretically less bugs than the others because it’s simpler to program.

### 2.2 Interpreter

In interpreter core, the opcodes are put in a cache after decoding them. When programming such core, the biggest difficulty is to detect when the cache become invalid when for example the game doesn’t need some code anymore and erases it with some new code. That’s why it can have some compatibility issues, but in its current state i’m considering the compatibility is nearly the same. Of course, this core is far much faster than the pure interpreter core and we are encouraging people to use this core by default with this release.

### 2.3 Dynamic recompiler

The mupen64’s dynamic recompiler is an extension of the interpreter core. Not only it decodes the opcodes and caches the result, but it produces native assembly code like a compiler would do from the source code except we don’t have source code here but a list of opcode. This is the fastest core we have. Compared to the intepreter core, it has some additional compatibility issues but generally it’s almost as compatible. Two new options (no audio delay and no

compiled jump) can fix some of your compatibility problems with this core, try them with your games and see what's working better for you.

## 3 Windows

Here we'll tell you how to configure mupen64 to run on windows.

### 3.1 Requirements

To get yourself up and running you're going to need at least a 600MHz processor based computer, with 64mb RAM, a video card that a N64 graphics plug-in supports (OpenGL<sup>1</sup>, D3D<sup>2</sup> or GLIDE<sup>3</sup>) and a keyboard/mouse/joystick depending on what you'd like to use for input.

Glide is bringing up the rear with Dave's Glide64<sup>4</sup> plugin if you have a voodoo card, but because more advanced cards (from NVidia<sup>5</sup> and ATI<sup>6</sup>) have optimizations for both OpenGL and D3D they are more commonly used and at the time of this document provide better results.

For the D3D plugins your going to need to upgrade your Direct X versions to at least 7 depending on the choosen plugin, but the newest Direct X is always suggested.

For OpenGL most Windows users should be fine as long as they have the latest drivers for their video card, NVidia GPU (Graphic Processing Unit) based cards (TNT, GeForce) should go and install the NVidia detonator drivers, they generally are faster.

Oh and of course Mupen64 will only run on versions of Windows 98 and up, but will probably not run on NT4.

### 3.2 Compatibility

We didn't include any compatibility list in this version but we can say you that many games are compatible, it would actually be easier to list games that aren't compatible. Thus said there're still a lot of roms that are not working or have problems. Feel free to report them on the message board. You should also know that many problems can come from the plugins you are using.

### 3.3 GUI

The Windows GUI<sup>7</sup> has been refined starting from 0.0.90. We now run the ROM inside the Mupen64 window, oppose to opening another dialog window.

---

<sup>1</sup><http://www.opengl.org>

<sup>2</sup><http://www.microsoft.com/directx>

<sup>3</sup><http://www.voodoofiles.com/>

<sup>4</sup><http://glide64.emuxhaven.net/>

<sup>5</sup><http://www.nvidia.com>

<sup>6</sup><http://www.ati.com>

<sup>7</sup>Graphical User Interface

Full screen now supported. There is also a toolbar, buttons done with Windows XP type style, and the configuration of plugins has been redone to its own dialog. Take some time and explore, nothing is hidden. There was many improvements in 0.1 release. Firstly there's an ini file which can be edited using the gui. You can access Rom Properties dialog, and from there add your own comments to the roms,also you have an option to calculate MD5 checksum of the file and emulator will automatically identify this rom,and will change it's name in Rom Browser if needed.

The emulator is now able to identify the dump version you're trying to play, if it's not a known good dump, the emulator will warn you. There is also a cache which enable you to keep your roms on slow media such as CDROM without having to wait for a long time before being able to use the emulator. The name displayed in the rom browser is the good name as the goodn64 tool is giving it. There's a few possibilities that mupen64 don't identify bad dumps during the first scan, but it will identify them the first time you'll play them. Finally an audit option will let mupen64 scan all roms to find the name of all the dumps you have just as goodn64 would do.

If you have any suggestion on the GUI please post them on emutalk where the mupen64's message board is hosted : "<http://www.emutalk.net/>".

### 3.3.1 Shortcut keys

Alt+Enter or ESC	Switch from Full Screen/Window mode (can also be done using toolbar)
F1	Reset Rom
F2	Pause Emu
F3	Make screenshot (plugin dependant)
F5	Quick Save
F7	Quick Load
,1..9	Choose Save slots you have up to 10 save slots for each game
CTRL+A	Save as
CTRL+L	Load previously saved file. Note: you can only load if the same game is running.
CTRL+S	Open settings dialog.
CTRL+P	Open rom properties dialog.
CTRL+R	Refresh rom browser.
ALT+T	Switch on/off toolbar
ALT+S	Switch on/off statusbar
+	Increase speed modifier
-	Decrease speed modifier
.	Put speed modifier to 100%
TAB	while pressed emulator will run without speed limit
...	

