

ScummVM ReadMe

The ScummVM Team

2005/03/26

Contents

1	About	2
2	Contacting	2
2.1	Reporting Bugs	2
3	Supported Games	2
3.1	Copy Protection	4
3.2	Simon the Sorcerer 1 and 2 notes	4
3.3	Broken Sword notes	4
3.4	Known Problems in ScummVM 0.7.1	4
4	Supported Platforms	5
5	Running ScummVM	5
5.1	Command Line Options	6
5.2	Language options	7
5.3	Graphics filters	8
5.4	Hot Keys	9
5.5	Using data files from Macintosh game versions	10
5.6	Multi-CD Games	10
5.6.1	The Curse of Monkey Island	10
5.6.2	Broken Sword 1	11
5.6.3	Broken Sword 2	11
6	Savegames	11
6.1	Autosaves	11

7	Music and Sound	11
7.1	Playing sound with Adlib emulation	12
7.2	Playing sound with MT-32 emulation	12
7.3	Playing sound with MIDI emulation	12
7.4	Playing sound with Native MIDI	12
7.5	Playing sound with Sequencer MIDI	12
7.5.1	Playing sound with ALSA sequencer	13
7.6	Using compressed audiofiles (MP3, Ogg Vorbis, Flac)	14
7.6.1	Using MP3 files for CD audio	14
7.6.2	Using Ogg Vorbis files for CD audio	14
7.6.3	Using Flac files for CD audio	14
7.6.4	Compressing MONSTER.SOU with MP3	14
7.6.5	Compressing MONSTER.SOU with Ogg Vorbis	14
7.6.6	Compressing MONSTER.SOU with Flac	15
7.6.7	Compressing sfx/speech in Simon the Sorcerer 1 and 2	15
7.6.8	Compressing speech/music in Broken Sword 1	15
7.6.9	Compressing speech/music in Broken Sword 2	16
7.7	Output sample rate	16
8	Configurations Files	16
9	Compiling	19
9.1	GCC	19
9.2	Microsoft Visual C++ 6.0	19
9.3	Microsoft Visual C++ 7.0	19
9.4	Windows Mobile with Microsoft eMbedded Visual C++ 3 or 4	20
9.5	Debian GNU/Linux	20
9.6	Mac OS X	20
10	Credits	21

1 About

ScummVM is a collection of interpreters, capable of emulating several adventure game engines. ScummVM mainly supports engines created using SCUMM (Script Creation Utility for Maniac Mansion), used in various LucasArts games such as Monkey Island, Day of the Tentacle, and others.

ScummVM also contains interpreters for several non-SCUMM games. Currently these are Beneath a Steel Sky, Broken Sword 1 & 2, Flight of the Amazon Queen and Simon the Sorcerer 1 & 2.

At this time ScummVM should be considered beta software, and is still under heavy development. Be aware that whilst we attempt to make sure that many games can be completed with few major bugs, crashes can happen.

If you enjoy ScummVM feel free to donate using the PayPal button on the ScummVM homepage. This will help us buy utilities needed to develop ScummVM easier and quicker. If you cannot donate, help and contribute a patch!

2 Contacting

The easiest way to contact the ScummVM team is by submitting bug reports or commenting in our forums. You can also join and e-mail the `scummvm-devel` mailing list, or chat with us on irc (`#scummvm` on `irc.freenode.net`) Please do not ask us to support an unsupported game – read our homepage FAQ first.

2.1 Reporting Bugs

To report a bug, please create a SourceForge account and follow the bugs link from our homepage. Please make sure the bug is reproducible, and still occurs in the latest daily build/current CVS version. Also check the known bugs list (below) and compatibility listing for that game, to ensure the issue is not already known.

Also do not report bugs on games that are not listed as being completable in the 'Supported Games' section, or compatibility list. We *know* those games have bugs.

Please include the following information:

- ScummVM version (PLEASE test the latest CVS/Daily build)
- Bug details, including instructions on reproducing
- Language of game (English, German, ...)
- Version of game (talkie, floppy, ...)
- Platform and Compiler (Win32, Linux, FreeBSD, ...)
- Attach a save game if possible
- If this bug only occurred recently, please note the last version without the bug, and the first version including the bug. That way we can fix it quicker by looking at the changes made.

3 Supported Games

At the moment the following games have been reported to work, and should be playable to the end. However, this list generally applies to PC versions. Mac versions usually should work, too (with some exceptions for older games), and Amiga versions of games may work with the 'Amiga' flag turned on – but this is not always true.

- SCUMM Games:

Maniac Mansion (Classic Version)	[Game: maniac]
Maniac Mansion (Enhanced Version)	[Game: maniac]
Zak McKracken (Classic Version)	[Game: zak]
Zak McKracken (Enhanced Version)	[Game: zak]
Indiana Jones & the Last Crusade (EGA)	[Game: indy3ega]
Indiana Jones & the Last Crusade (256)	[Game: indy3]
Indiana Jones & the Last Crusade (FM-TOWNS)	[Game: indy3towns]
Loom (16 color floppy version)	[Game: loom]
Loom (FM-TOWNS)	[Game: loomtowns]
Loom (256 color CD version)	[Game: loomcd]
Zak McKracken (FM-TOWNS)	[Game: zaktowns]
Monkey Island 1 (EGA)	[Game: monkeyega]
Monkey Island 1 (VGA)	[Game: monkeyvga]
Monkey Island 1 (CD)	[Game: monkey/monkey1]
Monkey Island 2	[Game: monkey2]
Indiana Jones and the Fate of Atlantis	[Game: atlantis]
Day of the Tentacle	[Game: tentacle]
Sam & Max	[Game: samnmax]
The Dig	[Game: dig]
Full Throttle	[Game: ft]
Curse of Monkey Island	[Game: comi]
Fatty Bear's Birthday Surprise	[Game: fbear]
Fatty Bear's Fun Pack	[Game: fbpack]
Putt-Putt's Fun Pack	[Game: funpack]
Putt-Putt Goes To The Moon	[Game: puttmoon]
Putt-Putt Joins the Parade	[Game: puttputt]
- Other Games:

Beneath a Steel Sky	[Game: sky]
Broken Sword I	[Game: sword1]
Broken Sword II	[Game: sword2]
Flight of the Amazon Queen	[Game: queen]
Simon the Sorcerer 1	[Game: simon1acorn/ simon1dos/ simon1talkie]
Simon the Sorcerer 2	[Game: simon2dos/ simon2talkie]

The following games should load, but are not yet fully playable. Play these at your own risk, and please do not file bug reports about them. If you want the latest updates on game compatibility, visit our web site and view the compatibility chart.

