

NAME

mscore-snapshot, **musescore-snapshot** — MuseScore 3 Snapshot package (3.0.0 unstable) sheet music editor

SYNOPSIS

```
mscore-snapshot [-deFfhIiLmnOPRstvw] [-a | --use-audio driver]
                [-b | --bitrate bitrate] [-c | --config-folder pathname]
                [-D | --monitor-resolution DPI]
                [-E | --install-extension extension file]
                [-j | --job file.json] [-M | --midi-operations file]
                [-o | --export-to file] [-p | --plugin name]
                [-r | --image-resolution DPI] [-S | --style style]
                [-T | --trim-image margin] [-x | --gui-scaling factor]
                [--debug] [--diff] [--dump-midi-in] [--dump-midi-out]
                [--experimental] [--export-score-parts]
                [--factory-settings] [--force] [--help] [--layout-debug]
                [--load-icons] [--long-version] [--new-score]
                [--no-fallback-font] [--no-midi] [--no-synthesizer]
                [--no-webview] [--raw-diff] [--revert-settings]
                [--run-test-script] [--score-media] [--template-mode]
                [--test-mode] [--version] [file ...]
```

DESCRIPTION

MuseScore is a Free and Open Source WYSIWYG cross-platform multi-lingual music composition and notation software, released under the GNU General Public Licence (GPLv2).

Running **mscore-snapshot** without any extra options launches the full graphical MuseScore program and opens any files specified on the command line.

The options are as follows:

- a** | **--use-audio** *driver*
Use audio driver: one of **jack**, **alsa**, **portaudio**, **pulse**
- b** | **--bitrate** *bitrate*
Set MP3 output bitrate in kbit/s
- c** | **--config-folder** *pathname*
Override configuration and settings directory
- D** | **--monitor-resolution** *DPI*
Specify monitor resolution (override autodetection)
- d** | **--debug**
Start MuseScore in debug mode
- E** | **--install-extension** *extension file*
Install an extension file; soundfonts are loaded by default unless **-e** is also specified
- e** | **--experimental**
Enable experimental features, such as layers
- F** | **--factory-settings**
Use only the standard built-in presets (“factory settings”) and delete user preferences; compare with the **-R** option

- f** | **--force**
Ignore score corruption and version mismatch warnings in “converter mode”
- h** | **--help**
Display an overview of invocation instructions
- I** | **--dump-midi-in**
Display all MIDI input on the console
- i** | **--load-icons**
Load icons from the filesystem; useful if you want to edit the MuseScore icons and preview the changes
- j** | **--job** *file.json*
Process a conversion job (see **EXAMPLES** below)
- L** | **--layout-debug**
Start MuseScore in layout debug mode
- M** | **--midi-operations** *file*
Specify MIDI import operations file
- m** | **--no-midi**
Disable MIDI input
- n** | **--new-score**
Start with the New Score wizard regardless whether it’s enabled or disabled in the user preferences
- O** | **--dump-midi-out**
Display all MIDI output on the console
- o** | **--export-to** *file*
Export the given (or currently opened) file to the specified output *file*. The file type depends on the extension of the filename given. This option switches to “converter mode” and avoids the graphical user interface.
- P** | **--export-score-parts**
When converting to PDF with the **-o** option, append each part’s pages to the created PDF file. If the score has no parts, all default parts will temporarily be generated automatically.
- p** | **--plugin** *name*
Execute the named plugin
- R** | **--revert-settings**
Use only the standard built-in presets (“factory settings”) but do not delete user preferences; compare with the **-F** option
- r** | **--image-resolution** *DPI*
Set image resolution for conversion to PNG files; default: 300
- S** | **--style** *style*
Load a style file first; useful for use with the **-o** option
- s** | **--no-synthesizer**
Disable the integrated software synthesiser
- T** | **--trim-image** *margin*
Trim exported PNG and SVG images to remove whitespace surrounding the score. The specified *margin*, in pixels, will be retained (use 0 for a tightly cropped image). When exporting to SVG, this option only works with single-page scores.