You can use make a quick save state by hitting F5 key. To load it again

press F7. There are 10 slots to save states, you can switch between them using number keys.

### 3.3.2 Command Line Support

Mupen supports command line arguments. Here are the options :

```
mupen64.exe -a <audio plugin dll name>      ,ex. -a audiobeta.dll
             -v <video plugin dll name>      ,ex. -v Jabod3d6.dll
             -c <control plugin dll name>    ,ex. -c JaboInput.dll
             -rsp <rsp plugin dll name>      ,ex. -rsp rsp.dll
             -g <full path to the rom>       ,ex. -g c:\roms\mario64.zip
             -f                               starts rom in full screen
             -nogui                          enters nogui mode, usefull to
             write a front-end
             -save                           saves plugin and other params
             on exit from emulator
```

If you have spaces in one of the parameter, you can use " ", for example :

```
mupen64.exe -g "c:\roms\mario 64 [u] [!].zip" -nogui -f -a audiobeta.dll
```

will start emulator in nogui mode, load audiobeta.dll as sound plugin, and will take current default plugins for gfx, control and rsp, and will go fullscreen after loading rom. In -nogui mode you can end emulation by pressing ESC key. All usual hotkeys not available.

### 3.3.3 Other features

:

**Configuration Dialog :** In this dialog you can access almost all available options and settings for mupen. Configure plugins, sets directories, change emulation options and so on.

**Recent roms menu :** Store all recent roms, that you play, have option to freeze recent roms, and not adding new, and to clear list. You can start a game by clicking on according menu item.

**Rom Properties Dialog :** Here you can see most rom header information, calculate md5 of the rom and identify it. Also you can pick what plugins you want to use with this rom. Those settings saves in the romsetting.ini file. When starting this rom next time, emulator will look in this file and if found will automatically change plugins.

**Speed modifier :** Usefull when you want to run rom at any speed you like from 1 to 200%. Hotkeys are + for increase speed, - for decrease and . for default 100%.

**Rom Browser :** Rom Browser allow you to view all your available roms,its supporting multiple directories. Right click on any rom in rombrowser opens menu with option to play game,rom properties, and refresh .You can select what columns to show in rom browser in the configuration dialog. Rom browser uses cache for storing all roms info,including crc,md5,name,country and so on, when refreshing rom browser it clears all of it's cash and make rescane of all roms. Be carefull with this option,as it removes all previos md5 info as well.

**Audit roms dialog :** Allows you to audit your roms,ie check them.Mupen stores md5 info of all known good,hacked or bad roms available in mupen64.ini .Using this file mupen can with 100% precision identify your rom, this utility can be runned on all roms,that are inside of rombrowser cache and calculate their md5 checksum.You can stop scan process any time and return to it later,previously scanned roms will be skipped.

**Multy Languages support :** Mupen64 have Multi languages support.Main language file is mupen64.lng , when mupen starts it reads from this file what languages to put in the Languages menu.All languages files is inside /Lang directory by default.If you want to make your own translation or fix errors in existing one, dont hesitate to contact us ,new translations always welcomed :)

**Video recording :** With mupen64 0.4, a new feature has been introduced that let you record videos of your games. This feature is split in two parts. The first part record all the keys you press while you play the game, this produces a .rec file. You can watch it later using mupen64. The second part of feature will convert the .rec file into a .avi file compressed with the video codec of your choice. I haven't found yet how to compress correctly audio, so in the current version, audio will be ADPCM, but it may change later.

Here the list of files mupen64 windows port creates in it's directory and description of each of them : mupen64.exe - main executable mupen64.ini - main ini file,used for storing md5,name,crc and saves types info. mupen64.cfg - main gui configuration file,plugins and different options inside mupen64.cache - cache file with rombrowser information mupen64.cch - rom browser directories inside mupen64.lng - index file for languages whatsnew.txt - log file with changes in all versions /Lang - directory with language files /Plugin - default directory for plugins 9can be changed in config dialog) /Screenshots - default directory for screenshots /Save - default directory for save states ...

### 3.3.4 Configuration

Before loading roms make sure, that you configured properly emulator plugins.

