

Kodi Config/Tips

Prevent beginning 1 second of video to be skipped

System → Display → Pause during refresh rate change, set to 1 second.

Create directory structure for movies

If movies are all in one directory, the following command can be used to move them into individual directories so kodi can properly parse them (replace .mkv with .avi or .mp4 as well):

```
cd /path/to/your/movies/files/  
find . -name "*.mkv" -exec sh -c 'mkdir "${1%.*}" ; mv "$1" "${1%.*}" ' _ {}  
\;
```

Caching/Buffering settings

In /home/kodi/.kodi/userdata/advancedsettings.xml the cache section:

```
<advancedsettings>  
  <cache>  
    <!-- The three settings will go in this space, between the two  
cache tags. -->  
  </cache>  
</advancedsettings>
```

<buffermode>2</buffermode>

0 Buffer all internet filesystems (like "2" but additionally also ftp, webdav, etc.) (default)

1 Buffer all filesystems, both internet and local

2 Only buffer true internet filesystems (streams) (http, etc.)

3 No buffer

<memorysize>20971520</memorysize>

- Value in bytes of how much RAM is used for cache. 20MB in this example, but kodi requires 60MB free ram for this (3x the setting).

- Value of 0 will use local disk memory for caching. This is slower and increases wear and tear of the HDD, but is helpful on low memory devices.

<readfactor>10</readfactor>

Fill-rate of the cache. Default is 4. This is the speed kodi loads the data into the cache. I.e. when loading a raw blueray file at 36Mbit/s, it attempts to fill the cache 4 times as fast and thus requires 144Mbit/s bandwidth to do so.

Kodi 18 Theme

Kodi 18 Estuary Modv2

<https://forum.kodi.tv/showthread.php?tid=306757>

changelog:

<https://raw.githubusercontent.com/Guilouz/repository.guilouz/master/skin.estuary.modv2/changelog.txt>

Addons

An overview of many (all?) kodi addons can be found here:

http://kodi.wiki/view/Category:All_add-ons

or directly at

<http://mirrors.kodi.tv/addons/>

Stinger Notification AddOn

Latest version is 1.0.3, not in Kodi repository. Download manually on device (or download and transfer), then install using zip file from untrusted sources:

wget

<https://github.com/rmrector/repository.rector.stuff/raw/master/latest/service.stinger.notification-latest.zip>

(check <https://forum.kodi.tv/showthread.php?tid=254004> - there may be a newer version)

Once installed, edit the settings file and enable set visibility of the action to true (last entry).

```
sudo vi
/home/kodi/.kodi/addons/service.stinger.notification/resources/settings.xml
#restart kodi using
sudo service kodi restart
```

install the 'Universal Movie Scraper' plugin and set it to download additional tags from the moviedb. Leave the remaining settings as they are, however, you might want to set the title/naming engine to TMDb rather than IMDB. I noticed IMDB returning making of's rather than actual movie entries by mistake which wasn't the case with TMDb as main source.

Edit the content of your movie sources in kodi to use the universal scraper instead of the moviedb scraper. Don't update the sources or you'll lose all watched status.

Now go to the scraper addon settings in advanced and let it update the tags.

Video Extras

This addon enables adding Bonus features/extras/etc to movie/video files and adds an option to exclude them from the library. Extras need to be saved in a subfolder of the main movie/video rather than a standalone folder.

<http://kodi.wiki/view/Add-on:VideoExtras>

Watchdog

This addon scans the configured media sources automatically for changes and updates the library. This is very useful if there is a download folder or 'new' folder. For this to work, the kodi device needs to be running when the change happens.

<http://kodi.wiki/view/Add-on:Watchdog>

NFO Watchstate Updater

This addon creates and updates nfo files with watched state as soon as the kodi database was updated. This is useful when multiple kodi instances access the same files or when video folders are moved on the filesystem.

<https://www.tvaddons.ag/kodi-addons/show/service.nfo.watchedstate.updater/>

An alternative to this is to do it manually:

Note: This will create and overwrite nfo files which may otherwise be in the folder already, the nfo files will have the same name as the video file 1.) Go to Kodi system settings → Video → Library 2.) Enable advanced settings if not already enabled 3.) Export library → Multiple Files → Overwrite existing files http://kodi.wiki/view/Videos_Settings

In advancedsettings.xml, make sure that importwatchedstate for libraries is enabled (see config note above)

download the zip file from the addons page, then use adb to push it to the device:

```
adb connect 192.168.1.10
adb push service.nfo.watchedstate.updater-1.0.0.zip /sdcard/
```

In Kodi, go to Systems → Settings → Add-ons → Install from Zip file → select the watchedstate zip file

Easy Advanced Settings

With this addon you can edit the advanced settings directly in kodi rather than having to create a file and transfer it awkwardly to the device.