- t** | **--test-mode**
Set test mode flag for all files
- v** | **--version**
Display the name and version of the application without starting the graphical user interface
- w** | **--no-webview**
Disable the web view component in the Start Centre
- x** | **--gui-scaling** *factor*
Scale the score display and other GUI elements by the specified *factor*; intended for use with high-resolution displays
- diff**
Print a conditioned diff between the given scores
- long-version**
Display the full name, version and git revision of the application without starting the graphical user interface
- no-fallback-font**
Don't use Bravura as fallback musical font
- raw-diff**
Print a raw diff between the given scores
- run-test-script**
Run script tests listed in the command line arguments
- score-media**
Export all media for a given score as a single JSON document to stdout
- template-mode**
Save files in template mode (e.g. without page sizes)

MuseScore supports the automatic Qt command line options (see below).

Batch conversion job JSON format

The argument to the `-j` option must be the pathname of a file comprised of a valid JSON document honouring the following specification:

- The top-level element must be a JSONArray, which may be empty.
- Each array element must be a JSONObject with the following keys:
 - `in` Value is the name of the input file (score to convert), as JSONString.
 - `plugin` Value is the filename of a plugin (with the `.qml` extension), which will be read from either the global or per-user plugin path and executed before the conversion output happens, as JSONString. Optional, but at least one of `plugin` and `out` *must* be given.
 - `out` Value is the conversion output target, as defined below. Optional, but at least one of `plugin` and `out` *must* be given.
- The conversion output target may be a filename (with extension, which decided the format to convert to), as JSONString.
- The conversion output target may be a JSONArray of filenames as JSONString, as above, which will cause the score to be written to multiple output files (in multiple output formats) sequentially, without being closed, re-opened and re-processed in between.

- If the conversion output target is a JSONArray, one or more of its elements may also be, each, a JSONArray of two JSONStrings (called first and second half in the following description). This will cause part extraction: for each such two-tuple, all extant parts of the score will be saved *individually*, with filenames being composed by concatenating the first half, the name (title) of the part, and the second half. The resulting string must be a valid filename (with extension, determining the output format). If a score has no parts (excerpts) defined, this will be silently ignored without error.

- Valid file extensions for output are:

`flac` Free Lossless Audio Codec (compressed audio)

`metajson` various score metadata (JSON)

`mid` standard MIDI file

`mlog` internal file sanity check log (JSON)

`mp3` MPEG Layer III (lossy compressed audio)

`mpos` measure positions (XML)

`mscx` uncompressed MuseScore file

`mscz` compressed MuseScore file

`musicxml` uncompressed MusicXML file

`mxl` compressed MusicXML file

`ogg` OGG Vorbis (lossy compressed audio)

`pdf` portable document file (print)

`png` portable network graphics (image)

Individual files, one per score page, with a hyphen-minus followed by the page number placed before the file extension, will be generated.

`spos` segment positions (XML)

`svg` scalable vector graphics (image)

`wav` RIFF Waveform (uncompressed audio)

`xml` uncompressed MusicXML file

See below for an example.

ENVIRONMENT

`SKIP_LIBJACK`

Set this (the value does not matter) to skip initialisation of the JACK Audio Connection Kit library, in case it causes trouble.

`XDG_CONFIG_HOME`

User configuration location; defaults to `~/ .config` if unset.

`XDG_DATA_HOME`

User data location; defaults to `~/ .local/share` if unset.

Note that MuseScore also supports the normal Qt environment variables such as `QT_QPA_GENERIC_PLUGINS`, `QT_QPA_PLATFORM`, `QT_QPA_PLATFORMTHEME`, `QT_QPA_PLATFORM_PLUGIN_PATH`, `QT_STYLE_OVERRIDE`, `DISPLAY`, etc.

