

ScummVM ReadMe

The ScummVM Team

2006/10/29

Contents

1	About	2
2	Contacting	2
2.1	Reporting Bugs	2
3	Supported Games	2
3.1	Copy Protection	4
3.2	Commodore64 games notes	4
3.3	Maniac Mansion NES notes	4
3.4	Macintosh games notes	5
3.5	Multi-CD games notes	5
3.6	The Curse of Monkey Island notes	5
3.7	Broken Sword notes	5
3.7.1	Broken Sword 1	6
3.7.2	Broken Sword 2	6
3.8	Beneath a Steel Sky notes	6
3.9	Flight of the Amazon Queen notes	6
3.10	Gobliiins notes	6
3.11	Inherit the Earth: Quest for the Orb notes	7
3.12	Simon the Sorcerer 1 and 2 notes	7
3.13	The Feeble Files notes	7
3.14	The Legend of Kyrandia notes	7
3.15	Known Problems	7
4	Supported Platforms	9
5	Running ScummVM	9
5.1	Command Line Options	11
5.2	Language options	12
5.3	Graphics filters	13
5.4	Hot Keys	14

6	Savegames	15
6.1	Autosaves	16
7	Music and Sound	16
7.1	Playing sound with Adlib emulation	16
7.2	Playing sound with FluidSynth MIDI emulation	16
7.3	Playing sound with MT-32 emulation	17
7.4	Playing sound with MIDI emulation	17
7.5	Playing sound with Native MIDI	17
7.5.1	Using MIDI options to customize Native MIDI output	17
7.6	Playing sound with Sequencer MIDI	17
7.6.1	Playing sound with ALSA sequencer	18
7.7	Using compressed audiofiles (MP3, Ogg Vorbis, Flac)	19
7.7.1	Using MP3 files for CD audio	19
7.7.2	Using Ogg Vorbis files for CD audio	19
7.7.3	Using Flac files for CD audio	19
7.7.4	Compressing MONSTER.SOU with MP3	19
7.7.5	Compressing MONSTER.SOU with Ogg Vorbis	19
7.7.6	Compressing MONSTER.SOU with Flac	20
7.7.7	Compressing sfx/speech in Simon the Sorcerer and The Feeble Files	20
7.7.8	Compressing speech/music in Broken Sword 1	20
7.7.9	Compressing speech/music in Broken Sword 2	21
7.8	Output sample rate	21
8	Configuration file	21
9	Compiling	24
9.1	GCC	24
9.2	Microsoft Visual C++ 6.0	24
9.3	Microsoft Visual C++ 7.0	24
9.4	Windows Mobile with Microsoft eMbedded Visual C++ 3 or 4	25
9.5	Debian GNU/Linux	25
9.6	Mac OS X	25
10	Credits	25

1 About

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

ScummVM also contains interpreters for several non-SCUMM games. Currently these are Beneath a Steel Sky, Broken Sword 1 and 2, Flight of the Amazon Queen, Gbaliins, Inherit the Earth: Quest for the Orb, Simon the Sorcerer 1 and 2, The Feeble Files and The Legend of Kyrandia.

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 the FAQ on our web site 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 SVN/Daily build version. Also check the known problems list (below) and the compatibility list on our website for that game, to ensure the issue is not already known:

http://www.scummvm.org/compatibility_stable.php

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 SVN/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 savegame 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:

- SCUMM Games by LucasArts:

Maniac Mansion	[maniac]
Zak McKracken and the Alien Mindbenders	[zak]
Indiana Jones and the Last Crusade	[indy3]
Loom	[loom]
The Secret of Monkey Island	[monkey]
Monkey Island 2: LeChuck's Revenge	[monkey2]
Indiana Jones and the Fate of Atlantis	[atlantis]
Day of the Tentacle	[tentacle]
Sam & Max Hit the Road	[samnmax]
Full Throttle	[ft]
The Dig	[dig]
The Curse of Monkey Island	[comi]

- Other Games:

Beneath a Steel Sky	[sky]
Broken Sword 1: The Shadow of the Templars	[sword1]
Broken Sword 2: The Smoking Mirror	[sword2]
Flight of the Amazon Queen	[queen]
Gobliins	[gob1]
Inherit the Earth: Quest for the Orb	[ite]
Simon the Sorcerer 1	[simon1]
Simon the Sorcerer 2	[simon2]
The Feeble Files	[feeble]
The Legend of Kyrandia	[kyra1]

- SCUMM Games by Humongous Entertainment:

Backyard Football	[football]
Big Thinkers First Grade	[thinker1]
Big Thinkers Kindergarten	[thinkerk]
Fatty Bear's Birthday Surprise	[fbear]
Fatty Bear's Fun Pack	[fbpack]
Freddi Fish 1: The Case of the Missing Kelp Seeds	[freddi]
Freddi Fish 2: The Case of the Haunted Schoolhouse	[freddi2]
Freddi Fish 3: The Case of the Stolen Conch Shell	[freddi3]
Freddi Fish 4: The Case of the Hogfish Rustlers of Briny Gulch	[freddi4]
Freddi Fish and Luther's Maze Madness	[maze]
Freddi Fish and Luther's Water Worries	[water]
Let's Explore the Airport with Buzzy	[airport]
Let's Explore the Farm with Buzzy	[farm]
Let's Explore the Jungle with Buzzy	[jungle]
Pajama Sam 1: No Need to Hide When It's Dark Outside	[pajama]
Pajama Sam 2: Thunder and Lightning Aren't so Frightening	[pajama2]
Pajama Sam 3: You Are What You Eat From Your Head to Your Feet	[pajama3]
Pajama Sam's Lost & Found	[lost]
Pajama Sam's Sock Works	[socks]
Putt-Putt Enters the Race	[puttrace]
Putt-Putt Goes to the Moon	[puttmoon]
Putt-Putt Joins the Circus	[puttcircus]
Putt-Putt Joins the Parade	[puttputt]
Putt-Putt Saves the Zoo	[puttzoo]
Putt-Putt Travels Through Time	[putttime]
Putt-Putt and Pep's Balloon-O-Rama	[balloon]
Putt-Putt and Pep's Dog on a Stick	[dog]
Putt-Putt & Fatty Bear's Activity Pack	[activity]
Putt-Putt's Fun Pack	[funpack]
SPY Fox 1: Dry Cereal	[spyfox]
SPY Fox 2: Some Assembly Required	[spyfox2]
SPY Fox in Cheese Chase	[chase]
SPY Fox in Hold the Mustard	[mustard]

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.

Backyard Baseball	[baseball]
Backyard Soccer	[soccer]
Blue's ABC Time [BluesABCTime]	
Blue's Birthday Adventure [BluesBirthday]	
SPY Fox 3: Operation Ozone	[spyozon]

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

Other 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 above) 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.

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

ScummVM will skip copy protection in the following games:

- Maniac Mansion
- Zak McKracken and the Alien Mindbenders
- Loom (EGA)
- The Secret of Monkey Island (VGA)
- Monkey Island 2: LeChuck's Revenge

Beneath a Steel Sky bypassed with kind permission from Revolution Software.

Inherit the Earth: Quest for the Orb (Floppy version) bypassed with kind permission from Wyrmskeep Entertainment, since it was bypassed in all CD releases of the game.

3.2 Commodore64 games notes

Both Maniac Mansion and Zak McKracken run but Maniac Mansion is not yet playable. Either use `extract_mmm_c64` from the tools package (but then the game will not be autodetected by ScummVM) or name the D64 disks as "maniac1.d64", "maniac2.d64" and "zak1.d64", "zak2.d64" respectively. If you add the game manually, make sure that the platform is set to Commodore64.

3.3 Maniac Mansion NES notes

Supported versions are English GB (E), French (F), German (G), Swedish (SW) and English US (U). ScummVM requires just the PRG section to run and not the whole ROM.

In order to get the game working, you will have to strip out the first 16 bytes from the ROM you are trying to work with. Any hex editor will work as long as you are able to copy/paste. After you open the ROM with the hex editor, copy everything from the second row (17th byte) to the end. After you do this, paste it to a new hex file. Name the new file "Maniac Mansion (XX).prg" while XX stands for the version you are working with (E, F, G, SW, or U). The final size should be exactly 262144 bytes.