Freddi Fish 1: The Case of the Missing Kelp Seeds [Game: freddi]
 Freddi Fish 2: The Case of the Haunted Schoolhouse [Game: fredd2]
 Freddi Fish 3: The Case of the Stolen Conch Shell [Game: freddi3]
 Let's Explore the Airport with Buzzy [Game: airport]
 Let's Explore the Farm with Buzzy [Game: farm]
 Let's Explore the Jungle with Buzzy [Game: jungle]
 Pajama Sam 1: No Need to Hide When It's Dark Outside [Game: pajama]
 Pajama Sam 2: Thunder and Lightning Aren't so Frightening [Game: pajama2]
 Putt-Putt Saves the Zoo [Game: puttzoo]
 Putt-Putt Travels Through Time [Game: putttime]

The following games are SCUMM engine, but NOT supported by ScummVM (yet).

- Most Humongous Entertainment titles

Please be aware that the engine may contain bugs and unimplemented features that sometimes make it impossible to finish the game. Save often, and please file a bug report (instructions on submitting bug reports are below) if you encounter such a bug in a 'supported' game.

3.1 Copy Protection

The ScummVM team does not condone piracy. However, there are cases when LucasArts themselves bundled cracked interpreters with their own games – the data files still contain the copy protection scripts, but the interpreter bypasses them. There is no way for us to tell the difference between legitimate and pirated data files, so for the games where we know the original interpreter may have been cracked ScummVM will always have to bypass the copy protection.

At the time of writing, that includes the following games:

- Indiana Jones & the Last Crusade (EGA)
- Indiana Jones & the Last Crusade (FM-TOWNS version)
- Loom (16 color floppy version)
- Maniac Mansion
- Monkey Island 1 (EGA)
- Monkey Island 1 (VGA)
- Monkey Island 2
- Zak McKracken (EGA)
- Zak McKracken (FM-TOWNS version)

Beneath a Steel Sky (bypassed with permission from Revolution)

In some cases ScummVM will still show the copy protection screen. Try entering any answer. Chances are that it will work.

3.2 Simon the Sorcerer 1 and 2 notes

The current game targets are:

simon1acorn	Use for Simon the Sorcerer 1 for Acorn (CD)
simon1dos	Use for Simon the Sorcerer 1 for DOS (Disk)
simon2dos	Use for Simon the Sorcerer 2 for DOS (Disk)
simon1talkie	Use for Simon the Sorcerer 1 Talkie (DOS/Windows)
simon2talkie	Use for Simon the Sorcerer 2 Talkie (Amiga/DOS/Mac/Windows)

Also make sure that the platform setting is correct, since it is used to tell the difference between the various ports of the games.

If you have the dual version of Simon the Sorcerer 1 or 2 on CD, then you will find the Windows version in the main directory of CD and the DOS Talkie version in the DOS directory of the CD.

3.3 Broken Sword notes

Broken Sword 1 and 2 both come with in-game cutscenes compressed using RAD Game Tools legacy Smacker(tm) format. As RAD is unwilling to open the older legacy versions of this format to us, and have requested we not reverse engineer it, Revolution Software has kindly allowed us to provide re-encoded Broken Sword cutscenes for download on our website.

These cutscenes are provided in MPEG2 format with OGG Vorbis audio. Viewing these cutscenes thus requires a version of ScummVM compiled with both libmpeg2 (preferably 0.4.0 or greater) and libVorbis support.

The cutscenes should be placed in the main game data directory. Note that currently this requires either copying the game to harddisk or reburning customised versions of the game CDs.

3.4 Known Problems in ScummVM 0.7.1

This release has the following known problems. There is no need to report them, although patches to fix them are welcome. If you discover a bug that is not listed here, nor in the compatibility table on the web site, please see section 2.1.

CD Audio Games:

- When playing games that use CD Audio (Towns variants, Loom CD, etc), users of Microsoft Windows 2000/XP may experience random crashes. This is due to a long-standing Windows bug, resulting in corrupt game files being read from the CD. Please copy the game data to your harddrive to avoid this.

Monkey Island 1 (EGA): Loom (EGA):

- MIDI support requires the Roland update from LucasArts

Beneath a Steel Sky:

- Not a bug: CD version is missing speech for some dialog, this is normal.

4 Supported Platforms

ScummVM has been ported to run on many platforms and operating systems. Links to these ports can be found either on the ScummVM web page or by a Google search. Many thanks to the effort of porters. If you have a port of ScummVM and wish to commit it into the main CVS, feel free to contact us!

Windows	SDL	
Windows Mobile	SDL	(iPAQ and other handheld devices)
Linux	X11/OSS audio	(includes iPAQs running Linux)
Mac OS X	SDL	
AmigaOS	SDL	
MorphOS	Custom backend	
BeOS	SDL	
Acorn (RiscOS)	???	
Dreamcast	Custom backend	
GP32	Custom backend	
PalmOS	Custom backend	
UNIX	SDL	(Linux, Solaris, IRIX, *BSD)

The Dreamcast port does not support Curse of Monkey Island, nor The Dig. The PalmOS port does not support Curse of Monkey Island, Beneath a Steel Sky, nor either Simon the Sorcerer 1 or 2. The Dig will only work on some Palm devices (those with a large dynamic heap).

In the Macintosh port, the right mouse button is emulated via Cmd-Click (that is, you click the mouse button while holding the Command/Apple/Propeller key).

5 Running ScummVM

Before you run the engine, you need to put the game's datafiles in a directory. The filenames must not be in mixed case on *nix systems (for example, these are valid names: "monkey2.000", "MONKEY2.000", while this is a bad one: "Monkey2.000"). If you use a game with speech, the file monster.sou must reside in the same directory as the datafiles.