- **Graphics :** For best graphics results use Jabo 1.4/1.5, Rice's unofficial Daedalus or icepir8's TR64 opengl graphics plugins for example. All plugins can be found on emutalk download section except jabo's plugin that can only be found inside PJ64 official package.

- **RSP** : The RSP is a chip inside the n64 used to produce visual and sound effects. At the time of this document, only two RSP plugins exists. The first one is the Jabo's RSP plugin included in PJ64 official package. It's a pretty good lle<sup>8</sup> RSP plugin very well optimized.

The second plugin is included with mupen64 and is emulating the RSP using hle<sup>9</sup> technic. It means it's very fast but has some compatibility problems. If you have a slow machine, this plugin can help you achieving better frame rate. Currently this plugin supports the three main audio ucodes<sup>10</sup> as well as jpeg ucodes in zelda and rare's mp3 ucode. It also implements some boot codes for games such as banjo tooie or donkey kong. If the game is using another ucode, it'll have some problems that may make it unplayable, in that case you will have to use jabo's plugin. The audio part of the hle rsp plugin is based on uhle source code which is very fast but has some inaccuracies. Again, if you get bad sound, try jabo's rsp plugin to see if the problem is coming from the rsp plugin or not.

Both of these plugins can process the audio ucodes (also called audio lists) or send them to the audio plugin. Only a few sound plugins are able to process the audio lists, they are called "hle sound plugins" The only existing hle sound plugins are the various Azimer's sound plugins and the ultrahle2064 audio plugin provided with ultrahle2064 emulator. Azimer's plugins are more accurate than uhle sound code used in the rsp hle plugin and are still nearly as fast but some people are having problem with the sound part of the plugin (that is different from the hle part that is processing sound before sending it to the sound card). For this reason i have also included an option in the hle rsp plugin that let you send the audio lists to a specific audio plugin without using it to send the sound to your sound card (the audio plugin specified in mupen64 configuration will still be used for that purpose).

- **Sound** : audio plugins are fully supported in this version. There's not much to say about audio plugins if you have read carefully the RSP explanations. If you have choosen to process the audio lists in the RSP plugin or in a specific audio plugin, you may be wondering if some lle plugins exists. If you have understood what has been said in the previous section, all hle audio plugins can be used as lle plugins (only half of it will be used but that's all). Some 100% lle sound plugins exists such as jabo's sound plugin or zilmar's basic plugin.
- **Input** : our preference for input plugins goes to nrage input plugin as it seems to have most features than all the other ones.
- **Core** : To choose the core read the "Technical Overview" section.

---

<sup>8</sup>low level emulation

<sup>9</sup>high level emulation: technic that simulate the software run on a chip instead of emulating every single low level aspects of it

<sup>10</sup>a ucode is a program run on the RSP



### 3.4 Compiling (Source)

Because the project has been open source for the past few versions I will give some instruction to getting what software you need to begin developing Mupen64. Before I begin I must explain a few matters.

We give you the source for Mupen64 for two major reasons, for your own reference (or help) and to aid us in the project. We don't want people taking the source, renaming the dialog, changing the GUI around and releasing it. Please honor the fact that we spend a lot of time toward the project and releasing a renamed Mupen64 is dishonest and detours the authors from ever releasing the emulator again. We trust you with the source; we trust you will use it honorably.

To download the source go to the Mupen64<sup>11</sup> downloads page and grab a source archive. After grabbing the source you're going to need to download Dev C++<sup>12</sup>. Dev C++ is the application of choice for the Mupen64 programming team. It is a free open-source programming interface.

Good luck, and if you have any improvements on the source that you think we should know about please go to EmuTalk<sup>13</sup> message boards and post the source in the Mupen64 area<sup>14</sup>. We will give you full credit if the source is used in Mupen64's next release.

## 4 Linux

The linux version is very similar to the windows one, so i suggest you to read the windows part of this guide first especially the plugins section to understand how things are working.

### 4.1 Requirements

Before downloading Mupen64 make sure that your video card is compatible with OpenGL (Mesa). The current N64 Linux plugins are only OpenGL compatible. At present NVidia based cards are the best ones to use under Linux for OpenGL, ATI is in second, and the rest are... the rest ;-). Make sure you have Mesa 4 or higher installed, and the Mesa GL, GLU and GLUT packages installed. Now your also going to need to have XFree86 4 installed. If your running off a NVidia based card you better mozzie over to the NVidia site and grab the Kernel and GLX updates, they will be essential for your card. Because I don't own an ATI based card I have not used any ATI Linux drivers, but I have found some information on ATI Linux drivers, which can be found here. Always try to use the drivers for your video card, the default Mesa drivers are pretty slow. If your card is OpenGL compatible and isn't NVidia or ATI go to RPMFind and search for your GPU, you'll probably find something. Once those are in place you should be up and running.