If you add the game manually make sure that the platform is set to NES.

Most common mistakes which prevents the game from running:

- Bad file
- ROM extracted with the 0.7.0 tools
- You try to feed ScummVM with the FULL ROM and not just the PRG section.

It is also possible to extract the separate LFL files from the PRG section. To do so use the `extract_mm_nes` utility from the tools package.

3.4 Macintosh games notes

All LucasArts SCUMM based adventures except COMI also exist in versions for the Macintosh. ScummVM can use most (all?) of them, however, in some cases 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.

For further information on copying Macintosh game files to your hard disk see:

http://wiki.scummvm.org/index.php/HOWTO-Mac_Games

3.5 Multi-CD games notes

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 the 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.

3.6 The Curse of Monkey Island notes

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.

3.7 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:

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

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 hard disk or reburning customised versions of the game CDs.

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.

3.7.1 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.

3.7.2 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.

3.8 Beneath a Steel Sky notes

Starting with ScummVM 0.8.0 you need the additional 'SKY.CPT' file to run Beneath a Steel Sky.

This file is available on the 'Compatibility' page of the ScummVM website, you can place it in either the directory containing the other game data files (SKY.DNR, SKY.DSK), in your extrapath, or in the directory where your ScummVM executable resides.

3.9 Flight of the Amazon Queen notes

In order to use a non-freeware version of Flight of the Amazon Queen (from original CD), you will need to place the 'queen.tbl' file (available from the Compatibility page on our website) in either the directory containing the 'queen.1' game data file, in your extrapath, or in the directory where your ScummVM executable resides.

Alternatively, you can use the 'compress_queen' tool from the tools package to 'rebuild' your FOTAQ data file to include the table for that specific version, and thus removing the run-time dependency on the 'queen.tbl' file. This tool also allows you to compress the speech and sound effects with MP3, OGG or FLAC.

3.10 Gblliins notes

The CD version of Gblliins contains one big audio track which you need to rip (See the section on using compressed audio files) and copy into the game directory if you want to have in-game music without the CD in the drive all the time.

3.11 Inherit the Earth: Quest for the Orb notes

In order to run the Mac OS X Wyrmskeep re-release of the game you will need to copy over data from the CD to your hard drive. If you're on a PC then consult:

http://wiki.scummvm.org/index.php/HOWTO-Mac_Games

Although it primarily talks about SCUMM games, it mentions the "HFVExplorer" utility which you need to extract the files. Note that you have to put the speech data "Inherit the Earth Voices" in the same directory as the game data which is stored in:

```
Inherit the Earth.app/Contents/Resources
```

For the old Mac OS 9 release you need to copy the files in MacBinary format, as they should include both resource and data forks. Copy all 'ITE *' files.

3.12 Simon the Sorcerer 1 and 2 notes

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

3.13 The Feeble Files notes

If you have the Windows version of The Feeble Files, there are several things to note.

Many of the files necessary for the game are stored in an InstallShield file called data1.cab, which ScummVM is unable to unpack. You will need to use the original installer or i5comp to unpack the contents of this file.

The game uses Smacker cutscenes extensively, which we can't support directly, due to reasons described above. The cutscenes have to be re-encoded to DXA, which is the format used by the cutscenes in the Amiga and Macintosh versions. See the tools README for a brief guide on converting the cutscenes.

The original speech files will need to be renamed, to use with ScummVM. Rename voices.wav on CD1 to voices1.wav Rename voices.wav on CD2 to voices2.wav Rename voices.wav on CD3 to voices3.wav Rename voices.wav on CD4 to voices4.wav

3.14 The Legend of Kyrandia notes

To run The Legend of Kyrandia under ScummVM you need the 'kyra.dat' file, which can be found in the downloads section of the ScummVM website.

3.15 Known Problems

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 list on the web site, please see the section on reporting bugs. 2.1.

CD Audio Games:

- When playing games that use CD Audio (FM-TOWNS games, 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.