Please note that by default, ScummVM will save games in the directory it is executed from, so you should refrain from running it from more than one location. Further information, including how to specify a specific save directory to avoid this issue, are in section 6.0.

ScummVM can be launched directly by running the executable. In this case, the in-built launcher will activate. From this, you can add games (click 'Add Game'), or launch games which have already been configured.

ScummVM can also be launched into a game directly using Command Line arguments - see the next section.

5.1 Command Line Options

Usage: `scummvm [OPTIONS]... [GAME]`

<code>[GAME]</code>	Short name of game to load. For example, 'monkey' for Monkey Island. This can be either a built-in gameid, or a user configured target.
<code>-v, --version</code>	Display ScummVM version information and exit
<code>-h, --help</code>	Display a brief help text and exit
<code>-z, --list-games</code>	Display list of supported games and exit
<code>-t, --list-targets</code>	Display list of configured targets and exit
<code>-c, --config=CONFIG</code>	Use alternate configuration file
<code>-p, --path=PATH</code>	Path to where the game is installed
<code>-x, --save-slot[=NUM]</code>	Save game slot to load (default: autosave)
<code>-f, --fullscreen</code>	Force full-screen mode
<code>-F, --no-fullscreen</code>	Force windowed mode
<code>-g, --gfx-mode=MODE</code>	Select graphics scaler (see also section 5.3)
<code>-e, --music-driver=MODE</code>	Select music driver (see also section 7)
<code>-q, --language=LANG</code>	Select language (see also section 5.2)
<code>-m, --music-volume=NUM</code>	Set the music volume, 0-255 (default: 192)
<code>-o, --master-volume=NUM</code>	Set the master volume, 0-255 (default: 192)
<code>-s, --sfx-volume=NUM</code>	Set the sfx volume, 0-255 (default: 192)
<code>-r, --speech-volume=NUM</code>	Set the voice volume, 0-255 (default: 192)
<code>-n, --subtitles</code>	Enable subtitles (use with games that have voice)
<code>-b, --boot-param=NUM</code>	Pass number to the boot script (boot param)
<code>-d, --debuglevel=NUM</code>	Set debug verbosity level
<code>-u, --dump-scripts</code>	Enable script dumping if a directory called 'dumps' exists in the current directory
<code>-cdrom=NUM</code>	CD drive to play CD audio from (default: 0 = first drive)
<code>-joystick[=NUM]</code>	Enable input with joystick (default: 0 = first joystick)
<code>-platform=WORD</code>	Specify version of game (allowed values: amiga, atari, fntowns, mac, pc, windows)
<code>-savepath=PATH</code>	Path to where savegames are stored
<code>-multi-midi</code>	Enable combination of Adlib and native MIDI
<code>-native-mt32</code>	True Roland MT-32 (disable GM emulation)
<code>-output-rate=RATE</code>	Select output sample rate in Hz (e.g. 22050)
<code>-aspect-ratio</code>	Enable aspect ratio correction
<code>-alt-intro</code>	Use alternative intro for CD versions of Beneath a Steel Sky and Flight of the Amazon Queen
<code>-copy-protection</code>	Enable copy protection in SCUMM games, when ScummVM disables it by default.
<code>-demo-mode</code>	Start demo mode of Maniac Mansion (Classic version)
<code>-tempo=NUM</code>	Set music tempo (in percent, 50-200) for SCUMM games (default: 100)
<code>-talkspeed=NUM</code>	Set talk speed for SCUMM games

The meaning of most long options can be inverted by prefixing them with "no-", e.g. `-no-aspect-ratio`. This is useful if you want to override a setting in the configuration file.

The short game name ('game target') you see at the end of the command line is very important. A short list is contained at the top of this file. You can also get the current list of games and game names at:

<http://www.scummvm.org/compatibility.php>

Examples:

- Win32:

Running Monkey Island, fullscreen, from a hard disk:

```
C:\Games\scummvm.exe -f -pC:\Games\monkey\ monkey
```

Running Full Throttle from CD, fullscreen and with subtitles enabled:

```
C:\Games\scummvm.exe -f -n -pD:\resource\ ft
```

- Unix:

Running Monkey Island, fullscreen, from a hard disk:

```
/path/to/scummvm -f -p/games/LucasArts/monkey/ monkey
```

Running Full Throttle from CD, fullscreen and with subtitles enabled:

```
/path/to/scummvm -f -n -p/cdrom/resource/ ft
```

5.2 Language options

ScummVM includes a language option for Maniac Mansion, Zak McKracken, The Dig, Curse of Monkey Island, Beneath a Steel Sky. and Simon the Sorcerer 1 & 2.

- Maniac Mansion and Zak McKracken:

- en - English (default)
- de - German
- fr - French
- it - Italian
- es - Spanish

- The Dig

- jp - Japanese
- zh - Chinese
- kr - Korean

- Curse of Monkey Island

- en - English (default)
- de - German
- fr - French
- it - Italian
- pt - Portuguese
- es - Spanish
- jp - Japanese
- zh - Chinese
- kr - Korean

- Beneath a Steel Sky

- gb - English (Great Britain) (default)
- en - English USA
- de - German
- fr - French

- it - Italian
- pt - Portuguese
- es - Spanish
- se - Swedish
- Broken Sword 1
 - en - English (default)
 - de - German
 - fr - French
 - it - Italian
 - es - Spanish
 - pt - Portuguese
 - cz - Czech
- Simon the Sorcerer 1 & 2
 - en - English (default)
 - de - German
 - fr - French
 - it - Italian
 - es - Spanish
 - hb - Hebrew
 - ru - Russian

5.3 Graphics filters

ScummVM offers several anti-aliasing filters to attempt to improve visual quality. These are the same filters used in many other emulators, such as MAME. These filters take the original game graphics, and scale it by a certain fixed factor (usually 2x or 3x) before displaying them to you. So for example, if the game originally run at a resolution of 320x200 (typical for most of the SCUMM games), then using a filter with scale factor 2x will effectively yield 640x400 graphics. Likewise with a 3x filter you'll get 960x600.

