gtkpod
Table Of Contents

Unlike other applications, cover art in gtkpod is not hidden away in a "difficult to find" image cache. In fact, cover art files, for a local database, can be located anywhere on the filesystem.

Setting Image on Track Import

Locating the image file in the same directory as the tracks gives the additional benefit that gtkpod will analyse the image upon track import and set the image as the album's cover artwork. It is Important to ensure the image's filename corresponds to at least one of the image filename templates entered in the Preferences window.

Setting Image on Existing Album

The use of the popup menu to select a cover art file from the filesystem has been documented here. It is also possible to download a cover art image file from the internet and drag n drop it into gtkpod.

Drag 'n Drop Using Firefox

1. Locate the preferred image and display it in firefox.
2. Ensure the url of the image ends with ".jpg" or ".JPG" (other file extensions could be supported if demand warrants it).
3. Drag the image from Firefox and drop it onto the Cover Art Display's centre/main cover image. The main cover image should be displaying the album which requires the new cover.
4. If a cover already exists with the same filename then gtkpod will ask whether the existing image be overwritten or the new image be named something else.
5. The image will be downloaded from the url location, saved in the album's directory on the filesystem and applied to each track belonging to that album.

Drag 'n Drop Using Konqueror

- A limitation of the dnd in Firefox is that only the url can be used to get at the image file. Thus, the page must be displaying just the image and gtkpod must redownload the image internally (using curl and the image url).
- Konqueror offers the additional functionality that the image file can be extracted directly from its cache. This has the benefits of avoiding the internal curl download and that the url is not restricted to those ending in ".jpg"
- The importance of the latter benefit concerns websites that do not offer the direct download of cover art images. allcdcovers.com is one such site, where its images are displayed in the browser

but the image url cannot be used a second time to download the same image. Thus, allcdcovers.com cannot be used as a download dnd source in Firefox. However, in Konqueror, simply open a full image from allcdcovers then just drag 'n drop into gtkpod.

- Maybe Firefox will do this some day too..

Getting Tracks into gtkpod

- You can add individual tracks, entire directories recursively, or existing playlists using "Add Files". A file selection dialogue will appear. By default existing tracks (same full filename) will be skipped.
- You can add directories recursively using "Add Dirs". A directory selection dialogue will appear.
- You can add existing playlists using "Add Playlists"
- You can delete tracks by marking them and pressing the "Ctrl-d" . If you delete tracks from the master playlist (the topmost playlist, called "gtkpod" by default). You can also delete tracks by selecting them, and using "Delete Track" in the Edit menu or from the context menu.
- You can update ID3 tags of selected tracks in gtkpod from the mp3 file by pressing "Ctrl-u" or choose "Update track info from file" in the Edit menu or from the context menu.
- You can rename ID3 tags by editing the fields in gtkpod. You can change an entire group of ID3 tags by editing an entry in the sort tab (or mark several tracks and edit the first).
- You can specify which tags to display in the preferences window.
- You can specify (in the prefs window) if the ID3 tags of the corresponding mp3 file(s) should also be updated.

Add Directories

- Can add music and video files invidually using Add Files.
- Can add directories of music and video file using Add Directories.
- The files are imported into the selected repository. If the selected repository is an iPod then the files will be copied directly onto it.
- See Supported File Types for types of files that can be imported.

Note: gtkpod is an iPod management tool. Tracks, videos and photographs already need to be on the computer filesystem before they can be imported. If adding a CD is the requirement then it needs to be ripped first. Have a look at:

  - grip
  - Sound Juicer
  - kaudiocreator

Dragging and Dropping

- Once imported tracks can be dragged around gtkpod, including inbetween repositories, playlists, filter tabs and the track display pane.

- A drop _onto_ an existing playlist will add the tracks to that playlist.
- A drop _between_ two existing playlists or behind the last playlist will create a new playlist to which the tracks are added.
- The default action for the drop is either move or copy as appears appropriate (e.g. playlists are moved within a database ('local' or 'iPod/Shuffle'), but copied when dragged across different databases). The applicable action is displayed within the drag icon and can be changed by pressing the control key during the drop.

- Adding individual files
- Adding directories
- View/Edit track details
    - Cover Art
- On-the-fly conversion
- Supported file types

- Saving files from the iPod to your hard drive
- Syncing directories and playlists

Most of the time it should be possible to download gtkpod from your O.S' package repositories. See Distro and OS specific info for details of some examples.

Should you want to install the latest version and there is not yet a package available then you will have to compile gtkpod from its source.

- The latest stable release is available for download from here.
- Should you want the latest cutting edge but maybe unstable source code then you can check it out from the sourceforge repository.

Like a lot of open source projects, gtkpod and libgpod following the conventional './configure -> make -> make install' command procedure.

1. gtkpod requires certain dependencies to be installed for a successful compilation. If you install packages from your linux distribution, keep in mind that you will need to install the "-dev" or "-devel" packages as well to be able to compile.
   Software required to build the gtkpod-0.99.12 includes:

Essential

- autoconf (at least 2.55)
- flex (or lex)
- gettext
- glib (at least 2.4.0)
- gtk+ (at least 2.6.0)
- libglade (at least 2.4.0)
- libgnomecanvas (at least 2.14.0)
- libgpod (at least 0.6.0) (see here for details if not installing from source)
- libid3tag (at least 0.15)
- perl XML::Parser module
- pkgconfig

Optional

- libflac - flac file support
- libvorbis - ogg vorbis file support
- libmp4v2 - aac / m4a file support
- libcurl - for drag n dropping coverart from web browser
- libhal > 0.5
- libgnome-vfs-2.0 > 2.6 (for iPod autodetection under GNOME)

Note: If you are building gtkpod from subversion, you will often need to build libgpod from subversion also. The other software and/or versions required may differ as well.