---

<sup>11</sup><http://mupen64.emulation64.com/down.htm>

<sup>12</sup><http://www.bloodshed.net/dev/devcpp.html>

<sup>13</sup><http://www.emutalk.net/>

<sup>14</sup><http://www.emutalk.net/forumdisplay.php?s=&forumid=50>

The Linux version of Mupen64 comes in a compressed tar/gzip archive. Once you download the file simply use Archiver, or in a Terminal type:

```
# tar -zxf mupen64{version}.tgz
```

After this you should have a nice Mupen64 directory formed in the same directory as the compressed archive.

Now you have a Mupen64 binary. To launch simply type `./mupen64` in the Mupen64 directory, you should now be in the GUI.

## 4.2 Compiling (Source)

Before compiling Mupen64 make sure you have gcc installed. To compile mupen64 you just have to decompress source code archive and to run make command.

```
# make
```

Maybe you'll have to modify a few things in the Makefile but if you are a developer i don't have to explain you all these things so that's all for this section ;)

## 4.3 Compatibility

The compatibility in Mupen64 is the exact same as the Windows version, the only thing holding the Linux version back is the graphics plugin.

## 4.4 GUI

Thanks to Blight, the linux gui is almost the same as the windows one. So look at the windows section of this document ;-)

## 4.5 Plugins

- **Graphics :** At the time of this document, three plugins are available. Firstly there's the mupen64's gfx plugin. It's very basic and should only be used for 2d demos. The second graphic plugin is TR64 plugin. It's very compatible and should work on most video cards that supports OpenGL 1.2. But there's a but... this plugin can be hard to use as you have to choose the ucode each time you run a game and there are graphic glitches sometimes, mainly caused by the fact that icepir8 only use OpenGL 1.2 without any extension. The third plugin is Glnintendo64, it has lower compatibility than TR64 and needs a video card supporting at least OpenGL 1.4 or nVidia extensions but it has nearly perfect quality and is easier to use.

- **RSP** : Only one plugin exists for linux, it's the hle rsp plugin and it's bundled with mupen64. Currently, there are no third party hle sound plugin so we don't have implemented a config dialog box like in the windows version.
- **Sound** : Mupen64's sound plugin is using the OSS library and should give you nice results in most cases.
- **Input** : The best linux input plugin is the latest blight's input plugin. It allows you to use keyboard or any joystick supported by the SDL. Another plugin is provided with mupen64 (mupen64's input plugin) but it's very basic and can't be configured, it can be of some help if you have some difficulties with blight's plugin.

## 5 Thanks

Special thanks to Icepir8 for his gfx plugin, and to NRage and Obsidian for their input plugins.

All the authors of an open source n64 emulator, they all have helped me accidentally at some point: Zilmar, Jabo, Niki Waibel, Schibo, Rice, Dave2001, StrmnNrmn and sorry for the ones i have forgotten ! Thanks also to Azimer for giving me some info on how he made his audio plugin. Thanks to Martin and the emu64 team for hosting mupen64 website and to DOUG and Mesmann for the time they spent on the web site design.

I would also like to thanks all the beta testers (you know who you are guys !).

## 6 Contact Information

Firstly, to get the latest news about mupen64, you should visit our web site at <http://mupen64.emulation64.com>.

For main support please go to EmuTalk.net, there are many Emulator Experts there who can solve the bulk of any problems. We also run a IRC channel, #Mupen64 on EfNet. If you would like to contact us directly here are our email addresses below.

I'd like everyone to note that we will not even respond to ROM requests. Also that using Mupen64 or any other emulator is a privladge, not a right. Telling us to release the next version or to do anything else isn't very respectful. We understand that you want all your games to work, and work perfectly. It is a goal, but one that will not be here any time soon. Please have patience.

Hacktarux: <a href="mailto:hacktarux_at_yahoo.fr">hacktarux_at_yahoo.fr</a> Blight	Main Coder linux GUI coder, input plugin coder and gfx plugin porter debugger coder
DavFR ShadowPrince	windows gui coder, web site programming. . .
Malcolm: <a href="mailto:Malcolm_at_emulation64.com">Malcolm_at_emulation64.com</a>	Original creator of this help file, GUI contributor, General Emulator specialist, Support dude ;-)
CoDeX, Pir4nhaX 'Flea Muad Claus Windeler	GUI Contributors MacOSX port QNX port BeOS port