FILES

`/usr/share/mscore-snapshot-3.0/` contains the application support data (demos, instruments, localisation, system-wide plugins, soundfonts, styles, chords, templates and wallpapers). In the Debian packages, system-wide soundfonts are installed into `/usr/share/sounds/sf2/`, `/usr/share/sounds/sf3/` or `/usr/share/sounds/sfz/`, respectively, instead.

The per-user data (extensions, plugins, soundfonts, styles, templates) and files (images, scores) are normally installed into subdirectories under `~/MuseScore3/` but may be changed in the configuration. Note that snapshot, alpha and beta versions use `MuseScore3Development` instead of `MuseScore3` in all of these paths.

`$XDG_CONFIG_HOME/MuseScore/MuseScore3.ini` contains the user preferences, list of recently used files and their locations, window sizes and positions, etc. See above for development version paths.

`$XDG_DATA_HOME/data/MuseScore/MuseScore3/` contains updated localisation files downloaded from within the program, plugin information, cached scores, credentials for the `musescore.com` community site, session information, synthesiser settings, custom key and time signatures and shortcuts. See above for development version paths.

EXAMPLES

Convert a score to PDF from the command line:

```
mscore-snapshot -o 'My Score.pdf' 'My Score.mscz'
```

Run a batch job converting multiple documents:

```
mscore-snapshot -j job.json
```

This requires the file `job.json` in the current working directory to have content similar to the following:

```
[
  {
    "in": "Reunion.mscz",
    "out": "Reunion-coloured.pdf",
    "plugin": "colornotes.qml"
  },
  {
    "in": "Reunion.mscz",
    "out": [
      "Reunion.pdf",
      [ "Reunion (part for ", ").pdf" ],
      "Reunion.musicxml",
      "Reunion.mid"
    ]
  },
  {
    "in": "Piece with excerpts.mscz",
    "out": [
      "Piece with excerpts (Partitura).pdf",
      [ "Piece with excerpts (part for ", ").pdf" ],
      "Piece with excerpts.mid"
    ]
  }
]
```

The last part of the job would, for example, cause files like “Piece with excerpts (part for Violin) .pdf” to be generated alongside the conductor’s partitura and a MIDI file with the full orchestra sound, whereas the equivalent part of the Reunion conversion will be silently ignored (because the Reunion piece (a MuseScore demo) has no excerpts defined).

DIAGNOSTICS

The **mscore-snapshot** utility exits 0 on success, and >0 if an error occurs.

SEE ALSO

`fluidsynth(1)`, `midicsv(1)`, `timidity(1)`, `qtoptions(7)`

<https://musescore.org/handbook>
Online Handbook, full user manual

<https://musescore.org/forum>
Support Forum

<https://musescore.org/handbook/command-line-options-0>
Further documentation of command line options

<https://musescore.org/handbook/revert-factory-settings-0>
Reverting to factory settings (troubleshooting)

<https://musescore.org/project/issues>
Project Issue Tracker

Please check first to if the bug you’re encountering has already been reported. If you just need help with something, then please use the support forum (see above) instead.

<http://doc.qt.io/qt-5/qguiapplication.html#supported-command-line-options>
Documentation of automatic Qt command line options

STANDARDS

MuseScore attempts to implement the following standards:

- MusicXML 3.1 (score interchange format)
- SF2 (SoundFont 2.01)
- SF3 (SoundFont with OGG Vorbis-compressed samples)
- SFZ (Sforzato soundfont)
- SMuFL (Standard Music Font Layout 1.20)

HISTORY

MuseScore was split off the MuseE sequencer in 2002 and has since become the foremost Open Source notation software.

AUTHORS

MuseScore is developed by Werner Schweer and others.

This manual page was written by mirabilos <tg@debian.org>.

CAVEATS

The automatic Qt command line options are removed from the argument vector before the application has a chance at option processing; this means that an invocation like

`mscore-snapshot -S -reverse`

has no chance at working because the **`-reverse`** is removed by Qt first.

BUGS

MuseScore does not honour `/etc/papersize`.

Saving a part (excerpt) as MIDI crashes the application.

Probably some more; check the project's bug tracker (cf. **SEE ALSO**).