First, make sure you have compiled and installed the dependencies. Most Linux distros will include them, except maybe for libid3tag (http://sourceforge.net/project/showfiles.php?group_id=12349) and libmp4v2 (http://mpeg4ip.sourceforge.net/ or http://resare.com/libmp4v2/)

- Please do not confuse the libid3 library with the libid3tag library -- they are not related. libid3tag is part of the MAD project (http://sourceforge.net/projects/mad/).
- libid3tag was successfully compiled and installed with ./configure ; make ; make install
- libmp4v2 was successfully compiled and install with ./bootstrap --disable-server ; make ; make install

If you install libraries to /usr/local/lib please don't forget to add the path to LD_LIBRARY_PATH and PKG_CONFIG_PATH. Should you install either application to an alternative location then you should adjust these values appropriately.

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig
export LD_LIBRARY_PATH
export PKG_CONFIG_PATH
```

You can add those lines to your ~/.bashrc or add it globally to /etc/profile.

Run the gtkpod 'configure' script to set up the compile. If using a subversion checkout of the code then you will need to use autogen.sh instead of configure.

```
./configure
or
./autogen.sh
```

The standard options to 'configure' apply. For a list of options:

```
./configure --help
```

Compile the gtkpod software:

```
make
```

Install the gtkpod software (may require root privileges):

```
make install
```

- Note: You can test gtkpod without installing it by running the following. Some features (translation catalogues, icons, scripts) may not be available, however:

```
src/gtkpod
```

- Note that you do not need to accept the default (usually /usr/local) install path when you compile and install the software. You may choose to install the software in a different location. For example, with libid3tag and libmp4v2 in a non-standard location, outside the $PATH and the usual build environment, you need to configure the build of gtkpod appropriately. You need to set a CFLAGS variable during the 'configure' that sets the -I and -L flags correctly.

- As a practical example, let's say the libmp4v2 software was installed in $HOME/Applications/mpeg4ip and the libid3tag software was installed in $HOME/Applications/libid3tag, and you want to install gtkpod into $HOME/Applications/gtkpod. Do this instead:

```
CFLAGS="-I$HOME/Applications/mpeg4ip/include \
-L$HOME/Applications/mpeg4ip/lib -I$HOME/Applications/libid3tag/includ
e \
-L$HOME/Applications/libid3tag/lib" \
./configure --prefix=$HOME/Applications/gtkpod ; make ; make install
```

Getting Tracks onto the iPod

- Tracks can be imported directly to an iPod by ensuring the latter is the selected repository before clicking Add Files or Add Directories.
- Normally though, tracks will already have been imported into the Local repository. To copy them onto the iPod simply select the tracks in the Track Window and drag them onto the iPod repository icon.
- To copy a whole album, selected the album in one of the sort tabs and drag that onto the iPod repository.

Playlists

- Use playlists to group together a set of tracks. Simply create a new playlist in the Local repository then drag n drop the tracks onto it.
- Tracks are not actually copied into a playlist so should a track be deleted then it will be deleted from all playlists.
- Drag n drop a playlist onto the iPod repository icon to copy it and all its tracks to the iPod. The playlist will be available in the Playlist section of the iPod's control system as well as all tracks being copied to the iPod.

Save Changes

- At any change to a repository, eg. track addition, deletion, edit ..., a change has occurred that requires a database save. To highlight this, the name of the repository is changed to italics, ie. Local -> *Local*. This shows the repository requires saving.
- Should gtkpod be closed prior to any saving then a dialog is displayed informing that data changes will be lost if a save is not performed.
- Clicking the "Save Changes" button will save all changes made to all of the repositories. This is the point where tracks are actually copied to the iPod and the iPod database updated. Only after a save is performed will the tracks copied to the iPod, be playable on the iPod.

Cover art is a visual element of your gtkpod music collection. It changes the black and white database of artists and title names back into something related to the physical CDs and DVDs that have been imported.

CoverArt Display

The display consists of 9 covers displayed at any one time. The entire album collection can be cycled through from one end to another. The main image is slighly larger and lacking in "shine" for viewing clarity.

- Use the scrollbar to quickly slide back and forth over the album collection.
- Use the scrollbar buttons to more precisely step through each album.
- Click on the main album cover image (one at the front and in the centre) once to display just the album's tracks in the right pane of gtkpod

---

CoverArt Popup Menu

- Right-click on the main album cover image to display the popup menu with 3 options:
  - Select Cover From File
  - View Full Size Artwork
  - Edit Track Details

Select Cover From File

- Selecting this option displays a file chooser dialog. Select the preferred image file and click OK. The album cover changes to a scaled version of the new image, while all the tracks in the album have their artwork path modified to the path of the preferred image.

View Full Size Artwork

- Selecting this options opens a dialog displaying the cover image. The image is either its true size of the biggest size it can be while still fitting on the desktop. Its true resolution is displayed underneath in the format width x height.
- The dialog is modal so must be closed before moving on. The OK button closes the dialog.

Edit Track Details

- This provides a shortcut to the "Edit Track Details" dialog for editing the ID3 tag information for all tracks belonging to the album displayed.

Distribution

Fedora Mandriva Ubuntu Suse Debian

Fedora 8

Available from Everything repository as an rpm:

```
yum install gtkpod
```

Mandriva 2008

Available from the Mandriva club (requires registration)