<https://www.tvaddons.ag/kodi-addons/show/plugin.program.advancedsettings/>

Missing Movie Scanner

This addon compares the media sources with the database and provides an easy option to add missing videos to the database that could not be automatically identified by the scanners.

<http://kodi.wiki/view/Add-on:Missing%20Movie%20Scanner>

Language Preference Manager

<https://github.com/cyberden/service.LanguagePreferenceManager>

Music Video scraper

The music video scraper TheAudioDb.com ignores unrecognised music videos and the IMVDb scraper matches by title and mismatches artists if the music video is not in the imvdb database.

The “stupid music video scraper” or “just the file name scraper” attempt to fix this by importing all files regardless whether any matches are found or not.

The filenames need to be in the format “ARTIST - TITLE” and should not contain special characters.

Installation instructions for Stupid Scraper:

```
cd /home/kodi/.kodi/addons
sudo curl -LOk
https://github.com/mrjwm2/metadata.musicvideos.stupid/archive/master.zip
sudo unzip master.zip
sudo rm master.zip
sudo service kodi restart
```

Installation instructions for just the file name scraper:

Download zip from: <https://forum.kodi.tv/showthread.php?tid=336042&pid=2779060#pid2779060>

```
Transfer zip to Kodi machine into /home/kodi/.kodi/addons directory
sudo unzip metadata.justthefilename.zip
sudo rm metadata.justthefilename.zip
cd metadata.justthefilename
sudo vi addons.xml
#replace .movies with .musicvideos
```

In Kodi browse to Addons → My Addons → Information Providers → select scraper and enable it. Then go to Settings → Media Sources → Videos → edit the music video source → Set content to scraper.

Let it clear the database, then scan the source for new content.

Separate node for Concerts

File naming same as movies

NFO files can be created if concert doesn't match parser. Create nfo file with same name as concert and add the full URL for either IMDB or TMDB in the file.

1. Add video source:

Videos » Files » Add Videos » Browse » Select your source folder for concerts

2. Set content and scanning:

This directory contains movies (we treat concerts the same as movies)

Make sure information provider is TheMovieDB. (you can even change language settings if you would like from here)

Enable [x] Movies are in separate folders that match the movie title (if appropriate)

3. Setting up a Concert Node

Movies » Left sideblade » Edit Nodes

If Kodi asks you to install and Add-on here, click yes. Its an official Kodi add-on that should make this type of configuration pretty easy (you can also edit XML files in the kodi user data folder if your feeling brave).

New Node » "Concerts" » move selection to this node » context menu ("C" on keyboard) » Edit Node order » Enter "4" as we want to it show just below Music Videos

Set Content » Movies

Add Rule » Change "Title" to "Path" » Change Contains to "Concerts"

4. Remove concerts from movie node

Videos » Left side blade » edit nodes » Movies » Title » Change "Contains" to "Does not Contain" » Change "Rule" to "Concerts"

Source: https://kodi.wiki/view/Music_concerts

To manually edit/create nodes, follow this guide:

https://kodi.wiki/view/Video_nodes

Default nodes: /usr/share/Kodi/system/library/

Custom nodes: /userdata/library/video/

Node rules are stored in:

[addon_data/plugin.library.node.editor/videorules.xml](#)

```
<rules>
  <node name="/home/kodi/.kodi/userdata/library/video/movies">
    <rule field="path" operator="doesnotcontain">
      <value>nfs://192.168.1.6/volume3/concerts/</value>
    </rule>
  </node>
  <node name="/home/kodi/.kodi/userdata/library/video/musicvideos" />
  <node name="/home/kodi/.kodi/userdata/library/video/tvshows" />
  <node name="/home/kodi/.kodi/userdata/library/video/concerts">
    <rule field="path" operator="startswith">
      <value>nfs://192.168.1.6/volume3/concerts/</value>
    </rule>
  </node>
</rules>
```

IPTV

Create or download m3u file with IPTV channels and store it in a folder accessible by kodi.