They are:

normal	No filtering, no scaling. Fastest.
2x	No filtering, factor 2x (default for non 640x480 games).
3x	No filtering, factor 3x.
2xsai	2xsai filter, factor 2x.
super2xsai	Enhanced 2xsai filtering, factor 2x.
supereagle	Less blurry than 2xsai, but slower. Factor 2x.
advname2x	Doesn't rely on blurring like 2xSAI, fast. Factor 2x.
advname3x	Doesn't rely on blurring like 2xSAI, fast. Factor 3x.
hq2x	Very nice high quality filter but slow. Factor 2x.
hq3x	Very nice high quality filter but slow. Factor 3x.
tv2x	Interlace filter, tries to emulate a TV. Factor 2x.
dotmatrix	Dot matrix effect. Factor 2x.

To select a graphics filter, pass its name via the '-g' option to scummvm, for example:

```
scummvm -g advname2x monkey2
```

Note #1 Not all backends support all or any filters. The ones listed above are for the default SDL backend.

Note #2 Filters can be very slow when ScummVM is compiled in a debug configuration without optimizations. And there is always a speed impact when using any form of anti-aliasing/linear filtering.

Note #3 The FM-TOWNS version of Zak (zaktowns target) uses an original resolution of 320x240 - hence for this game scalers will scale to 640x480 or 960x720.

5.4 Hot Keys

ScummVM supports various in game hotkeys. They differ between the SCUMM and Simon games.

- Common:

Cmd-q	Quit (Mac OS X)
Ctrl-q	Quit (other unices including Linux)
Ctrl-z OR Alt-x	Quit (other platforms)
Keyboard Arrow Keys	Simulate mouse movement
Ctrl-f	Toggle fast mode
Ctrl-m	Toggle mouse capture
Ctrl-Alt 1-8	Switch between graphics filters
Ctrl-Alt + and -	Increase/Decrease the scale factor
Ctrl-Alt a	Toggle aspect-ratio correction on/off
	Most of the games use a 320x200 pixel resolution, which may look squashed on modern monitors. Aspect-ratio correction stretches the image to use 320x240 pixels instead, or a multiple thereof
Alt-Enter	Toggles full screen/windowed
- Scumm:

Ctrl 0-9 and Alt 0-9	Load and save game state
Ctrl-g	Runs in really REALLY fast mode
Ctrl-d	Starts the debugger
Tilde ~	Show/hide the debugging console
Ctrl-s	Shows memory consumption
[and]	Music volume, down/up
- and +	Text speed, slower/faster
F5	Displays a save/load box
Space	Pauses
Period (.)	Skips current line of text in some games
Enter	Simulate left mouse button press
Tab	Simulate right mouse button press
- Beneath a Steel Sky:

Ctrl 0-9 and Alt 0-9	Load and save game state
Ctrl-g	Runs in really REALLY fast mode
F5	Displays a save/load box
Escape	Skips the game intro
Period (.)	Skips current line of text
- Broken Sword I:

F5 or ESC	Displays save/load box
-----------	------------------------
- Broken Sword II:

Ctrl-d	Starts the debugger
c	Displays the credits
p	Pauses
- Flight of the Amazon Queen:

Ctrl-d	Starts the debugger
F1	Displays save/load box
- Simon the Sorcerer 1 & 2:

Ctrl 0-9 and Alt 0-9	Load and save game state
Ctrl-d	Starts the debugger
F1 - F3	Text speed, faster - slower
F10	Shows all characters and objects you can interact with
- and +	Music volume, down/up
m	Music on/off
s	Sound effects on/off
b	Background sounds on/off
p	Toggles pause
t	Switch between speech and subtitles
v	Switch between subtitles only and combined speech & subtitles (Simon the Sorcerer 2 only)

Note that using ctrl-f and ctrl-g are not recommended: games can crash when being ran faster than their normal speed, as scripts will lose synchronisation.

Ctrl-f is not supported by the Broken Sword games.

5.5 Using data files from Macintosh game versions

All LucasArts SCUMM based adventures except CMI also exist in versions for the Macintosh. ScummVM can use most (all?) of them, however, in some case some additional work is required. First off, if you are not using a Macintosh for this, accessing the CD/floppy data might be tricky, since the mac uses a special disk format called HFS which other systems usually do not support. However, there are various free tools on the net which allow reading such HFS volumes (for example "HFVExplorer" for Windows and "hfsutils" for Linux and other Unix-like operating systems).

Secondly, most of the newer games shipped only with a single data file on the Macintosh. You used to have to manually convert that data file, but this is no longer necessary, as ScummVM can now open and understand the format natively.

5.6 Multi-CD Games

In general, ScummVM does not deal very well with multi-CD games. This is because ScummVM assumes everything about a game can be found in one directory. Even if ScummVM does make some provisions for asking the user to change CD, the original games usually install a small number of files to hard disk. Unless these files can be found on all the CDs, ScummVM will be in trouble.

Fortunately, ScummVM has no problems running the games entirely from hard disk, if you create a directory with the correct combination of files. Usually, when a file appears on more than one CD you can pick either of them.

These instructions are written for the PC versions (which in some case is the only version) of the games. Windows and DOS use case-insensitive file systems, so if one CD has a file called MONKEY.DAT and another has a file called monkey.dat, they are the same files. These instructions give file names in all lower-case names, even if that's not always how they appear on the CDs. In fact, on case-sensitive file systems you will have to make sure that all filenames use either all upper- or all lower-case letters for ScummVM to be able to find the files.

The instructions for the Broken Sword games are for the Sold-Out Software versions, which are the ones you are probably most likely to find in stores now.

5.6.1 The Curse of Monkey Island

For this game, you'll need the comi.la0, comi.la1 and comi.la2 files. The comi.la0 file can be found on either CD, but since they are identical it doesn't matter which one of them you use.

In addition, you'll need a resource subdirectory with all of the files from the resource subdirectories on both CDs. Some of the files appear on both CDs, but again they're identical.