FM-TOWNS Versions:

- The Kanji versions require the FM-TOWNS Font ROM
- ScummVM will crash randomly when using the FM-TOWNS Font ROM for the Kanji versions of the following games: The Secret of Monkey Island, Monkey Island 2: LeChuck's Revenge and Indiana Jones and the Fate of Atlantis

Loom (EGA): The Secret of Monkey Island (EGA):

- MIDI support requires the Roland update from LucasArts

Beneath a Steel Sky:

- Amiga versions aren't supported
- Floppy demos aren't supported
- Not a bug: CD version is missing speech for some dialogs, this is normal.

Broken Sword 1:

- Macintosh version isn't supported
- PlayStation 1 version isn't supported

Broken Sword 2:

- PlayStation 1 version isn't supported

Flight of the Amazon Queen:

- Amiga versions aren't supported

Inherit the Earth:

- Amiga CD version isn't supported

Simon the Sorcerer 1:

- Amiga versions aren't supported
- Subtitles aren't available in the English and German CD versions as they are missing the majority of subtitles.

Simon the Sorcerer 2:

- Combined speech and subtitles will often cause speech to be cut off early, this is a limitation of the original game.
- Only default language (English) of data files is supported in Amiga and Macintosh versions.

The Feeble Files

- Subtitles are often incomplete, they were always disabled in the original game.

Humongous Entertainment Games

- No support for printing images

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 SVN, feel free to contact us!

Supported platforms include (but are not limited to):

UNIX	(Linux, Solaris, IRIX, *BSD)
Windows	
Windows Mobile	(iPAQ and other handheld devices)
Mac OS X	
AmigaOS	
BeOS	
Dreamcast	
PalmOS	
Playstation 2	
Playstation Portable	
RISC OS	
Symbian	

The Dreamcast port does not support The Curse of Monkey Island, nor The Dig. The PalmOS port does not support The 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).

For the following platforms, custom backends were supported at one point but currently are not being maintained (usually due to lack of a qualified maintainer). In some cases (e.g. Linux), the standard SDL support works instead.

Linux	(includes iPAQs running Linux)
MorphOS	
GP32	

We have reports about unofficial ports to the following platforms. Please note that these are not made by us, so we neither endorse nor support them. Use at your own risk!

Nintendo DS
Xbox

5 Running ScummVM

Before you run the engine, you need to put the game's data files 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 data files.

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 built-in 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]

[GAME]	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.
-v, --version	Display ScummVM version information and exit
-h, --help	Display a brief help text and exit
-z, --list-games	Display list of supported games and exit
-t, --list-targets	Display list of configured targets and exit
-c, --config=CONFIG	Use alternate configuration file
-p, --path=PATH	Path to where the game is installed
-x, --save-slot[=NUM]	Savegame slot to load (default: autosave)
-f, --fullscreen	Force full-screen mode
-F, --no-fullscreen	Force windowed mode
-g, --gfx-mode=MODE	Select graphics scaler (see also section 5.3)
--gui-theme=THEME	Select GUI theme (default, modern, classic)
--themepath=PATH	Path to where GUI themes are stored
-e, --music-driver=MODE	Select music driver (see also section 7)
-q, --language=LANG	Select language (see also section 5.2)
-m, --music-volume=NUM	Set the music volume, 0-255 (default: 192)
-s, --sfx-volume=NUM	Set the sfx volume, 0-255 (default: 192)
-r, --speech-volume=NUM	Set the voice volume, 0-255 (default: 192)
--midi-gain	Set the gain for MIDI playback, 0-1000 (default: 100) (only supported by some MIDI drivers)
-n, --subtitles	Enable subtitles (use with games that have voice)
-b, --boot-param=NUM	Pass number to the boot script (boot param)
-d, --debuglevel=NUM	Set debug verbosity level
-u, --dump-scripts	Enable script dumping if a directory called 'dumps' exists in the current directory
--cdrom=NUM	CD drive to play CD audio from (default: 0 = first drive)
--joystick[=NUM]	Enable input with joystick (default: 0 = first joystick)
--platform=WORD	Specify version of game (allowed values: 3do, acorn, amiga, atari, c64, fntowns, mac, nes, pc, segacd, windows)
--savepath=PATH	Path to where savegames are stored
--multi-midi	Enable combination of Adlib and native MIDI
--soundfont	Select the SoundFont for MIDI playback. (Only supported by some MIDI drivers.)
--native-mt32	True Roland MT-32 (disable GM emulation)
--enable-gs	Enable Roland GS mode for MIDI playback
--output-rate=RATE	Select output sample rate in Hz (e.g. 22050)
--aspect-ratio	Enable aspect ratio correction
--render-mode=MODE	Enable additional render modes (cga, ega, hercGreen, hercAmber, amiga)
--alt-intro	Use alternative intro for CD versions of Beneath a Steel Sky and Flight of the Amazon Queen
--copy-protection	Enable copy protection in games, when ScummVM disables it by default.
--demo-mode	Start demo mode of Maniac Mansion (Classic version)
--tempo=NUM	Set music tempo (in percent, 50-200) for SCUMM games (default: 100)
--talkspeed=NUM	Set talk speed for 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_stable.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, The Curse of Monkey Island, Beneath a Steel Sky, Broken Sword 1 and Simon the Sorcerer 1 and 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