Monthly updated lists: <https://fluxustv.blogspot.com/p/m3u-list.html>

fluxus-iptv.m3u

Install Kodi PVR IPTV Simple Client addon if it's not available already.

```
sudo apt-get install kodi-pvr-iptvsimple kodi-inputstream-adaptive kodi-  
inputstream-rtmp  
sudo service kodi restart
```

From the Kodi home screen, click each of these items in order to install IPTV:

1. Add-ons
2. My Add-ons
3. PVR Clients
4. PVR IPTV Simple Client
5. Configure
 1. In General Settings, select local M3U Playlist file.
 2. browse to the local m3u file then select OK.
 3. Select OK again to save Kodi settings.
 4. Back in PVR IPTV Simple Client, select Enable.
 5. A dialog will tell you how many IPTV channels were added.

If the TV menu is not on the home screen, navigate to Settings > Interface > Skin > Configure Skin > Enable TV and it will now be added to your main menu.

Note: Each time the m3u file is edited or changed, the plugin needs to be disabled and enabled to reload the new file. Also note that this plugin only supports a single m3u file.

Youtube HD support

Youtube videos contain multiple streams in various quality. Streams containing both video and audio max out at about 360p. Higher quality streams are video only and audio only and the players need to combine them. Youtube uses MPEG-DASH to adapt to bandwidth changes of clients. While the default kodi youtube plugin can play the 360p videos by default, additional settings and plugins are required to play 720p and higher content and to download those if desired.

Requirements:

Kodi Plugins: youtube-dl control, youtube, adaptive input stream, rtmp input stream

```
apt-get install kodi-inputstream-rtmp kodi-inputstream-adaptive rtmpdump  
librtmp1 ffmpeg ffprobe
```

Restart kodi after installing the external plugins.

Enable the adaptive and rtmp input stream plugins in kodi

Add-ons → My Add-ons → VideoPlayer InputStream → InputStream Adaptive
set adaptive to auto select stream (manual will not ask but select low quality stream)
min bandwidth 10000000
enable MPEG Dash in youtube addon

The settings above are enough for HD playback.

Youtube video downloader

The next steps are only required to be able to download youtube video files:

Install the Youtube-dl Context Menu addon:

```
#as kodi user
cd ~
wget
https://github.com/AnonTester/kodi-repo/raw/master/repository.anontester/repository.anontester-1.0.0.zip
```

Kodi → Addons → Addon Browser → Install from zip file → Home directory → repository.anontester-1.0.0.zip

Kodi → Addons → Addon Browser → Install from repository → Wuff's Repository → Context Menus → Youtube dl Context Menu → install

Then enable it via:

Add-ons → My Add-ons → Context Menus → Youtube dl Context Menu → Enable

Updating youtube-dl core

Update youtube-dl source as part of the kodi youtube dl control plugin:

```
#as kodi user:
cd ~/.kodi/addons/script.module.youtube.dl/lib
mkdir tmp
cd tmp
git clone https://github.com/rg3/youtube-dl.git
mv ../youtube_dl ../youtube_dl-old
mv youtube-dl/youtube_dl ..
cd ..
rm -rf tmp
ls -la
#adjust chown command if kodi user/group is different
chown kodi.kodi youtube_dl
python -m youtube_dl --version
```

Amazon Prime

Amazon Prime uses Widevine-DRM (Digital Rights Management) and the Content Decryption Module (CDM) component from Google. Firefox and Google Chrome under Linux come with libwidevinecdm.so to access DRM protected content. There are Kodi plugins that use Chrome or Firefox for Amazon Prime or Netflix.

This description here is for Kodi 18 that comes with the "InputStream Adaptive" plugin which facilitates accessing encrypted content.

Settings → System → Addons → My Addons → Videoplayer InputStream Addons and activate the InputStream Adaptive plugin

enable auto stream selection

disable 'hdcp override' to get the best quality. As of Jan 2019, the InputStream addon doesn't do HDCP handshake (yet) and stream qualities higher than 1080p for TV shows and ~576p for movies (520.000 pixel resolution allowed without HDCP) are prevented by the Amazon DRM requirements on x86 platforms. It should work fine on Android systems.

download the Repo from <https://github.com/Sandmann79/xbmc/releases> and add it to Kodi via Addon Browser → Install from ZIP

Install Amazon VOD Addon from Sandmann79's Repository

at Settings → General you have to choose the "Input Stream" method, disable saving user and log in with valid Amazon credentials

Starting and trying to play any Amazon video will prompt a download box for the widevine library.