```
http://club.mandriva.com/xwiki/bin/view/rpms/Application/libgpod2
http://club.mandriva.com/xwiki/bin/view/rpms/Application/gtkpod
```

or have a look at http://easyurpmi.zarb.org/ for mirror sites.

Ubuntu Gutsy Gibbon

Current instructions for settings up the iPhone can be found here. Also, if you have the Classic or latest "fat" nano then look at the instructions at the bottom of the page for the repository on where to install gtkpod from.

Available from the official repositories:

```
apt-get install gtkpod
```

For aac support, this would be preferred:

```
apt-get install gtkpod-aac
```

For those using Amarok

add the following line to your sources.list

```
deb http://ppa.launchpad.net/ipod-touch/ubuntu gutsy main
```

and run the following if you do not have amarok installed:

```
sudo apt-get update
sudo apt-get install amarok
```

or run this if you just want to upgrade your existing amarok instalation

```
sudo apt-get update
sudo apt-get upgrade
```

Suse

Debian

on Lenny

```
apt-get install gtkpod
```

Contributing Financially

Donations are welcome: please go to here for details.

Copying from the iPod

Unlike other iPod management applications, it is possible to copy tracks resident on the iPod back onto a PC, using gtkpod.

- Mark the tracks you want to export and select "Export Tracks from Database" from the file menu (or use the context sensitive menu).
- A file selection dialog window appears and you can choose the directory you'd like the selected files to be written to.
- You can specify the output filename in the prefs dialog by specifying a template (e.g. "%A/%a - %t"). You can specify multiple templates for different file formats by separating them by a semicolon (e.g. "%A/%a - %t.mp3;%t.wav").
- Identifiers: You can separate several patterns by semicolons -- gtkpod will determine which one to use by the filename extension given. Artist: %a, album: %A, composer: %c, title: %t, genre: %G, track nr: %T, CD nr: %C, year: %Y, original filename (requires extended information file): %o, current playlist: %p, the character '%': %%.

Contents:

I have a new iphone or ipod touch and I want to sync with linux. What to I do please?

Solution
Marcan has provided some excellent blog entries so have a look at these. If you are unsure about your firmware version and your iphone or itouch is recent then its probably version 3.

Users with 3.x firmware
http://marcansoft.com/blog/2009/10/iphone-syncing-on-linux/

http://marcansoft.com/blog/2009/10/iphone-syncing-on-linux-part-2/

Users with 2.x firmware
http://marcansoft.com/blog/2009/01/using-amarok-and-other-itunesdb-compatible-software-with-the-iphone-2x

---

I am receiving the error message "ipod database Import failed illegal seek to offset 0" when trying to use an iphone 3G with gtkpod?

Solution
This error means that the libgpod library installed on your system is the incorrect version. At the moment you will require the version from the
sourceforge git repository. Ensure other versions of libgpod are removed from the system to avoid gtkpod using them by mistake.

I have just upgraded my new ipod to the 1.1.3 firmware and have successfully jailbreaked but gtkpod will not work with it?

Solution
To quote DarkShadow in full:
"I just upgraded to the dev team 1.1.3 jailbreak and to use it with gtkpod you have to modify the mount command and change /var/root/Media to /var/mobile/Media in iPod convenience's scripts. After that it will work with gtkpod again."

My iPhone or iTouch doesnt have a USB connection. How can I mount it on my linux box?

Solution
The iPhone or iTouch, at present has to be mounted using sshfs. Openssh must also be installed on the iPod Touch for this to work.
The folks at Ubuntu have provided a very helpful site for information on getting these iPods to work.

I have heard that my iPhone or iTouch must be jailbroken. What does that mean?

Solution
Answer taken from here

The iPhone or iPod Touch must first be "jailbroken" so that you can modify the software which it runs.
Determine your Firmware Version
Follow this wiki page for information on how to properly jailbreak.
It has directions for both 1.1.1 and 1.1.2 firmwares. Note that if you want to keep firmware 1.1.2, you may need a Mac or Windows machine to assist you.

My iPod is an iPhone, iTouch, Classic or Nano3g. My tunes show up in gtkpod but I see nothing on the iPod when I eject?

Solution
This is to do with requiring your iPod model number in a sysinfo file located on the iPod. Please see here for details.

Loaded up gtkpod for the first time and I receive the message "Could not open "iTunesDB.ext" for reading extended info. Extended info will not be used."?

Solution
iTunesDB.ext is a gtkpod/libgpod specific file. If you have never uploaded music using gtkpod to the your iPod, then it's perfectly normal that this file is absent. This file will be created automatically by gtkpod at the save stage (if you enabled "Edit->Preferences->Write extended information" of course).

But even if you consider not using this extended information, all should work. You will lose only some bonuses like detection of duplicates.

Installed libmp4v2 or libgpod from source to /usr/local/lib, but gtkpod is unable to locate libmp4.so.0 or libgpod.so.0?

Solution
If you install to /usr/local/lib please don't forget to add the path to LD_LIBRARY_PATH:

LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
export LD_LIBRARY_PATH

You can add those lines to your ~/.bashrc or add it globally to
/etc/profile.

Files copied to gtkpod but they don't appear in the database (0.80, Tony Williams)?

I'm having a problem that I wonder if you've seen. I've setup gtkpod. launch it and add files. I can hear the ipod harddrive spin up. If I go onto the ipod directly I can see the space being used and can even see the new files on the ipod. However the ipod interface doesn't show the new files and gtkpod keeps telling me that there are orphaned files.

Solution
I finally figured out the problem. In my fstab I had the filesystem set to 'auto' and it was mounting as msdos instead of vfat. I specified vfat in fstab and voila! I'm a happy happy man.

Filenames on the iPod appear in DOS 8.3 format and syncing to the iPod is not working as expected?

Solution