5.6.2 Broken Sword 1

For this game, you'll need all of the files from the clusters directories on both CDs. You will also need the speech.clu files from the speech directories, but since they are not identical you'll need to rename them speech1.clu and speech2.clu for CD 1 and 2 respectively.

In addition, you will need a music subdirectory with all of the files from the music subdirectories on both CDs. Some of these files appear on both CDs, but in these cases they are either identical or, in one case, so nearly identical that it makes little difference.

ScummVM does not support the original cutscene files, so there is no need to copy them.

5.6.3 Broken Sword 2

For this game, you'll need all of the files from the clusters directories on both CDs. (Actually, a few of them may not be strictly necessary, but the ones that I'm uncertain about are all fairly small.) You will need to rename the speech.clu and music.clu files speech1.clu, speech2.clu, music1.clu and music2.clu so that ScummVM can tell which ones are from CD 1 and which ones are from CD 2. Any other files that appear in both cluster directories are identical. Use whichever you like.

In addition, you will need the cd.bin, cd.inf and startup.inf files from the sword2 directory on CD 1.

ScummVM does not support the original cutscene files, so there is no need to copy them.

6 Savegames

Savegames are by default put in the current directory. You can specify the save in the config file by setting the savepath parameter. See the example config file later in this readme.

You can also use the environment variable SCUMMVM_SAVEPATH to specify where to put save games. Don't forget the trailing directory separator. Also be aware that saved games may break between ScummVM releases.

Bash (Unix) example:

```
export SCUMMVM_SAVEPATH=/tmp/scummvm_savegames/
```

Windows example:

```
set SCUMMVM_SAVEPATH=C:\saved_games\
```

6.1 Autosaves

Because ScummVM is still a beta product, it -can- crash and/or hang occasionally. As such, every five minutes it will save a game in Slot 0. This game can be loaded via Ctrl-0, or the F5 menu. This autosaving does not, however, occur with Simon the Sorcerer 1 and 2, nor with Broken Sword I and II.

7 Music and Sound

By default, on most operating systems, ScummVM will automatically use Adlib emulation. MIDI may not be available on all operating systems or may need manual configuration. If you ARE using MIDI, you have several different choices of output, depending on your operating system and configuration.

adlib	Uses internal Adlib Emulation (default)
mt32	Uses internal MT-32 Emulation
pcjr	Uses internal PCjr Emulation
pcspk	Uses internal PC Speaker Emulation
towns	Uses FM-TOWNS YM2612 Emulation
windows	Windows MIDI. Uses built-in sequencer, for Windows users
seq	Uses /dev/sequencer for MIDI, *nix users. See below.
qt	Quicktime sound, for Macintosh users.
core	CoreAudio sound, for MacOS X users.
amidi	Uses the MorphOS MIDI system, for MorphOS users
alsa	Output using ALSA sequencer device. See below.
null	Null output. Don't play any music.

To select a sound driver, pass its name via the '-e' option to scummvm, for example:

```
scummvm -e adlib monkey2
```

7.1 Playing sound with Adlib emulation

By default an Adlib card will be emulated and ScummVM will output the music as sampled waves. This is the default mode for most games, and offers the best compatibility between machines and games.

7.2 Playing sound with MT-32 emulation

Some games which contain MIDI music data also have improved tracks designed for MT-32 sound module. ScummVM can now emulate this card, however you should provide original MT-32 ROMs to make it work. Put the roms in game directory or directory specified by extrapath.

You don't need to specify `-native-mt32` with this driver, as it automatically gets turned on.

NOTE: You need to have enough processor power to use this emulator as it uses heavy floating-point computations.

7.3 Playing sound with MIDI emulation

Some games (such as Sam and Max) only contain MIDI music data. This once prevented music for these games from working on platforms that do not support MIDI, or soundcards that do not provide MIDI drivers (e.g. many soundcards will not play MIDI under Linux). ScummVM can now emulate MIDI mode using sampled waves and Adlib or MT-32 emulation using the `-eadlib` or `-emt32` options respectively. However, if you are capable of using native MIDI, we recommend using one of the MIDI modes below for best sound.

7.4 Playing sound with Native MIDI

Use the appropriate `-emode` command line option from the list above to select your preferred MIDI device. For example, if you wish to use the Windows MIDI driver, use the `-ewindows` option.

7.5 Playing sound with Sequencer MIDI

If your soundcard driver supports a sequencer, you may set the environment variable "SCUMMVM_MIDI" to your sequencer device - e.g., `/dev/sequencer`

If you have problems with not hearing audio in this configuration, it is possible you will need to set the "SCUMMVM_MIDI_PORT" variable to 1 or 2. This selects the port on the selected sequencer to use. Then start scummvm with the `-eseq` parameter. This should work on several cards, and may offer better performance and quality than Adlib emulation. However, for those systems where sequencer support does not work, you can always fall back on Adlib emulation.

7.5.1 Playing sound with ALSA sequencer

If you have installed the ALSA driver with the sequencer support, then set the environment variable `SCUM-MVM_PORT` or the config file parameter `alsa_port` to your sequencer port. The default is "65:0".

Here is a little howto on how to use the ALSA sequencer with your soundcard. In all cases, to have a list of all the sequencer ports you have, try the command

```
aconnect -o -l
```

This should give output similar to:

```
client 64: 'External MIDI 0' [type=kernel]
  0 'MIDI 0-0'
client 65: 'Emul0k1 WaveTable' [type=kernel]
  0 'Emul0k1 Port 0'
  1 'Emul0k1 Port 1'
  2 'Emul0k1 Port 2'
  3 'Emul0k1 Port 3'
client 128: 'Client-128' [type=user]
  0 'TiMidity port 0'
  1 'TiMidity port 1'
```

This means the external MIDI output of the sound card is located on the port 64:0, four WaveTable MIDI outputs in 65:0, 65:1, 65:2 and 65:3, and two TiMidity ports, located at 128:0 and 128:1.

If you have a FM-chip on your card, like the SB16, then you have to load the soundfonts using the sbiload software.