The addon script.module.inpustreamhelper that is a dependency of the Amazon VOD plugin is responsible for the download prompt and will ask to update the widevine library when necessary.

Location of the library is in ~/.kodi/cdm/libwidevinecdm.so

Google's libwidevinecdm.so is in /opt/google/chrome/libwidevinecdm.so

Manual extraction of libwidevinecdm.so from google chrome package:

```
mkdir -p ~/.kodi/cdm
cd ~/.kodi/cdm
curl -Lf --progress-bar --url
https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb -o
./temp.deb
ar x temp.deb data.tar.xz
echo "Extracting libwidevinecdm.so..."
tar xJf0 data.tar.xz ./opt/google/chrome/libwidevinecdm.so
>./libwidevinecdm.so
chmod 755 ./libwidevinecdm.so
rm -f temp.deb data.tar.xz
```

If the screen freezes while audio is playing when starting a movie, ensure the HDCP override is disabled in the inputstream adaptive plugin!

Additional resources:

<https://www.kodinerds.net/index.php/Thread/44211-Release-Amazon-Prime-Instant-Video/?pageNo=31&s=eb7e71a1bb5185556a39101524a7fdf27e361d55>

<https://forum.kodi.tv/showthread.php?tid=331136&page=10>

Kodi downgrade

Old versions of Kodi binaries can be downloaded from here:

https://launchpad.net/~team-xbmc/+archive/ubuntu/xbmc-nightly/+packages?field.name_filter=&field.status_filter=superseded&field.series_filter=bionic

Further details and options to downgrade are here:

https://kodi.wiki/view/HOW-TO:Install_Kodi_for_Linux#Reverting_to_an_earlier_nightly

SQLite DBs

Kodi uses SQLite databases by default. For a GUI to edit the database directly (or other databases used by addons), sqlitebrowser works well on linux:

```
sudo apt-get install sqlitebrowser
```

Allow user to restart service

Allow user to start/stop/restart/status services w/o password using sudo

```
visudo /etc/sudoers
```

```
kodi ALL = NOPASSWD: /bin/systemctl restart kodi, /bin/systemctl start kodi,  
/bin/systemctl stop kodi, /bin/systemctl status kodi  
kodi ALL = NOPASSWD: /usr/sbin/service kodi restart, /usr/sbin/service kodi  
start, /usr/sbin/service kodi stop, /usr/sbin/service kodi status
```

Fix refresh rate

In case 60Hz refresh rate is supported but is causing problems and Kodi keeps reverting to 60Hz, the following xrandr command in the `/etc/rc.local` file can be used to adjust the refresh rate before Kodi starts:

[/etc/rc.local](#)

```
#!/bin/sh  
DISPLAY=:0 xrandr --output DP-1 --mode 3840x2160 --rate 30.00
```

```
sudo chmod 755 /etc/rc.local
```

Smart Playlists

Smart playlists are xsp files stored in `/userdata/playlists/music`

Example

[80-90s.xsp](#)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<smartplaylist type="songs">
  <name>80-90s</name>
  <match>all</match>
  <rule field="year" operator="greaterthan">
    <value>1979</value>
  </rule>
  <rule field="year" operator="lessthan">
    <value>2000</value>
  </rule>
  <rule field="time" operator="greaterthan">
    <value>01:00</value>
  </rule>
  <rule field="genre" operator="doesnotcontain">
    <value>Acapella</value>
    <value>Acid</value>
    <value>Acid Jazz</value>
    <value>Celtic</value>
    <value>Comedy</value>
    <value>Dance</value>
    <value>Dance-Pop</value>
    <value>Dance/Electronic</value>
    <value>Dancehall</value>
    <value>Deep House</value>
    <value>Drum And Bass</value>
    <value>Dubstep</value>
    <value>Electronic</value>
    <value>Electronica</value>
    <value>Electropop</value>
    <value>Eurodance</value>
    <value>Happy Hardcore</value>
    <value>Hard Trance</value>
    <value>Jazz</value>
    <value>Neue Deutsche Welle</value>
    <value>Opera</value>
    <value>Progressive House</value>
    <value>Progressive Metal</value>
    <value>Progressive Rock</value>
    <value>Psychedelic Rock</value>
    <value>Rave</value>
    <value>Reggae</value>
    <value>Reggaeton</value>
    <value>Schlager</value>
    <value>Ska</value>
    <value>Techno</value>
    <value>Thrash Metal</value>
    <value>Trance</value>
    <value>Vocal</value>
    <value>World/Jazz</value>
  </rule>
  <rule field="title" operator="doesnotcontain">
```

