NAME

runmix — Automate music mixing with pianod(1).

SYNOPSIS

runmix [**-cniwL**] [*mixplan*]

DESCRIPTION

runmix evaluates and runs a mixplan. A mixplan is a simple declarative program that describes what playlist(s) to mix at particular times. Each line features a time in 24-hour format followed by a list of playlists, separated by plus (+) signs. Times must be in chronological order or **runmix** considers it an error.

runmix allows 2 pseudo genres: start requests the player start playback at the specified time, and stop requests stopping at the specified time.

runmix also allows arbitrary pianod commands. These must occur at the start of the plan, before any times. Use the keyword configure instead of a time, followed by the full command line.

Comments are formed by hash/pound sign (#this is a comment) or using double-slash, //as in this form.

runmix assumes the plan starts today. If the first time is later in the day, it schedules playback to start then. If, however, there are mix definitions prior to the present time of day, they are ignored except for the single playlist set prior to the present time, which is selected immediately. Times may wrap past midnight once.

runmix uses at(1) to schedule playlist changes according to the mixplan.

OPTIONS

-n Evaluate the mixplan without running it.

-c Cancel mixing. This clears all future playlist changes.

-i Interactive mode. **runmix** will prompt the user to select a mix, using either Finder, X, or terminal dialogs; if nothing better is available, it will simply prompt in its terminal or window.

If a mix is already scheduled or in progress, **runmix** lets the user choose between cancelling the existing mix, choosing a new one, or no action.

-w Wait. If the mixplan specifies playlists that are not available, **runmix** waits until another source becomes available, then retries. There are some hard-coded timeouts to prevent indefinite delays. Consider late binding instead.

-L Late binding. Playlists are matched when they are due to be enabled instead of when runmix is invoked. This disables mixplan playlist validation since sources and playlists are expected to change.

Other options are for testing purposes, and may be changed or removed in future versions.

PLATFORM SPECIFIC SUPPORT

MAC OS X Finder will launch shell scripts in a Terminal window if they have the extension command. **runmix** recognizes this and enters interactive mode if there are no parameters. Note **runmix** must be natively named runmix.command; a symbolic link will not work.

When using **runmix** on OS X in any form, at(1) job processing must be enabled:

```
sudo launchctl load -w
/System/Library/LaunchDaemons/com.apple.atrun.plist
```

Linux Linux users may need to install at(1). Depending on your distribution, this may be helpful:

```
sudo apt-get --no-install-recommends install at
```

ENVIRONMENT

`PIANOD_HOST`
: The `pianod` host to connect to.

`PIANOD_PORT`
: The port at which to connect to `pianod`.

`PIANOD_USER`
: The username to authenticate with.

`PIANOD_PASSWORD`
: The password with which to authenticate.

FILES

`~/Music/Mixes` Directory in which mixes are expected to be found. If the directory is not found, tries ~/Music, ~, and so forth back to /.

RETURN VALUES

0 if the mixplan is valid

1 if the mixplan is not valid

SEE ALSO

`at`(1), `dialog`(1), `piano`(1), `pianod`(1) `whiptail`(1), `xdialog`(1),

Example mixplans at deviousfish.com/pianod.

BUGS

Mixplans crossing two midnights won't work as expected.

`configure` directives are issued to the server even when the -n option is given.

`runmix` no longer interferes with unrelated jobs in the `at`(1) queue.

AUTHOR

Perette Barella <perette@deviousfish.com>.

COPYRIGHT