Example:

```
sbiload -p 65:0 /etc/std.o3 /etc/drums.o3
```

If you have a WaveTable capable sound card, you have to load a sbk or sf2 soundfont using the sfxload software

Example:

```
sfxload /path/to/8mbgmsfx.sf2
```

If you don't have a MIDI capable soundcard, there are two options: FluidSynth and TiMidity. We recommend FluidSynth, as on many systems TiMidity will 'lag' behind music. This is very noticable in iMUSE-enabled games, which use fast and dynamic music transitions. Running TiMidity as root will allow it to setup real time priority, which may reduce music lag.

Asking TiMidity to become an ALSA sequencer:

```
timidity -iAqqq -B2,8 -Os1S -s 44100 &
```

If you get distorted output with this setting, you can try dropping the -B2,8 or changing the value.

Asking FluidSynth to become an ALSA sequencer (using SoundFonts):

```
fluidsynth -m alsa_seq /path/to/8mbgmsfx.sf2
```

Once either TiMidity or FluidSynth are running, use

```
aconnect -o -l
```

as described earlier in this section.

7.6 Using compressed audiofiles (MP3, Ogg Vorbis, Flac)

7.6.1 Using MP3 files for CD audio

Use LAME or some other MP3 encoder to rip the cd audio tracks to files. Name the files track1.mp3 track2.mp3 etc. ScummVM must be compiled with MAD support to use this option. You'll need to rip the file from the CD as a WAV file, then encode the MP3 files in constant bit rate. This can be done with the following LAME command line:

```
lame -t -q 0 -b 96 track1.wav track1.mp3
```

7.6.2 Using Ogg Vorbis files for CD audio

Use oggenc or some other vorbis encoder to encode the audio tracks to files. Name the files track1.ogg track2.ogg etc. ScummVM must be compiled with vorbis support to use this option. You'll need to rip the files from the CD as a WAV file, then encode the vorbis files. This can be done with the following oggenc command line with the value after q specifying the desired quality from 0 to 10:

```
oggenc -q 5 track1.wav
```

7.6.3 Using Flac files for CD audio

Use flac or some other flac encoder to encode the audio tracks to files. Name the files track1.flac track2.flac etc. In your filesystem only allows three letter extensions, name the files track1 fla track2 fla etc. ScummVM must be compiled with flac support to use this option. You'll need to rip the files from the CD as a WAV file, then encode the flac files. This can be done with the following flac command line:

```
flac --best track1.wav
```

Remember that the quality is always the same, varying encoder options will only affect the encoding time and resulting filesize.

7.6.4 Compressing MONSTER.SOU with MP3

You need LAME, and our extract util from the scummvm-tools package to perform this task, and ScummVM must be compiled with MAD support.

```
extract monster.sou
```

Eventually you will have a much smaller monster.so3 file, copy this file to your game directory. You can safely remove the monster.sou file.

7.6.5 Compressing MONSTER.SOU with Ogg Vorbis

As above, but ScummVM must be compiled with OGG support. Run:

```
extract --vorbis monster.sou
```

This should produce a smaller monster.sog file, which you should copy to your game directory. Ogg encoding may take a considerable longer amount of time than MP3, so have a good book handy.

7.6.6 Compressing MONSTER.SOU with Flac

As above, but ScummVM must be compiled with Flac support. Run:

```
extract --flac --best -b 1152 monster.sou
```

This should produce a smaller monster.sof file, which you should copy to your game directory. Remember that the quality is always the same, varying encoder options will only affect the encoding time and resulting filesize. Playing with the blocksize (-b ;value;), has the biggest impact on the resulting filesize – 1152 seems to be a good value for those kind of soundfiles. Be sure to read the encoder documentation before you use other values.

7.6.7 Compressing sfx/speech in Simon the Sorcerer 1 and 2

Use our simon2mp3 util from the scummvm-tools package to perform this task. You can choose between multiple target formats, but note that you can only use each if ScummVM was compiled with the respective decoder support enabled.

simon2mp3 effects	(For simon1acorn)
simon2mp3 simon	(For simon1acorn)
simon2mp3 effects.voc	(For simon1talkie)
simon2mp3 simon.voc	(For simon1talkie)
simon2mp3 simon.wav	(For simon1win)
simon2mp3 simon2.voc	(For simon2talkie)
simon2mp3 simon2.wav	(For simon2win)
simon2mp3 mac	(For simon2mac)

For Ogg Vorbis add -vorbis to the options, i.e.

```
simon2mp3 --vorbis
```

For Flac add -flac and optional parameters, i.e.

```
simon2mp3 --flac --best -b 1152
```

Eventually you will have a much smaller *.mp3, *.ogg or *.fla file, copy this file to your game dir. You can safely remove the old file.

7.6.8 Compressing speech/music in Broken Sword 1

The sword1mp3 tool from the scummvm-tools package can encode music and speech to MP3 as well as Ogg Vorbis. Easiest way to encode the files is simply copying the executable into your BS1 directory (together with the lame encoder) and run it from there. This way, it'll automatically encode everything to MP3. Afterwards, you can manually remove the SPEECH?.CLU files and the wave music files.

Running

```
sword1mp3 --vorbis
```

will compress the files using Ogg Vorbis instead of MP3.

Use

```
sword1mp3 --help
```

to get a full list of the options.

7.6.9 Compressing speech/music in Broken Sword 2

Use our sword2mp3 util from the scummvm-tools package to perform this task. You can choose between multiple target formats, but note that you can only use each if ScummVM was compiled with the respective decoder support enabled.

```
sword2mp3 speech1.clu
sword2mp3 music1.clu
```

For Ogg Vorbis add `-vorbis` to the options, i.e.

```
sword2mp3 --vorbis
```

Eventually you will have a much smaller *.cl3 or *.clg file, copy this file to your game dir. You can safely remove the old file.

It is possible to use Flac compression by adding the `-flac` option. However, the resulting *.clf file will actually be larger than the original.

Please note that sword2mp3 will only work with the four speech/music files in Broken Sword 2. It will not work with any of the other *.clu files, nor will it work with the speech files from Broken Sword 1.

7.7 Output sample rate

The output sample rate tells ScummVM how many sound samples to play per channel per second. There is much that could be said on this subject, but most of it would be irrelevant here. The short version is that for most games 22050 Hz is fine, but in some cases 44100 Hz is preferable. On extremely low-end systems you may want to use 11025 Hz, but it's unlikely that you have to worry about that.

