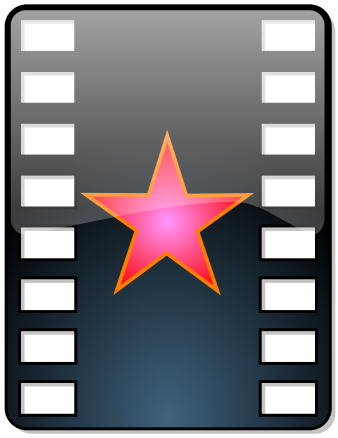
The KMPlayer Handbook

Koos Vriezen



Contents

1 Introduction **5**

2 The user interface **6**

2.1 Playlist window 6

2.1.1 Persistent Playlists category 6

2.1.1.1 Storage format 6

2.2 Direct XML editing 7

3 Settings **8**

3.1 Intro/exit animation 8

4 The backend players **9**

4.1 MPlayer 9

4.1.1 Trouble shooting 9

4.2 Phonon 9

4.2.1 Trouble shooting 9

4.3 Browser plugin 10

4.3.1 Trouble shooting 10

4.4 Force backend player for a mimetype 10

5 The playable sources **11**

5.1 TV 11

5.2 VDR 12

5.3 Commandline 12

6 Questions, Answers, and Tips **14**

7 Credits and License **15**

Abstract

KMPlayer is an application that is a KDE frontend to MPlayer and Phonon.

Chapter 1

Introduction

KMPlayer is a simple frontend to MPlayer, Phonon and FFMpeg. You can use it for watching all the movie file formats that MPlayer, or Phonon supports as well as watching from DVD, VCD or TV and camera.

The documentation for KMPlayer was not finished when KDE was installed on this computer.

If you need more help, please check The KDE Website for updates, or by submitting your question to The KDE User Mailing list.

The KDE Team

Chapter 2

The user interface

The application window is made of so called dockable windows. There is always the central view window. Optional there is the playlist window. And finally there is an info window, that just shows itself when some information is there to been shown.

2.1 Playlist window

The playlist window has as of this writing five categories, current item(s), **Optical Disks**, **Television**, **Persistent Playlists** and **Most Recent**. When clicked on a category's text, it will show the contents below this line. Of course there has to be contents, eg. see TV settings for adding content for the **Television** category. The **Most Recent** category will fill itself when playing locale media files or network URL. This category can hold 60 items of which the last 50 are under the **More ..** branch.

2.1.1 Persistent Playlists category

It's possible to store your favorite links in the **Bookmark** menu, but those are single items. If you want to create a list of items that play after each other, than you should use the **Persistent Playlists** category.

The easiest way to add an item in this list is to drag it from **Most Recent** and drop it on **Persistent Playlists**. As you will see, when dropped, a pop-up menu appears where you can choose to add to this list or create a new group. Top level items behave like a bookmark item, but items in groups will play the whole group wherein they reside. Items in this category can easily be moved, copied, added to a new group or deleted just by dragging this item to another place in this category. Additionally there are **Delete item**, **Move up** and **Move down** items in the context menu (when clicking with the right mouse button on an item).

Of course you can add groups to groups to get a deeper hierarchy. When clicked on an item, its group and all its child groups will be copied to the current category and playings starts at the clicked item position.

2.1.1.1 Storage format

The data is stored in `~/.kde/share/apps/kmplayer/playlist.xml`, and its format is like this example. See here for manipulating this data directly.

```
<playlist>
    <group title="feeds">
        <item url="http://www.lugradio.org/episodes.rss"/>
        <item url="http://science.nasa.gov/podcast.xml"/>
        <item url="http://www.theworld.org/rss/tech.xml"/>
    </group>
    <group title="iradio">
        <group title="sunny">
            <item mplayeropts="-cache 256" url="http://www.swissgroove.ch/ ↩
                listen.m3u"/>
            <item url="http://www.boombasticradio.com/shoutcast/64.pls"/>
        </group>
        <group title="rainy">
            <item url="http://www.radioparadise.com/musiclinks/rp_64.m3u"/>
            <item url="http://www.woxy.com/stream/wma64.asx"/>
            <item url="http://radionigel.com/winmedia.asx"/>
            <item url="rtsp://livestreams.omroep.nl/live/vpro/ ↩
                vpro3voor12radioalt.rm"/>
        </group>
    </group>
</playlist>
```

Also note that for MPlayer custom options can be set using the `mplayeropts` argument. KMPlayer searches for this option upwards in the tree, so you can set it for a whole group and override it for a child group or item.