- The 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 and 2
 - en - English (default)
 - de - German
 - fr - French
 - it - Italian
 - es - Spanish
 - hb - Hebrew
 - pl - Polish
 - 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:

1x	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 -gadvame2x 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 McKracken 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 SCUMM games and other 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-d	Starts the debugger
Ctrl-g	Runs in really REALLY fast mode
Ctrl-t	Switch between 'Speech only', 'Speech and Subtitles' and 'Subtitles only'
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-d	Starts the debugger
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 1:

F5 or ESC	Displays save/load box
-----------	------------------------
- Broken Sword 2:

Ctrl-d	Starts the debugger
c	Displays the credits
p	Pauses

- Ctrl-d Starts the debugger
 - F1 Displays save/load box
 - F11 Quicksave
 - F12 Quickload
 - Escape Skips cutscenes
 - Space Skips current line of text
- Flight of the Amazon Queen:
 - 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 Pauses
 - t Switch between speech and subtitles
 - v Switch between subtitles only and combined speech and subtitles (Simon the Sorcerer 2 only)
- The Feeble Files:
 - Ctrl-d Starts the debugger
 - F7 Switch characters
 - F9 Hitbox names on/off
 - s Sound effects on/off
 - b Background sounds on/off
 - p Pauses
 - t Switch between speech and subtitles
 - v Switch between subtitles only and combined speech and subtitles (Simon the Sorcerer 2 only)
- The Legend of Kyrandia:
 - Ctrl 0-9 and Alt 0-9 Load and save game state
 - Ctrl-d Starts the debugger

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

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

6 Savegames

Savegames are by default put in the current directory on some platforms and preset directories on others. You can specify the save in the config file by setting the savepath parameter. See the example config file later in this readme.

The platforms that currently have a different default directory are:

- Mac OS X: `$HOME/Documents/ScummVM Savegames/`
- Other unices: `$HOME/.scummvm/`

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 1 and 2.

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.

null	Null output. Don't play any music.
adlib	Uses internal Adlib Emulation (default)
fluidsynth	Uses FluidSynth MIDI Emulation
mt32	Uses internal MT-32 Emulation
pcjr	Uses internal PCjr Emulation
pcspk	Uses internal PC Speaker Emulation
towns	Uses FM-TOWNS YM2612 Emulation
alsa	Output using ALSA sequencer device. See below.
core	CoreAudio sound, for Mac OS X users.
coremidi	CoreMIDI sound, for Mac OS X users. Use only if you have a hardware MIDI synthesizer.
qt	Quicktime sound, for Macintosh users.
seq	Uses /dev/sequencer for MIDI, *nix users. See below.
windows	Windows MIDI. Uses built-in sequencer, for Windows users