To elaborate, most of the sounds ScummVM has to play were sampled at either 22050 Hz or 11025 Hz. Using a higher sample rate will not magically improve the quality of these sounds. Hence, 22050 Hz is fine.

Some games use CD audio. If you use compressed files for this, they are probably sampled at 44100 Hz, so for these games that may be a better choice of sample rate.

When using the Adlib, FM Towns, PC Speaker or IBM PCjr music drivers, ScummVM is responsible for generating the samples. Usually 22050 Hz will be plenty for these, but there is at least one piece of Adlib music in Beneath a Steel Sky that will sound a lot better at 44100 Hz.

Using frequencies in between is not recommended. For one thing, your sound card may not support it. In theory, ScummVM should fall back on a sensible frequency in that case, but don't count on it. More importantly, ScummVM has to resample all sounds to its output frequency. This is much easier to do well if the output frequency is a multiple of the original frequency.

8 Configurations Files

By default, the configuration file is saved in, and loaded from:

- Windows: <windir>\scummvm.ini
- Unix: ~/.scummvmrc
- Mac OS X: ~/Library/Preferences/ScummVM Preferences
- Others: scummvm.ini in the current directory

An example config file looks as follows:

```
[scummvm]
gfx_mode=supereagle
fullscreen=true
savepath=C:\saves\

[sky]
path=C:\games\SteelSky\

[germanskyl]
gameid=sky
language=de
path=C:\games\SteelSky\
description=Beneath a Steel Sky w/ German subtitles

[germandott]
gameid=tentacle
path=C:\german\tentacle\
description=German version of DOTT

[tentacle]
path=C:\tentacle\
subtitles=true
master_volume=98
music_volume=40
sfx_volume=255

[loomcd]
cdrom=1
path=C:\loom\
talkspeed=5
savepath=C:\loom\saves\

[monkey2]
path=C:\amiga_mi2\
music_driver=windows
amiga=true
```

The following keywords are recognized:

basename	string
path	string The path to where a game's data files are
read_only	bool If true, ScummVM will never try to overwrite the configuration file.
save_slot	number The save game number to load on startup.
savepath	string The path to where a game will store its savegames.
versioninfo	string The version of the ScummVM that created the configuration file.
gameid	string The real id of a game. Useful if you have several versions of the same game, and want different aliases for them. See the example.
description	string The description of the game as it will appear in the launcher.
language	string Specify language (en, de, fr, it, pt, es, jp, zh, kr, se, gb, hb, cz, ru)
subtitles	bool Set to true to enable subtitles.
talkspeed	number Text speed
fullscreen	bool Fullscreen mode
aspect_ratio	bool Enable aspect ratio correction
gfx_mode	string Graphics mode (normal, 2x, 3x, 2xsai, super2xsai, supereagle, advmame2x, advmame3x, hq2x, hq3x, tv2x, dotmatrix)
cdrom	number Number of CD-ROM unit to use for audio. If negative, don't even try to access the CD-ROM.
joystick_num	number Number of joystick device to use for input
master_volume	number The master volume setting (0-255)
music_driver	string The music engine to use.
output_rate	number The output sample rate to use, in Hz. Sensible values are 11025, 22050 and 44100.
alsa_port	string Port to use for output when using the ALSA music driver.
music_volume	number The music volume setting (0-255)
multi_midi	bool If true, enable combination Adlib and native MIDI.
native_mt32	bool If true, disable GM emulation and assume that there is a true Roland MT-32 available.
sfx_volume	number The sfx volume setting (0-255)
tempo	number The music tempo (50-200) (default: 100)
speech_volume	number The speech volume setting (0-255)
copy_protection	bool Enable copy protection in SCUMM games ,when ScummVM disables it by default.
demo_mode	bool Start demo in Maniac Mansion
alt_intro	bool Use alternative intro for CD versions of Beneath a Steel Sky and Flight of the Amazon Queen
boot_param	number Pass this number to the boot script
debuglevel	number Enable debug output. The higher number, the more verbose output.

Broken Sword II adds the following non-standard keywords:

<code>gfx_details</code>	number	Graphics details setting (0-3)
<code>music_mute</code>	bool	If true, music is muted
<code>object_labels</code>	bool	If true, object labels are enabled
<code>reverse_stereo</code>	bool	If true, stereo channels are reversed
<code>sfx_mute</code>	bool	If true, sound effects are muted
<code>speech_mute</code>	bool	If true, speech is muted

Flight of the Amazon Queen adds the following non-standard keywords:

<code>music_mute</code>	bool	If true, music is muted
<code>sfx_mute</code>	bool	If true, sound effects are muted
<code>speech_mute</code>	bool	If true, speech is muted

Simon the Sorcerer 1 & 2 add the following non-standard keywords:

<code>fade</code>	bool	If true, fade effect is enabled
<code>music_mute</code>	bool	If true, music is muted
<code>slow_down</code>	number	Makes games slower (1- 10)
<code>sfx_mute</code>	bool	If true, sound effects are muted
<code>speech_mute</code>	bool	If true, speech is muted
[Simon the Sorcerer 2 only]		

9 Compiling

You need SDL-1.2.2 or newer (older versions may work, but are unsupported), and a supported compiler. Several compilers, including GCC, mingw and Microsoft Visual C++ are supported. If you wish to use MP3-compressed CD tracks or .SOU files, you will need to install the MAD library and define `USE_MAD`. Tools for compressing .SOU files to .SO3 files can be found in the 'tools' CVS module, or in the 'scummvm-tools' package.

Some parts of ScummVM, particularly scalars, have highly optimized versions written in assembler. If you wish to use this option, you will need to install nasm assembler (see <http://nasm.sf.net>). Note, that currently we have only x86 MMX optimized versions, and they will not compile on other processors.

On Win9x/NT/XP you can define `USE_WINDBG` and attach WinDbg to browse debug messages (see <http://www.sysinternals.com/ntw2k/freeware/debugview.shtml>).