You need to specify "vfat" as file system type. How to do that depends on which way you use to mount the iPod -- see the README file for more details.

./autogen.sh does not work?

Solution
A ubuntu user has reported that he had to set

ACLOCAL_FLAGS=" -I /usr/share/aclocal/"

in order to get ./autogen.sh to work.

The following error message is displayed when accessing the device:

ieee1394: sbp2: aborting sbp2 command
Test Unit Ready 00 00 00 00 00

Solution
(Markus Gaugusch):
It is possible that hotplug and the "sg" support are not working well together. Try disabling "sg" support in the kernel configuration or
unload the "sg" module if you are using modules.

(Justin Thiessen):
Forcing the sbp2 module to serialized I/O seems (so far) to have solved the problem.

Ref.:
http://www.netzwerk-aktiv.com/pub/doc/newsletters/linux1394-user/html/1676.html
http://www.ubuntuforums.org/printthread.php?t=6678
http://66.102.7.104/search?q=cache:Xh_gu43y6w8J:themikecam.com/newmikecam/blog/index.php/geek/2005/+ipod+serialize_io&hl=en

Looks like the driver is going to be set to default to serialized I/O in kernel 2.6.14, anyways.

http://linuxtoday.com/developer/20050930001026NWKNDV

The following error message is displayed when accessing the device

usb-storage: Attempting to get CSW...
usb-storage: usb_stor_bulk_transfer_buf: xfer 13 bytes
usb-storage: Status code -75; transferred 0/13

Solution (by Ingo Reimann)
I tried to use an iPod Mini with my nforce2-Board, kernel 2.6.14/2.6.15 debian sid and got messages like [above] in dmesg. /dev/sda appeared, but fdisk -l did not show anything

The solution, that i found in a discussion on http://kerneltrap.org/node/3844 was to unload ehci_hcd.

When connecting an iPod via USB to a 2.6 kernel machine the iPod wil be recognized but not work correctly. In /var/log/messages you'll see the a bunch of "Buffer I/O error on device sd?" when connecting the iPod (Jonas Bergler, Kevin Venkiteswaran)

Solution (by "jeffmock")
Disable CONFIG_EFI_PARTITION (File Systems -> Partition Types -> Advanced Partition Selection -> EFI GUID Partition support) in your kernel configuration, recompile.
Details can be found at
http://www.linuxquestions.org/questions/showthread.php?postid=1197015#post1197015

Excerpt:

"This problem could potentially happen with both 2.4 and 2.6 kernels.

A longer story follows and perhaps someone can come up with a more sensible solution for the long run.

The iPod looks like a removable disk drive to the host computer. When it is attached to the computer, the mini iPod reports a capacity of 7999488 512-byte sectors (or about 4GB). This turns out to be wrong for whatever reason. The mini iPod only really has 7999376 sectors and it exaggerates by 112 sectors. The other quality of the iPod is that if the computer attempts to read a sector greater than the actual capacity but less than the reported capacity, the iPod will dutifully report an I/O error, but it won't respond to any future requests until you unplug/plug the iPod."

I followed the kernel recompile instructions for distro, disabled only the CONFIG_EFI_PARTITION option, and things ran perfectly for me afterwards. As indicated above, hopefully a better long-term solution will emerge soon."

(Jorg Schuler: it seems a patch was introduced in kernel version 2.6.10: "<phil@ipom.com> [PATCH] USB Storage: Add unusual_devs entry for iPod This patch adds an unusual_devs.h entry for the Apple iPod as it reports one too many sectors. The patch was submitted by Avi Kivity <avi@argo.co.il> and re-diffed by me.")

If you have a similar problem with a newer kernel, make sure the USB port you connected your iPod to has sufficient power, because it may cause a similar error.

SHUFFLE won't play music after reformatting

Solution (by Mark Davis)
The SHUFFLE seems to care about the volume name which has to be "IPOD". Try to format as (replace /dev/sda1 with the appropriate device file for your SHUFFLE!):

mkdosfs -F32 -n IPOD /dev/sda1

Calendar entries mixed up?

Solution
The iPod does not appear to like times specified in UTC (indicated by a trailing 'Z'). KOrganizer seems to do this. If you know how to work around it let me know.

m4a files created by faac cannot be added by gtkpod (gentoo)?

Solution
There appear to be some versions of faac that do not create 'good' m4a files. The problem could be solved under gentoo by using version 1.24.
In order to make error tracking easier, more detailled error messages are displayed when tracks could not be added for any reason starting with version 0.91-CVS of gtkpod.

gtkpod crashes when reading the iTunesDB (Fedora)

Solution
It appears that crashes were observed with kernel version 2.6.11-1.35_FC3. An upgrade to 2.6.12-1.1376_FC3 got rid of the problem. This was with gtkpod-0.94.0 and Athlon64 3000+.

After updating the firmware of a 1st-gen iPod nano**, crashes are also observed with Fedora Core 5, kernel 2.6.20-1.2320.fc5, gtkpod 0.99.4 (which worked with the previous firmware). Downloading, building, and installing gtkpod 0.99.10 and libgpod 0.6.0 solved the problem.

Problems connecting the iPod to Solaris/SPARC

Solution
Current (as of 2006/03/30) versions of the Solaris pcfs SPARC driver have a bug where the correct filesystem/partition layout may not be
recognized, and this is true for iPods. This prevents the iPod partition from being mounted on Solaris SPARC. In order to work
around this, one must prevent pcfs from detecting the first FAT32 filesystem, forcing it to move on to the second one. This can be done

by changing the filesystem identifier like so:

1. dd if=/dev/rdsk/c3t0d0s2 of=/tmp/ipod.orig count=1
2. cp /tmp/ipod.orig /tmp/ipod.modified
3. dd if=/tmp/ipod.modified of=/dev/rdsk/c3t0d0s2 count=1

The above modification may or may not affect usage on other systems. Some reports encounter no problems using the iPod or iTunes after
making it. In my own experience, my iPod hard locked moving from a SPARC system to an x86 system, and I had to undo the change. Wrapper
scripts to handle this may be desirable on SPARC.

Eric Enright

---

iTunes cannot read the contents of the iPhone

If you connect your iPhone to iTunes and get the message "iTunes cannot read the contents of the iPhone," you can easily fix this by removing or renaming the /var/mobile/Media directory:

```
$ ssh root@iphone
Password:
$ mv /var/mobile/Media /var/mobile/Media.aside
```

OR

```
$ rm -r /var/mobile/Media
```

After this you should be able to sync from iTunes again. Note that it will create the /var/mobile/Media hierarchy from scratch, and you will lose any music, videos, and photos currently on the phone.

Content for this section:

- Hooking up your iPod for the first time
- Compiling and Installing
- OS / Distro Specific Info
- Setting the iPod Properties
- Supported iPods

(these will likely end up being separate pages)

Instructions on how to checkout source code from the git repository is available from sourceforge here. The git code is the latest and cutting edge iteration and as such may contain bugs. This is really for experienced users and anyone who enjoys tinkering at the code level.

To checkout the gtkpod trunk from source forge:

- `git clone git://git.code.sf.net/p/gtkpod/code`

To checkout the libgpod trunk from sourceforge:

- `git clone git://git.code.sf.net/p/gtkpod/libgpod`

In addition, the code repositories are mirrored on gitorious at

- http://gitorious.org/gtkpod/gtkpod
- http://gitorious.org/libgpod

The benefit of gitorious is that the repositories can be cloned on the site as personal sandbox repositories. This is beneficial if a feature has been developed and requires public exposure, for testing maybe, before being added to the main codebase.

Welcome!

This wiki has now been deprecated in favour of the wiki at gtkpod.org.

Please have a look at the latest information and versions of both gtkpod and libgpod.

- **gtkpod is the application for managing your ipod on linux.**
- **libgpod is the ubiquitous library for making linux ipod management possible.**

This wiki is an ever-growing archive of information dedicated to the installation and operation of gtkpod and libgpod. If you need more information please have a look at the mailing lists or come along to the irc channel for a chat. Find all the contact information here.

To navigate this wiki site please use to the navigation panel on the left.

Current Development Status

Releases

- libgpod-0.7.2
- gtkpod-0.99.14

Support

- iPod Classic (late 2009), iPod Touch/iPhone (including firmware 3.x) and iPod Nano 5th Generation are supported with libgpod 0.7.90 (development release) and newer releases.
- iPod Shuffle 3G/4G have no support at this time

sourceforge git repository at git://gtkpod.git.sourceforge.net/gitroot/gtkpod/gtkpod

Introduction

The iPod has largely been an USB device yet there are older models of iPods that used IEEE1394/ / firewire for their connection interface. Historically, it took something of an effort for linux to recognise the iPod model correctly. However, this situation has greatly improved in recent times where it is now possible to simply plug the iPod into a USB port and have your distro recognise it straight off.

gtkpod first and foremost relies on the successful mounting of a recognised iPod. Thus, if the iPod cannot be mounted then gtkpod is NOT going to do it for you! Thus, before firing up gtkpod make sure you can see the filesystem of your iPod at its designated mount point, eg. /mnt/ipod.

Using udev

For linux distros installed with hal and udev, plugging an iPod in and mounting it becomes a trivial exercise. A device node will normally be created under /dev, eg. /dev/sdc.
Using udev rules it is possible to "play" and refine this device node to reflect personal requirements. For example, including these udev rules will allow 2 iPods to be loaded at the same time without interfering with one another:

```
#80GB IPOD
SUBSYSTEMS=="usb", ATTRS{serial}=="000A2700XXXXXXXX", KERNEL=="sd?2",
\
NAME="80gbipod", MODE="0664", OPTIONS="last_rule"

#4GB IPOD NANO
SUBSYSTEMS=="usb", ATTRS{serial}=="000A2700YYYYYYYY", KERNEL=="sd?2",
\
NAME="4gbnano", MODE="0664", OPTIONS="last_rule"
```

By including the test against the serial number it is possible to uniquely identify an individual iPod and load it appropriately.

Mounting

The result of this is that the iPod will be located on a device node and this can be mounted manually using the command (performed as root):