2.2 Direct XML editing

The current, **Television** and **Persistent Playlists** categories can also be manipulated by editing their XML storage. Of course one can do that when KMPlayer is not running and editing the various files in `~/.kde/share/apps/kmplayer/` , but also in the application itself. Simply click on a category label, then choose **View** and next **Edit mode**. You can also click on a child branch and will then see only the XML contents of the branch.

After you have edited some XML, click on the **Sync with playlist** and either deselect the **Edit mode** item or click on another catergory.

> CAUTION
> All changes are lost when leaving edit mode or selecting another branch without syncing with playlist.

Because of the lazyness of KMPlayer's author(s), some features can only be enabled by editing the XML .

Chapter 3

Settings

3.1 Intro/exit animation

Disabling the starting/ending animation can be configured by editing `~/.kde/share/config/kmplayerrc` and modify

```
[General Options]
  No Intro=0
```

the above option to '1'. Just add this option if it's not there yet.

It's also possible to set an alternative animation. KMPlayer will look at resp. `~/.kde/share/apps/kmplayer/intro.xml` and `~/.kde/share/apps/kmplayer/exit.xml`. The format has to be a supported XML file.

The intro animation doesn't have to be a short one, because it will automatically end when another source is opened. Of course for the exit animation this would be a bit annoying, however the user can always end the application twice.

A simple example of an intro, showing your favorite photo can be

```
<smil>
  <body>
    <img dur="indefinite" src="file:///home/koos/mum.gif" fit="meet"/>
  </body>
</smil>
```

In KMPlayer's source package, in the `tests/` are some examples of SMIL that KMPlayer supports.

Chapter 4

The backend players

KMPlayer uses external programs to do the actual playing.

4.1 MPlayer

There is a configuration tab for MPlayer in KMPlayer's config dialog. Here you can set MPlayer's path and it cache size for network stream, among other things.

This player configuration name is *mplayer*.

4.1.1 Trouble shooting

The best way to debug this backend is to start KMPlayer from a console application like konsole. Then start a movie with KMPlayer and see the exact commandline that is used to start mplayer in the console application. Now copy/paste that to another console, remove the '-wid 12344' options, and make it play. Probably it's just choosing the right -ao/-vo params. Also make sure mplayer doesn't start default with a GUI.

Another debugging tool is the 'console' button in KMPlayer's toolbar. Here you can see the output from mplayer.

4.2 Phonon

Phonon is the KDE desktop wide multimedia framework and can be configured in System Settings.

This player configuration name is *phonon*.

4.2.1 Trouble shooting

Basically the same way as trouble shooting MPlayer, with the application name of `kphononplayer`.

4.3 Browser plugin

There is no configuration tab for this player in KMPlayer's config dialog. Everything must be configured by manually editing the `~/.kde/share/config/kmplayerrc` like with configuring the backend per mimetype. This is an example how to let flash be played with the swfdec-mozilla plugin:

```
[application/x-shockwave-flash]
player=npp
plugin=/usr/lib/mozilla/plugins/libswfdecmozilla.so
```

You can play flash inside KMPlayer, actually you can even paste the special *embed* field for YouTube in the playlist XML.

In order to let KMPlayer play flash in HTML pages, you should make the KMPlayer plugin the service provider for the *application/x-shockwave-flash* mimetype in KDE's **File Associations** configuration dialog.

As in the above example, this player configuration name is *npp*.

4.3.1 Trouble shooting

Start Konqueror or KMPlayer in a console application and look for errors about loading the browser plugin library. The process started is called `knpplayer`. If that doesn't give any errors, check the backend output capturing inside KMPlayer (**View**, **Console**, or **Console** from the popup menu).

4.4 Force backend player for a mimetype

To configure to always use a given backend for a particular mimetype (if detected of course) can be done by manually editing the `~/.kde/share/config/kmplayerrc`. For example, let phonon play all ogg files:

```
[audio/ogg]
player=phonon
```

Chapter 5

The playable sources

KMPlayer can play from various sources. These sources are listed in the Source menu.

5.1 TV

KMPlayer can play from TV cards using MPlayer or XVideo (using kxvplayer). Before you can watch TV, you have to configure which device to use and the TV channels. You can use the scanner from the configure dialog for this or edit the config file manually.