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

```
scummvm -eadlib 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 FluidSynth MIDI emulation

If ScummVM was build with libfluidsynth support it will be able to play MIDI music through the FluidSynth driver. You will have to specify a SoundFont to use, however.

Since the default output volume from FluidSynth can be fairly low, ScummVM will set the gain by default to get a stronger signal. This can be further adjusted using the `-midi-gain` command-line option, or the "midi_gain" config file setting.

The setting can take any value from 0 through 1000, with the default being 100. (This corresponds to FluidSynth's gain settings of 0.0 through 10.0, which are presumably measured in decibel.)

NOTE: The processor requirements for FluidSynth can be fairly high in some cases. A fast CPU is recommended.

7.3 Playing sound with MT-32 emulation

Some games which contain MIDI music data also have improved tracks designed for the MT-32 sound module. ScummVM can now emulate this device, however you must provide original MT-32 ROMs to make it work:

MT32_PCM.ROM - IC21 (512KB)

MT32_CONTROL.ROM - IC26 (32KB) and IC27 (32KB), interleaved byte-wise

Place these ROMs in the game directory, in your extrapath, or in the directory where your ScummVM executable resides.

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

NOTE: The processor requirements for the emulator are quite high; a fast CPU is strongly recommended.

7.4 Playing sound with MIDI emulation

Some games (such as Sam & 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, FluidSynth MIDI emulation or MT-32 emulation using the `-eadlib`, `-efluidsynth` 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.

7.5 Playing sound with Native MIDI

Use the appropriate `-e|mode|` 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.

7.5.1 Using MIDI options to customize Native MIDI output

ScummVM supports a variety of MIDI modes, depending on the capabilities of your MIDI device.

If `-native-mt32` is specified, ScummVM will treat your device as a real MT-32. Because the instrument mappings and system exclusive commands of the MT-32 vary from those of General MIDI devices, you should only enable this option if you are using an actual Roland MT-32, LAPC-I, CM-64, CM-32L, CM-500, or GS device with an MT-32 map.

If `-enable-gs` is specified, ScummVM will initialize your GS-compatible device with settings that mimic the MT-32's reverb, (lack of) chorus, pitch bend sensitivity, etc. If it is specified in conjunction with `-native-mt32`, ScummVM will select the MT-32-compatible map and drumset on your GS device. This setting works better than default GM or GS emulation with games that do not have custom instrument mappings (Loom and Monkey1). You should only specify both settings if you are using a GS device that has an MT-32 map, such as an SC-55, SC-88, SC-88 Pro, SC-8820, SC-8850, etc. Please note that `-enable-gs` is automatically disabled in both DOTT and Samnmax, since they use General MIDI natively.

If neither of the above settings is enabled, ScummVM will initialize your device in General MIDI mode and use GM emulation in games with MT-32 soundtracks.

Some games contain sound effects that are exclusive to the Adlib soundtrack. For these games, you may wish to specify `-multi-midi` in order to combine MIDI music with Adlib sound effects.

7.6 Playing sound with Sequencer MIDI

If your soundcard driver supports a sequencer, you may set the environment variable `"SCUMMVM_MIDI"` to your sequencer device – for example, to `/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.

7.6.1 Playing sound with ALSA sequencer

If you have installed the ALSA driver with the sequencer support, then set the environment variable SCUMMVM_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 noticeable 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.

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

7.7.1 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
```

7.7.2 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
```

7.7.3 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.

7.7.4 Compressing MONSTER.SOU with MP3

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

```
compress_scumm_sou 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.

7.7.5 Compressing MONSTER.SOU with Ogg Vorbis

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

```
compress_scumm_sou --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.

7.7.6 Compressing MONSTER.SOU with Flac

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

```
compress_scumm_sou --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.

7.7.7 Compressing sfx/speech in Simon the Sorcerer and The Feeble Files

Use our compress_simon 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.

compress_simon effects	(For Acorn CD version of Simon 1)
compress_simon simon	(For Acorn CD version of Simon 1)
compress_simon effects.voc	(For DOS CD version of Simon 1)
compress_simon simon.voc	(For DOS CD version of Simon 1)
compress_simon simon.wav	(For Windows CD version of Simon 1)
compress_simon simon2.voc	(For DOS CD version of Simon 2)
compress_simon simon2.wav	(For Windows CD version of Simon 2)
compress_simon mac	(For Macintosh version of Simon 2)
compress_simon voices1.wav	(For Windows 2CD/4CD version of Feeble)
compress_simon voices2.wav	(For Windows 2CD/4CD version of Feeble)
compress_simon voices3.wav	(For Windows 4CD version of Feeble)
compress_simon voices4.wav	(For Windows 4CD version of Feeble)