```
mount /dev/sdc2 /mnt/ipod
```

This assumes an iPod is loaded onto the device /dev/sdc and that it is a 2 partition model. It seems that

post-2006, iPods have become 1-partition items.

However, it should be noted that modern window managers such as gnome and kde take on the responsibility of managing connected devices. Thus, the result of connecting an iPod will be an icon on the desktop which will either be mounted automatically or can be mounted by the user with a click of the mouse on a popup menu.

iTouch / iPhone / iPad(?) / Nano 5g Users

This section is taken from the README files in the libgpod source repository. These devices are a little more complicated to connect hence this tries to make the process a little clearer. Ideally it will be helpful to help application writers/distributors/... figure out what's going wrong when trying to interact with an iPod-like device.

Please note that not all devices need all these steps (especially the hash58 / hash72 stuff and the sqlite databases), see the table at the end for more details

Overview

1. **Device is plugged in** - HAL/udev detects the insertion and runs a callout. This callout sends the appropriate commands (raw SCSI/USB commands or special AFC command) to the device to get an XML file describing the device capabilities (artwork formats supported, serial number, ...) and dumps this XML file to iPod_Control/Device/SysInfoExtended for future use by libgpod. libgpod doesn't do it directly because sending these SCSI/USB commands might need elevated priviledges.
2. **Application uses libgpod to read the device content** - iPod_Control/iTunes/iTunesDB is accessed as well as a few other files. If the device needs an hash72 (either in its iTunesDB or because it's using sqlite databases), then the necessary information is extracted from the existing iTunesDB to generate an iPod_Control/Device/HashInfo file (if it doesn't exist) which will be useful to generate any hash72 we need for this device.
3. **Application uses libgpod to write to the device -** libgpod generates an iPod_Control/iTunes/iTunesDB file. If an hash58 is needed then the device FirewireID is used to generate it. If an hash72 is needed then the HashInfo file is used to generate it. If the device uses sqlite databases, they are generated as well. Post process SQL commands are then extracted from the device (either by getting them from SysInfoExtended or by getting them through AFC) and executed after the sqlite database generation. These post process commands are useful to make sure the sqlite databases layout matches what the current device firmware expects.

Glossary

- iTunesDB: binary file containing information about all the songs, playlists, ... stored on the device. On recent devices (iPhoneOS 3.x, Nano 5g), it's replaced by iTunesCDB which is a compressed version of iTunesDB.
- sqlite: recent devices (iPhoneOS 3.x, Nano5g) use sqlite databases instead of an iTunesDB to store song information, playlists, ... More accurately, they have both an iTunesCDB (compressed iTunesDB) used by iTunes to know the device content and sqlite databases used by the device. Along with these sqlite databases, there's a .cbk file which contains checksum information and a hash72 for the sqlite data.
- iPhoneOS: umbrella for iPhone-like devices, ie iPhones, iPod Touch and (likely) iPad. On these devices, what matters is the firmware version, not the device type.
- hash58: first iTunesDB hashing scheme introduced by Apple. It appeared in the iPod Nano Video (3g) and iPod Classic and was used by iPhoneOS 1.x. It's fully reverse-engineered and uses the iPod FirewireID (called this way even on USB devices) as part of the calculation.
- hash72: hashing scheme that was first introduced in iPhoneOS 2.x and has then been used on the iPod Nano with a camera (5g). It was much more complicated to reverse engineer and is not yet 100% known. However, given a file with a valid hash72, we can extract some hashing data to be able to generate valid hash72 hashes for any file.

iPod feature matrix

Model	SysInfoExtended	hash58	hash72	sqlite	iTunesCDB
iPod 1G	no	no	no	no	no
iPod 2G	no	no	no	no	no
iPod 3G	no	no	no	no	no
Mini 1G	no	no	no	no	no
Mini 2G	no	no	no	no	no
iPod 4G	yes	no	no	no	no
iPod 5G	yes	no	no	no	no
Nano 1G	yes	no	no	no	no
Nano 2G	yes	no	no	no	no
iPod Classic	yes	yes	no	no	no
Nano3G	yes	yes	no	no	no
Nano4G	yes	yes	no	no	no

iPhoneOS 1.x	yes	yes	no	no	no
iPhoneOS 2.x	yes	no	yes	no	no
Nano5G	yes	yes[1]	yes	yes	yes
iPhoneOS 3.x	yes	no	yes	yes	yes

[1] surprisingly, the Nano5G uses a hash58 for its iTunesCDB and a hash72 for its sqlite cbk file. iPhoneOS 3.x uses hash72 everywhere.

Environment variables

The following environment variables can be set if needed:

- IPOD_MOUNTPOINT: Defines the mountpoint of the iPod. This overwrites the value stored in the preferences, but is overwritten by the command line argument "-m" or "--mountpoint".
- GTKPOD_DF_COMMAND: Only used on systems without statvfs(). Defines the "df" command to be used for probing the free space on the iPod including command line arguments. Default is "df -k -P". On some systems it may be necessary to remove the "-P" option. The mount point is added to this command line automatically. You can switch off calls to df by setting this environment variable to an empty string.

Checking iPod's Files

For whatever reason -- it may happen that tracks are present in your iTunesDB that are no longer present on the iPod (dangling tracks), or
that tracks are on the iPod but not in the iTunesDB (orphaned tracks). The function "Checking iPod's Files" under the "File" menu will identify both types and take the following actions:

Orphaned tracks

A new playlist "[Orphaned]" will be created with all orphaned tracks in it. The only exception are orphaned tracks that would become duplicates (if duplicate detection is activated). Those are marked for deletion with the next sync.

Dangling tracks

These tracks will be marked for deletion with the next sync unless the original PC file is still available. In that case you can have them restored with the next sync.

Restoring the iPod After a File System Error

If iPod's file system gets corrupted and you need to reformat your iPod, there is a way to restore the contents semi-automatically if you have been using the "write extended information file" (iTunesDB.ext) options:

- If the directory structure on the iPod doesn't exist yet, load the iPod in gtkpod and have it created for you. Then unload the iPod again.
- Copy your backup files in .gtkpod/ (usually iTunesDB and iTunesDB.ext) to your iPod (usually /iPod_Control/iTunes/
- On the iPod the files must be named iTunesDB and iTunesDB.ext.
- Load the iPod in gtkpod.
- Select the iPod repository and start "Check iPod's files" from the File menu.
- Unload the iPod (or save changes).

This should restore your iPod to what it was before, provided you didn't move or remove any of the original tracks on your harddrive, and the charset information was stored correctly.

Typical View of gtkpod

What is a Repository?

The left hand pane in gtkpod displays all the repositories that have been created. A repository is (like it implies) a collection of all music tracks, videos and photographs imported into it. They are associated together by a binary XML database file that is the same format as the one that sits on an iPod.

By default, gtkpod creates a Local repository and a Podcast Repository. These sit on the computer with gtkpod (they can be found under the ~/.gtkpod directory). By having a local repository, it is possible to import tracks into gtkpod without needing to copy them straight to an iPod. This means that if the cover art, for example, needs to be changed then tracks can be managed without even having the iPod plugged in.

Of course, gtkpod is pointless without accessing an iPod. In the same pane as the local repository, new iPod repositories can be created. These repositories related to iPods by model and mount point. Normally, one repository is sufficient but as you can see below, it is possible to set up multiple iPod repositories on the same computer.

Repository Pane

- Below each repository, available playlists are listed along with a podcast playlist.
- Under iPods that support photographs, a special dynamic node is created that, when clicked, invokes the photo management tool. This allows straight addition of photos to the iPod as well as management of photo albums on the iPod. **Note:** this photo node cannot be deleted!! There is no point trying as it is created by gtkpod for the sole purpose of displaying the photo management tool.

Menus and Toolbar

- Menus across the top provide alternative access to the most frequently used functions.
- Toolbar provides the following actions:

- Load iPod(s) -> used for manually mounting any iPods already set up in gtkpod
- Save Changes -> vital button! The Save All function that save any changes made to loaded iPods or the local database.
- Files -> Add individual music files to the selected repository.
- Dirs -> Add any music files located in the selected directories to the selected repository.
- Playlist -> Add a playlist file to the selected repository.
- New PL -> Create a new playlist on the selected repository.

Filter Tabs

- The two notebooks above the track display are called "Filter Tabs".
- They allow you to filter which tracks are displayed.
- If you edit an entry in the filter tab, the corresponding entry in all associated tracks will be updated as well. When writing the tags to disk as well, updating of a large number of tracks may take a while.
- So in the image above, choosing "Alice Cooper" on the first filter tab and "Trash" on the second filter tab will only display the tracks from that most excellent of albums (no dissenting opinions required! ;-)).
- Obviously, these filters can be altered to suit. Selecting ALL on a sort tab effectively removes the filter and displays all tracks.

Track Window

- Displays all tracks residing in the selected repository and conforming to the selected filters.
- The columns can be sorted and are interchangeable.

Cover Art Display

- Displays the cover art of all albums stored in the repository.
- If no cover art exists for an album then the default "?" image is displayed.
- See here for more information.

Trademark Proviso

iPod, iTunes, Mac, Macintosh, iBook are trademarks of Apple Computer, Inc., registered in the U.S. and other countries.

License

gtkpod is licensed under the Gnu Public License Version 2. An online version is available here, while an offline version is available in the tarball of each gtkpod release.

- Normal Playlists
- Smart Playlists
  There seems to be an issue with SmatPlaylist in the latest relase: They do not appear in on the iPod
  I was told that this got fixed in svn

Whenever a track is played completely on the iPod (firmware version 1.3 or higher) a counter in the "Play Counts" file will be increased by one. The same file also contains the rating set with the 3rd generation (and later) iPods. This file appears to be deleted whenever the iPod resets itself, e.g. because the iPod is disconnected from the computer. Therefore, be careful! Charging the iPod seems to do no harm.

When gtkpod is started, it will also read this file and incorporate the information into the iTunesDB that can be written back to the iPod. There have been several requests to also register playcounts when playing a track on the local machine. This is possible by calling gtkpod as