```

    <value>christmas</value>
  </rule>
  <rule field="title" operator="doesnotcontain">
    <value>xmas</value>
  </rule>
  <rule field="path" operator="doesnotcontain">
    <value>christmas</value>
  </rule>
  <rule field="path" operator="doesnotcontain">
    <value>xmas</value>
  </rule>
  <limit>100</limit>
  <order direction="ascending">random</order>
</smartplaylist>

```

Keyboard shortcuts

<https://kodi.wiki/view/Keymap>

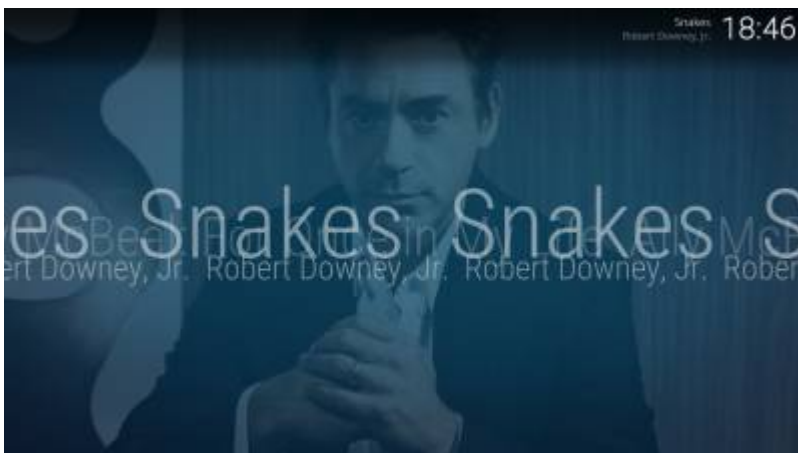
.kodi/userdata/keymaps/keyboard.xml

```

<keymap>
  <global>
    <keyboard>
      <1>ReloadSkin()</1>
    </keyboard>
  </global>
</keymap>

```

Clock in music



.kodi/addons/skin.estuary.modv2/xml/MusicVisualization.xml Add before the last </controls> at end

```

<!-- Start - Clock in Music Vis -->

```

```
<control type="image">
  <left>0</left>
  <top>0</top>
  <width>110%</width>
  <height>160</height>
  <texture
colordiffuse="$VAR[Bars_Opacity]">frame/osdfade.png</texture>
  <animation effect="fade" start="100" end="0" time="300"
condition="Window.Next(screencalibration)">WindowClose</animation>
  <visible>!Player.ShowInfo</visible>
</control>
<control type="grouplist">
  <top>0</top>
  <align>right</align>
  <right>20</right>
  <height>200</height>
  <orientation>horizontal</orientation>
  <width>1270</width>
  <itemgap>10</itemgap>
  <visible>!Window.IsVisible(extendedprogressdialog) +
!Window.IsActive(fullscreenvideo)</visible>
  <animation effect="fade" time="150">VisibleChange</animation>
  <visible>!Player.ShowInfo</visible>
  <control type="group">
    <width>600</width>
    <animation effect="fade"
time="200">VisibleChange</animation>
    <control type="grouplist">
      <left>-75</left>
      <orientation>horizontal</orientation>
      <height>38</height>
      <top>14</top>
      <align>right</align>
      <width>660</width>
      <itemgap>5</itemgap>
      <usecontrolcoords>true</usecontrolcoords>
      <control type="image">
        <top>7</top>
        <width>1</width>
        <height>23</height>
        <texture />
        <visible>Player.HasAudio + !Player.Paused</visible>
      </control>
      <control type="image">
        <top>4</top>
        <width>32</width>
        <height>32</height>
        <aspectratio>keep</aspectratio>
        <texture
colordiffuse="$VAR[SkinColorVar]">$VAR[NowPlayingIconVar]</texture>
        <visible>![Player.HasAudio +
```