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

```
compress_simon --vorbis
```

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

```
compress_simon --flac --best -b 1152
```

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

7.7.8 Compressing speech/music in Broken Sword 1

The compress_sword1 tool from the scummvm-tools package can encode music and speech to MP3 as well as Ogg Vorbis. The 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

```
compress_sword1 --vorbis
```

will compress the files using Ogg Vorbis instead of MP3.

Use

```
compress_sword1 --help
```

to get a full list of the options.

7.7.9 Compressing speech/music in Broken Sword 2

Use our `compress_sword2` 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.

```
compress_sword2 speech1.clu
compress_sword2 music1.clu
```

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

```
compress_sword2 --vorbis
```

Eventually you will have a much smaller `*.cl3` or `*.clg` file, copy this file to your game directory. 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 `compress_sword2` 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.

7.8 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 Configuration file

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\

[germansky]
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
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
```

The following keywords are recognized:

basename	string
path	string The path to where a game's data files are
autosave_period	number The seconds between autosaving (default: 300)
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, us, de, fr, it, pt, es, jp, zh, kr, se, gb, hb, cz, ru)
speech_mute	bool If true, speech is muted
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
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.
soundfont	string The SoundFont to use for MIDI playback. (Only supported by some MIDI drivers.)
native_mt32	bool If true, disable GM emulation and assume that there is a true Roland MT-32 available.
enable_gs	bool If true, enable Roland GS-specific features to enhance GM emulation. If native_mt32 is also true, the GS device will select an MT-32 map to play the correct instruments.
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)
midi_gain	number The MIDI gain (0-1000) (default: 100) (Only supported by some MIDI drivers.)
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

Broken Sword 2 adds the following non-standard keywords:

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

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

music_mute	bool	If true, music is muted
sfx_mute	bool	If true, sound effects are muted

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

music_mute	bool	If true, music is muted
sfx_mute	bool	If true, sound effects are muted

The Legend of Kyrandia adds the following non-standard keyword:

walkspeed	int	The walk speed (0-4)
-----------	-----	----------------------

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' SVN 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

ScummVM Team

Project Leaders

James Brown
Max Horn
Eugene Sandulenko

Engine Teams

SCUMM

Torbjörn Andersson
James Brown
Jonathan Gray
Max Horn
Travis Howell
Paweł Kołodziejski
Eugene Sandulenko

Codecs, iMUSE, Smush, etc.

FT INSANE, MM NES, MM C64, game detection, Herc/CGA

HE

Jonathan Gray
Travis Howell
Gregory Montoir
Eugene Sandulenko

AGI

Paweł Kołodziejski
Eugene Sandulenko

BASS

Robert Göffringmann
Oliver Kiehl
Joost Peters

Broken Sword 1

Robert Göffringmann

Broken Sword 2

Torbjörn Andersson
Jonathan Gray

Cinematique evo 1

Paweł Kołodziejski
Gregory Montoir
Eugene Sandulenko

FOTAQ

David Eriksson
Gregory Montoir
Joost Peters

Gob

Torbjörn Andersson
Sven Hesse
Willem Jan Palenstijn
Eugene Sandulenko

Kyra

Torbjörn Andersson
Oystein Eftevaag
Gregory Montoir