```
gtkpod -p <filename>
```

If gtkpod is already running, the playcount of that track there will be increased by one. If gtkpod is not already running, the playcount will be registered in ~/.gtkpod/offline_playcount. This file will be read the next time you import the iTunesDB from your iPod, and the playcounts will be updated accordingly.

Please note that if you start several instances of gtkpod, only the first instance will register playcounts through "gtkpod -p".

Now all that is required is to tell your favorite player to call gtkpod with the filename of the played track.

For xmms this is possible as of July 18th, 2004 (CVS version). Versions before that (including 1.2.10) only allow you to have an external program called at the beginning of a track. Our patch allows to have an external program called also at the end of a track.

In each case you will need to activate the "General Plugin" called "song_change" and configure it to call 'gtkpod -p "%f"'.

Playlists

Playlists reside under the iPod repository node in the left hand pane of gtkpod. Clicking on them will display the tracks associated with them, effectively filtering the other gtkpod panes. You can ...

- create playlists with the button on the toolbar or pressing "Ctrl-n" in the playlist listview.
- create playlists by adding an existing playlist file with or .
- add tracks to playlists by marking the tracks you want to add, and then dragging them onto the playlist.
- rename playlists.
- delete playlists by selecting the desired playlist and pressing "Ctrl-D", or by selecting "Delete Playlist" from the Edit menu.

Podcasts

You have to download podcasts using a third party tool like bashpodder or gpodder .

Podcasts should be added directly into the 'Podcasts' playlist of the iPod repository, for example by selecting the Podcasts playlist before manually adding files/directories. Podcasts will then appear only in the Podcasts section on the iPod. If you add podcasts to the main playlist 'gtkpod/iPod' or any other iPod playlist first and then drag them over to the Podcasts playlist, the podcasts will appear in the Podcasts section on the iPod, as well as in the normal music section.

The podcast 'repository' is a local repository (like 'Local') where you can keep all local podcasts. No mechanism exists to automatically synchronize the iPod repository with the Podcast repository at this time. You have to drag the podcasts over manually.

Photos

If your ipod supports photos then a special dynamic playlist is created in the playlist tree. **YOU CANNOT DELETE IT!** Indeed when clicking on it all the usual track related ipod functions are disabled or removed.

- The track window elements are replaced with the photo management pane. This displays the Photo Library and all the saved photos from the ipod and allows you to add new albums, add new images, remove albums, remove images and drag n drop image between albums.
- You cannot delete the Photo Library as this is the original photo database. All images are stored in it and thus clicking on it, all the images on the ipod are displayed in it. Subsequent albums display a subset of the Photo Library's images.
- Removing an image from an album will remove it from the album but it will still be stored on the

ipod in the Photo Library.

- Removing an image from the Photo Library will delete it completely from the ipod.
- You can select a set of images from a selected album (inc. the Photo Library) and drag n drop them into another album.
- Any changes made in the photo management window will change the Photo playlist to italics indicating that a "Save Changes" should occur before exiting in order to preserve those changes.

Preferences File

On startup gtkpod will read the preferences from ~/.gtkpod/prefs (or /etc/gtkpod/prefs if the former doesn't exist). The file is created if preferences are changed using the Preference window.

Changing Preferences

Preferences can be changed by opending the Preference window. This is displayed by clicking on Edit > Edit Preferences or hitting Ctrl+P.

Duplicate Detection

You can instruct gtkpod (in the prefs window) to use file-size-dependent SHA1 checksums to prevent the same file from being copied to your iPod twice.

If a duplicate is detected, gtkpod will print out the the filenames of the duplicate files. If the filename of the already existing file is not available (it is not stored in the iTunes database, see "Extended Information File" below), other available information of the track is printed.

Extended Information File

Some (I believe) essential information is not stored in Apple's iTunes database. You can therefore instruct gtkpod to write an additional file (iTunesDB.ext) with extended information. For each track it stores

- SHA1 hash
- filename in the locale's encoding
- filename in UTF8 encoding
- hostname where the file was added (not used for anything yet)
- filename of an associated converted file (for example an .mp3 for a .flac file)
- if the file is present in the local database a reference to there
- in order for playcounts to work on the local database as well
- last modification time
- the charset used for the file when adding it

Since the extended information file is only valid with the corresponding standard iTunes database, a checksum of the iTunes database is also stored in the extended information file.

Using an extended information file will considerably speed up the import of an existing iTunes database when using duplicate detection,

since the SHA1 checksums do not have to be re-calculated.

Using an extended information file will also allow modification of ID3 tags in the track files after the initial import, because the full
filenames are still available.

Encoding of ID3 tags (charsets)

If you use correctly written unicode ID3V2 tags you don't have to worry about the charset setting. Otherwise you must specify the charset to be used for representing ID3 tags in the preferences menu. The default is "System Charset", which is the charset associated with the locale gtkpod is running under. If your tags are stored in a different encoding, you should set it appropriately.

Please note that if necessary you can change the charset each time you add files or directories: the iTunesDB itself is using UTF16, so once tags are imported correctly, changing the charset has no influence.

If you chose "Japanese (automatic detection)", gtkpod will try to determine if the string is in ISO-2022-JP, Shift_JIS, or EUC-JP (Hankaku Katakana (1-byte Katakana) may not be recognized correctly -- specify the correct encoding if you run into this problem). The actual encoding used for the ID tags will be stored and will be used when writing tags or doing updates/syncs. Check the "Use selected charset also when updating or syncing tracks" and "Use selected charset when writing tags" options if you want to specify a particular character set when writing or updating/syncing. The default charset is "EUC-JP" -- it will be used when the charset cannot be autodetected, as well as when writing tags if a specific charset could not be determined before.

gtkpod will recognize ID3V2 tags encoded in unicode automatically and ignore your charset setting when necessary.

Extracting tag information from the filename

Tags can also be extracted from the filename if you activate the option 'Use this template to parse filename for tag information' and supply a template that explains how the filenames are constructed.

For filenames like