9.1 GCC

- Type `./configure`
- Type `make` (or `gmake`, or `gnumake`, depending on what GNU make is called on your system) and hopefully ScummVM will compile for you.

9.2 Microsoft Visual C++ 6.0

- Open the workspace, `scummwm.dsw`
- Enter the path to the needed libraries and includes in Tools—Options—Directories
- Now it should compile successfully.

9.3 Microsoft Visual C++ 7.0

- Open the solution file `scummwm.sln`
- Enter the path to the needed libraries and includes in Tools—Options—Directories
- Now it should compile successfully.

9.4 Windows Mobile with Microsoft eMbedded Visual C++ 3 or 4

- Download SDL with additional Windows Mobile tweaks:
<http://arisme.free.fr/ports/SDL.php>
- Download additional third party libraries:
<http://arisme.free.fr/ports>
- Modify your include and library paths accordingly in EVC3/EVC4.
- Open the ScummVM project dists\msevc4\PocketSCUMM.vcw
- Modify the libraries and config parameters if necessary.
- Now it should compile successfully.

9.5 Debian GNU/Linux

- Install the packages 'build-essential', 'fakeroot', 'debhelper', and 'libsdl1.2-dev' on your system.
- Install any of these packages (optional): 'libvorbis-dev' (for Ogg Vorbis support), 'libasound2-dev' (for ALSA sequencer support), 'libmad0-dev' (for MAD MP3 support), 'zlib1g-dev' (for compressed saves support).
- Run 'make deb'
- Finally run 'dpkg -i ../scummvm-cvs*.deb', and you're done.

9.6 Mac OS X

- Make sure you have the developer tools installed.
- The SDL developer package for OS X available on the SDL web site is *not* suitable. Rather, you require a unix-style build of SDL. One way to get that is to install SDL via Fink (<http://fink.sf.net>).
Alternatively you could compile SDL manually from source using its unix build system (`configure && make`).
- Type `./configure` in the ScummVM directory
- You can now type `make` to create a command line binary.
- To get a version you can run from Finder, type `make bundle` which will create ScummVM.app (this only works if you installed SDL etc. via Fink and into /sw. If you have installed SDL in another way, you'll have to edit the Makefile).

10 Credits

• The ScummVM team

James Brown	Lead developer
Max Horn	Lead developer
Torbjörn Andersson	Engine: SCUMM, Broken Sword II, SAGA
David Eriksson	Engine: Flight of the Amazon Queen
Robert Göffringmann	Engine: Beneath a Steel Sky, Broken Sword I
Jonathan Gray	Engine: SCUMM, HE, Broken Sword II
Travis Howell	Engine: SCUMM, HE, Simon the Sorcerer
Oliver Kiehl	Engine: Beneath a Steel Sky, Simon
Pawel Kolodziejski	Engine: SCUMM (Codecs, iMUSE, Smush, etc.)
Andrew Kurushin	Engine: SAGA
Gregory Montoir	Engine: Flight of the Amazon Queen, HE
Joost Peters	Engine: Beneath a Steel Sky, Flight of the Amazon Queen
Eugene Sandulenko	Engine: SCUMM (FT INSANE), HE, SAGA
Chris Apers	Port: PalmOS
Nicolas Bacca	Port: PocketPC/WinCE
Marcus Comstedt	Port: Dreamcast
Ruediger Hanke	Port: MorphOS
Jamieson Christian	iMUSE, MIDI, all things musical
Jerome Fisher	MT-32 emulator
Jochen Hoenicke	Speaker & PCjr sound support, Adlib work

• Retired Team Members

Ralph Brorsen	Help with GUI implementation
Vincent Hamm	Co-Founder
Felix Jakschitsch	Zak256 reverse engineering
Mutwin Kraus	Original MacOS porter
Peter Moraliyski	Port: GP32
Jeremy Newman	Former webmaster
Ludvig Strigeus	Original ScummVM and SimonVM author
Lionel Ulmer	Port: X11

• Contributors

Tore Anderson	Packaging for Debian GNU/Linux
Stuart Caie	Decoders for Simon 1 Amiga data files
Janne Huttunen	V3 actor mask support, Dig/FT SMUSH audio
Kovács Endre János	Several fixes for Simon1
Jeroen Janssen	Numerous readability and bugfix patches
Robert Kelsen	Packaging for SlackWare
Claudio Matsuoka	Daily Linux/BeOS builds
Mikesch Nepomuk	MI1 VGA floppy patches
Nicolas Noble	Config file and ALSA support
Willem Jan Palenstijn	Packaging for Fedora/RedHat
Quietust	Sound support for Amiga SCUMM V2/V3 games
Andreas Röver	Broken Sword 1/2 MPEG2 cutscene support
Edward Rudd	Fixes for playing MP3 versions of MI1/Loom audio
Daniel Schepler	Final MI1 CD music support, initial Ogg Vorbis support
André Souza	SDL-based OpenGL renderer
Tim ???	Initial MI1 CD music support

And to all the contributors, users, and beta testers we've missed. Thanks!

• Special thanks to

Sander Buskens	For his work on the initial reversing of Monkey2
Canadacow	For the original MT-32 emulator
Kevin Carnes	For Scumm16, the basis of ScummVM's older gfx codecs
Jezar	For his freeverb filter implementation
Jim Leiterman	Various info on his FM-TOWNS/Marty SCUMM ports
Jimmi Thøgersen	For ScummRev, and much obscure code/documentation
Tristan	For additional work on the original MT-32 emulator

Tony Warriner and everyone at Revolution Software Ltd. for sharing with us the source of some of their brilliant games, allowing us to release Beneath a Steel Sky as freeware... and generally being supportive above and beyond the call of duty.

John Passfield and Steve Stamatiadis for sharing the source of their classic title, Flight of the Amazon Queen and also being incredibly supportive.

Joe Pearce from The Wyrnkeep Entertainment Co. for sharing the source of their famous title Inherit the Earth and always prompt replies to our questions.

Aric Wilmunder, Ron Gilbert, David Fox, Vince Lee, and all those at LucasFilm/LucasArts who made SCUMM the insane mess to reimplement that it is today. Feel free to drop us a line and tell us what you think, guys!