Abstract

Haruna is an open source media player built with Qt/QML and libmpv.
Chapter 1

Settings

Explains some of the more complex settings of Haruna.

1.1

1.2 General

1.2.1 File dialog location

The folder the file dialog will load when used to open a file. If empty it will remember the last folder from which a file was opened.

1.2.2 Maximum recent files

How many recent files to save and display in the recent files menu.

1.2.3 Remember window size and position

Restores the size and position the window had when it was closed. On Wayland only size is restored.

1.2.4 Breeze icon theme

When checked the application uses the Breeze icon theme. When unchecked the application uses the system icon theme. Doesn’t always work as expected.

1.2.5 GUI style

Allows to change the style of Qt Quick Controls (buttons, checkboxes etc.), to styles available on the system.
1.3  Playback

1.3.1  Seek steps

How much to seek when triggering the corresponding action. Seek mode is exact

1.3.2  Open last played file on startup

On startup it opens the file that was playing when the application was closed.

1.3.3  Seek to last playback position

When opening a file that was played before it will seek to the position it was when playback
stopped.

1.3.4  Start playing

This applies only when resuming playback (seeking to last playback position) of a file that was
played before.

1.3.5  Pause on minimize

Pauses the player while the window is minimized, playback resumes when restored.

1.3.6  Hardware decoding

If you enable hardware decoding it’s recommended to use the auto-safe or auto options.
You can read more about each option here: https://mpv.io/manual/stable/#options-hwdec.
If you want to use vaapi on X11, you must set the environment variable
QT_XCB_GL_INTEGRATION to xcb_egl.

1.3.7  Remember time position

Determines when to save and restore the position of previously played files.
First field controls for which files to save the position:

- -1: feature is disabled
- 0: save for all files
- 1-9999: save for files longer than the set value, in minutes

Second field controls how often to save the position.
1.3.8 Skip chapters

When enabled it automatically skips chapters containing certain words/characters.
The words/characters to skip are defined as a comma separated string.
The match is not exact, meaning *op* will match words containing it like *opening*.

1.3.9 Youtube-dl format selection

Haruna uses youtube-dl to play online videos. Some sites offer multiple versions of the
video/audio (different codecs, resolution). This setting allows you to configure what version
blob/master/README.md#format-selection

1.4 Video

1.4.1 Screenshot Template

https://mpv.io/manual/stable/#screenshot

Specify the filename template used to save screenshots. The template specifies the filename
without file extension, and can contain format specifiers, which will be substituted when tak-
ing a screenshot. By default, the template is mpv-shot%n, which results in filenames like mpv-
shot0012.png for example.

The template can start with a relative or absolute path, in order to specify a directory location
where screenshots should be saved.

If the final screenshot filename points to an already existing file, the file will not be overwritten.
The screenshot will either not be saved, or if the template contains %n, saved using a different,
newly generated filename.

Allowed format specifiers:

- %[#][0X]n
  A sequence number, padded with zeros to length X (default: 04). E.g. passing the format %04n
  will yield 0012 on the 12th screenshot. The number is incremented every time a screenshot is
taken or if the file already exists. The length X must be in the range 0-9. With the optional #
sign, mpv will use the lowest available number. For example, if you take three screenshots--
0001, 0002, 0003--and delete the first two, the next two screenshots will not be 0004 and 0005,
but 0001 and 0002 again.

- %f
  Filename of the currently played video.

- %F
  Same as %f, but strip the file extension, including the dot.

- %x
  Directory path of the currently played video. If the video is not on the filesystem (but e.g.
  http:///), this expands to an empty string.

- %X{fallback}
  Same as %x, but if the video file is not on the filesystem, return the fallback string inside the
  [...].
• %p
  Current playback time, in the same format as used in the OSD. The result is a string of the form
  "HH:MM:SS". For example, if the video is at the time position 5 minutes and 34 seconds, %p
  will be replaced with "00:05:34".

• %P
  Similar to %p, but extended with the playback time in milliseconds. It is formatted as
  "HH:MM:SS.mmm", with "mmm" being the millisecond part of the playback time.
  Note
  This is a simple way for getting unique per-frame timestamps. (Frame numbers would be more
  intuitive, but are not easily implementable because container formats usually use timestamps
  for identifying frames.)

• %wX
  Specify the current playback time using the format string X. %p is like %wH:%wM:%wS, and
  %P is like %wH:%wM:%wS.%wT.
  Valid format specifiers:
  – %wH
    hour (padded with 0 to two digits)
  – %wh
    hour (not padded)
  – %wM
    minutes (00-59)
  – %wm
    total minutes (includes hours, unlike %wM)
  – %wS
    seconds (00-59)
  – %ws
    total seconds (includes hours and minutes)
  – %wf
    like %ws, but as float
  – %wT
    milliseconds (000-999)

• %tX
  Specify the current local date/time using the format X. This format specifier uses the UNIX
  strftime() function internally, and inserts the result of passing "%X" to strftime. For example,
  %tm will insert the number of the current month as a number. You have to use multiple %tX
  specifiers to build a full date/time string.

• %{prop[:fallback text]}
  Insert the value of the input property ‘prop’. E.g. %{filename} is the same as %f. If the property
  does not exist or is not available, an error text is inserted, unless a fallback is specified.

• %%
  Replaced with the % character itself.

1.5 Custom commands

1.5.1 Command

What mpv command to run, see MPV Commands for a list of available commands.
Not all commands will work as some are specific for mpv.
Most useful are the commands to manipulate properties, like set, add, cycle.

*set*: sets the value of a property

```
set volume-max 150.
```

*add*: increases/decreases the value of a property

```
add volume 15, increases volume by 15. To decrease the value use
add volume -15.
```

*cycle*: cycle through the values of a property

```
cycle aid up.
```

### 1.5.2 OSD message

An OSD message to display when triggering an action type custom command. Use `$\{property_name\}` to display the value of a property `Volume: $\{volume\}`.

### 1.5.3 Keyboard shortcut

The command will be triggered with a shortcut. The shortcut can be set after saving.

### 1.5.4 Run at startup

The command will be run at application startup. Startup commands can be toggled, meaning you can choose whether they are set on startup or not, this is available after saving.
Chapter 2

Credits and License

Haruna
Program copyright 2019-2021 George Florea Bănuș;
Contributors:

• George Florea Bănuș; georgefb899@gmail.com

Documentation Copyright (c) 2019-2021 George Florea Bănuș; georgefb899@gmail.com

This documentation is licensed under the terms of the Creative Commons Attribution-ShareAlike 4.0 International.
This program is licensed under the terms of the GNU General Public License.