```

!Player.Paused]</visible>
    </control>
    <control type="label">
        <label>$INFO[Player.Title]</label>
        <font>font27</font>
        <shadowcolor>$VAR[ShadowColorVar]</shadowcolor>
        <height>35</height>
        <width min="1" max="555">auto</width>
        <scroll>>true</scroll>
    </control>
</control>
<control type="label" id="7700">
    <label>$VAR[NowPlayingSublabelVar]</label>
    <left>5</left>
    <top>48</top>
    <font>font12</font>
    <textcolor>grey</textcolor>
    <shadowcolor>$VAR[ShadowColorVar]</shadowcolor>
    <height>25</height>
    <width>580</width>
    <align>right</align>
    <scroll>>true</scroll>
</control>
</control>
<control type="label">
    <animation effect="fade" start="100" end="0" time="300"
condition="Window.Next(screencalibration)">WindowClose</animation>
    <animation effect="slide" end="0,-10" time="0"
condition="String.IsEqual(Skin.Font,Arial)">Conditional</animation>
    <font>font_clock</font>
    <shadowcolor>$VAR[ShadowColorVar]</shadowcolor>
    <height>200</height>
    <right>203</right>
    <width>auto</width>
    <label>[B]$INFO[System.Time][B]</label>
    <visible>!Player.ShowInfo</visible>
</control>
</control>
<!-- End - Clock in Music Vis -->

```

Screenshot

Stored in Screenshot folder configured in debug settings:

https://kodi.wiki/view/Settings/System/Logging#Screenshot_folder

Keyboard shortcut Ctrl-s or remote command:

```
kodi-send --action="TakeScreenshot"
```



```
# other closing times defined in xml/Custom_1110_CloseVideo0SD.xml
```

Move next episodes to watch to top. Install embuary helper addon (through skin settings) in Home.xml move this:

```

                <include content="WidgetNextWatchEpisodes"
condition="System.AddonIsEnabled(script.embuary.helper) +
Library.HasContent(tvshows) + !Skin.HasSetting(hide_nextepisodes) +
Skin.HasSetting(HomeBanner)">
                    <param name="content_path"
value="plugin://script.embuary.helper?info=getnextup"/>
                    <param name="widget_header"
value="$LOCALIZE[31212]"/>
                    <param name="widget_header_focus"
value="$INFO[Container(6700).ListItem.TVShowTitle, | , -
]$INFO[Container(6700).ListItem.Season,,x]$INFO[Container(6700).ListItem.Epi
sode,,.
]$INFO[Container(6700).ListItem.Title]$INFO[Container(6700).ListItem.Year,
(,)]"/>
                    <param name="browse" value="never"/>
                    <param name="widget_target"
value="videos"/>
                    <param name="list_id" value="6700"/>
                </include>
                <include
content="WidgetNextWatchEpisodesBanner"
condition="System.AddonIsEnabled(script.embuary.helper) +
Library.HasContent(tvshows) + !Skin.HasSetting(hide_nextepisodes) +
!Skin.HasSetting(HomeBanner)">
                    <param name="content_path"
value="plugin://script.embuary.helper?info=getnextup"/>
                    <param name="widget_header"
value="$LOCALIZE[31212]"/>
                    <param name="widget_header_focus"
value="$INFO[Container(67001).ListItem.TVShowTitle, | , -
]$INFO[Container(67001).ListItem.Season,,x]$INFO[Container(67001).ListItem.E
pisode,,.
]$INFO[Container(67001).ListItem.Title]$INFO[Container(67001).ListItem.Year,
(,)]"/>
                    <param name="widget_target"
value="videos"/>
                    <param name="browse" value="never"/>
                    <param name="list_id"
value="67001"/>
                </include>

```

just under:

```

                <include content="WidgetListCategoriesSmall"
condition="Library.HasContent(tvshows) +
!Skin.HasSetting(hide_tvshowscategory)">

```

```
value="$LOCALIZE[31148]"/>
value="69001"/>
value="Skin.HasSetting(UseSmallerCategories)" />
</include>
```

From:

<http://wuff.dyndns.org/> - **Wulf's Various Things**

Permanent link:

<http://wuff.dyndns.org/doku.php?id=kodi:kodi-config>

Last update: **2025/09/18 15:38**