Johannes Schickel

VQA Player

Lure

Paul Gilbert

SAGA

Torbjörn Andersson
Andrew Kurushin
Eugene Sandulenko

Simon

Travis Howell
Oliver Kiehl

Backend Teams

Dreamcast

Marcus Comstedt

GP2X

John Willis

GP32

Won Star

Nintendo DS

Neil Millstone

PalmOS

Chris Apers

PocketPC / WinCE

Nicolas Bacca

Kostas Nakos

PlayStation 2

Robert Göffringmann

PSP (PlayStation Portable)

Joost Peters

SDL (Win/Linux/OS X/etc.)

Max Horn

Eugene Sandulenko

Asm routines, GFX layers

SymbianOS

Jurgen Braam

Lars Persson

Other subsystems**Infrastructure**

Max Horn

Backend & Engine APIs, file API, sound mixer, audiostreams, data structures, etc.

Eugene Sandulenko

GUI

Eugene Sandulenko

Johannes Schickel

Miscellaneous

Jerome Fisher

MT-32 emulator

Jochen Hoenicke

Speaker & PCjr sound support, Adlib work

Website (content)

All active team members

Documentation

Joachim Eberhard

Documentation manager

Retired Team Members

Tore Anderson

Former Debian GNU/Linux maintainer

Ralph Brorsen

Help with GUI implementation

Jamieson Christian

iMUSE, MIDI, all things musical

Vincent Hamm

Co-Founder, original CinE engine author

Rüdiger Hanke

Port: MorphOS

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

Other contributions

Packages

AmigaOS 4

Hans-Jörg Frieden
Juha Niemimäki

Atari/FreeMiNT

Keith Scroggins

BeOS

Stefan Parviainen

Debian GNU/Linux

David Weinehall

Fedora / RedHat

Willem Jan Palenstijn

Mac OS X

Max Horn

Mandriva

Dominik Scherer

MorphOS

Fabien Coeurjoly

OS/2

Paul Smedley

SlackWare

Robert Kelsen

Solaris x86

Laurent Blume

Solaris Spark

Markus Strangl

Win32

Travis Howell

Win64

Chris Gray

Websites (design)

Dobó Balázs
Yaroslav Fedevych
David Jensen
Jean Marc
Raina
Clemens Steinhuber

Website design
HTML/CSS for the website
SVG logo conversion
ScummVM logo
ScummVM forum buttons
ScummVM forum theme

Code contributions

Ori Avtalion	<i>Subtitle control options in the GUI</i>
Stuart Caie	<i>Decoders for Simon 1 Amiga data files</i>
Paolo Costabel	<i>PSP port contributions</i>
Benjamin Haisch	<i>Encoder for DXA videos</i>
Janne Huttunen	<i>V3 actor mask support, Dig/FT SMUSH audio</i>
Kovács Endre János	<i>Several fixes for Simon1</i>
Jeroen Janssen	<i>Numerous readability and bugfix patches</i>
Andreas Karlsson	<i>Initial port for EPOC/SymbianOS</i>
Claudio Matsuoka	<i>Daily Linux builds</i>
Thomas Mayer	<i>PSP port contributions</i>
Mikesch Nepomuk	<i>M11 VGA floppy patches</i>
Nicolas Noble	<i>Config file and ALSA support</i>
Quietust	<i>Sound support for Amiga SCUMM V2/V3 games, MM NES support</i>
Andreas Röver	<i>Broken Sword 1/2 MPEG2 cutscene support</i>
Edward Rudd	<i>Fixes for playing MP3 versions of M11/Loom audio</i>
Daniel Schepler	<i>Final M11 CD music support, initial Ogg Vorbis support</i>
André Souza	<i>SDL-based OpenGL renderer</i>
Tim Phillips	<i>Initial M11 CD music support</i>

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

Special thanks to

Sander Buskens	<i>For his work on the initial reversing of Monkey2</i>
Canadacow	<i>For the original MT-32 emulator</i>
Kevin Carnes	<i>For Scumm16, the basis of ScummVM's older gfx codecs</i>
Patrick Combet	<i>For the original Goblins ADL player</i>
Ivan Dubrov	<i>For contributing the initial version of the Goblins engine</i>
Jezar	<i>For his freeverb filter implementation</i>
Jim Leitterman	<i>Various info on his FM-TOWNS/Marty SCUMM ports</i>
lloyd	<i>For deep tech details about C64 Zak & MM</i>
Sarien Team	<i>Original AGI engine code</i>
Jimmi Thøgersen	<i>For ScummRev, and much obscure code/documentation</i>
Tristan	<i>For additional work on the original MT-32 emulator</i>

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 Wyrmskeep 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!

Alan Bridgman, Simon Woodroffe and everyone at Adventure Soft for sharing the source code of The Feeble Files with us.