The config file for TV settings is a XML file, located `~/.kde/share/apps/kmplayer/tv.xml`. See here for manipulating this data directly. The XML format is like this example:

```
<tvdevices>
    <device path="/dev/video0" width="320" height="240" name="BT878 video ( ↩
        Hauppauge (bt878))" audio=""
              minwidth="48" minheight="32" maxwidth="924" maxheight="576"  ↩
                  playback="0" xvport="240">
        <input name="Television" id="0" tuner="1" norm="PAL" xvenc="0">
            <channel name="Ned1" frequency="216"/>
            <channel name="VCR" frequency="594.90"/>
        </input>
        <input name="Composite1" id="1" xvenc="7"/>
        <input name="S-Video" id="2" xvenc="14"/>
        <input name="Composite3" id="3"/>
    </device>
    <device path="/dev/video1" width="640" height="480" name="Philips 740  ↩
        webcam" audio=""
              minwidth="160" minheight="120" maxwidth="640" maxheight="480" ↩
                  playback="1">
        <input name="Webcam" id="0"/>
    </device>
</tvdevices>
```

Note that 'input' elements having channels, should have the 'tuner' attribute set to '1'.

XVideo only uses the 'xvport' and 'xvenc' attributes and they can only be set by manually editing this file. The values of these two attributes can be retrieved by running 'kxvplayer' in a terminal application like 'konsole'. For every port, 'kxvplayer' will output lines like below and the numbers can be set for the attributes mentioned.

```
    xvport 240
    ....
    encoding: 0 PAL-television
    encoding: 1 NTSC-television
    encoding: 2 SECAM-television
    ....
```

Also width and height attributes are ignored by the XVideo player. This player gets the sizes from the X server. One of the nice things of XVideo is that scaling is done in hardware, by the video card, so fullscreen support without CPU usage and without resolution change of your monitor. Also no annoying flicker when another window overlaps this video window.

See the VDR section for more information about using XVideo.

5.2 VDR

To configure your VDR settings in KMPlayer, see the Configure panel->Source->VDR->XVideo port panel. There should be ports detected, from which you should select one. You have to figure out what encoding you should use (eg. for western europe, that is PAL) and likely the correct one has a name with 'dvb' in it.

For instance, I have a TV card, a DVB-S card and a webcam plugged into my system. With these, my settings are that there are three Ports. The first one has a long list of NTSC(-JP)/PAL(-M)/SECAM entries all saying something about television/composite/svideo (my TV card). The second one has NTSC-dvb/PAL-dvb/SECAM-dvb (my DVB-S card). And finally the third has only ntsc/pal (probably my webcam). I should therefore configure (just select it) the second port and second entry (PAL-dvb).

Because VDR support from KMPlayer can only use XVideo, of course XVideo should work (and I believe for video devices, this way of viewing only works on linux). Also this only works on your first display (:0.0). So make sure 'videoforlinux' extension works with the X server. For the XFree86 server, therefore in its configuration (`/etc/X11/XF86Config`) it should have

```
Section "Module"
     ....
     Load    "v4l"
EndSection
```

Unfortunately, when ever you update your video driver, you have to reconfigure this setting.

5.3 Commandline

Internally for the sources, KMPlayer uses distinct names. At the time of this writing these are ″dvdnavsource″, ″dvdsource″, ″exitsource″, ″introsource″, ″pipesource″, ″tvscanner″, ″tvsource″, ″urlsource″, ″vcdsource″ and ″vdrsource″. When activated, they mostly starts playing the list of playable items.

You can activate a particular source with a `kmplayer://` type URL. Just set the host part as the source name and optional path part as extra arguments, so for a channel for the ″vdrsource″ this could look like `kmplayer://vdrsource/22 MTV 2`. Eg. this script starts KMPlayer using ″dvdnavsource″ and using D-Bus to play fullscreen.

```
/bin/bash
kmplayer kmplayer://dvdnavsource/ &
PID=$!
sleep 2
qdbus org.kde.kmplayer-$PID /kmplayer/MainWindow_1 showFullScreen
```

For some sources this will not do much because they are not meant to be for normal playing. For ″pipesource″ this is disabled because that would be a bad security hole of course.

Chapter 6

Questions, Answers, and Tips

Frequently asked questions

1. *Any hints to make my CDROM less noisy when playing from it (not VCD://)?*

   Sure, try **hdparm -E 8 /dev/cdrom** as root after mounting the device. But be sure to double check `/dev/cdrom` pointing to your CDROM drive.

Chapter 7

Credits and License

KMPlayer copyright 2002,2003 Koos Vriezen

This documentation is licensed under the terms of the GNU Free Documentation License.
This program is licensed under the terms of the GNU General Public License.