```
music/new/latin1/alan_jackson - drive/01 drive_for_daddy_gene.mp3
```

you could use

```
%a - %A/%T %*.mp3
```

to extract artist, album, track number and title.

The following character sequences are supported:

- %t: title
- %a: artist
- %A: album
- %c: composer
- %t: title
- %g: genre
- %T: track number
- %C: CD number
- %*: placeholder, ignore data
- %%: the character '%'

You cannot supply a template like "%a%t.mp3" because gtkpod would not know how to separate artist and title. "%a_%t.mp3" would be correct, if artist and title are separated by an underscore. You can also omit the trailing ".mp3" if you want the template to apply to all files instead of only to mp3 files.

Startup and Shutdown Scripts

During startup and after reading the preferences file, gtkpod will try to execute

```
~/.gtkpod/gtkpod.in or /etc/gtkpod/gtkpod.in
```

Just before exiting the program, gtkpod will try to execute

```
~/.gtkpod/gtkpod.out or /etc/gtkpod/gtkpod.out
```

Thus, extra arguments can be entered into these scripts and gtkpod will run them. An example of this could be uploading data to last.fm.

Instructions on how to checkout source code from the subversion repository is available from sourceforge here. The subversion code is the latest
and cutting edge iteration and as such may contain bugs. This is really for experienced users and anyone who enjoys tinkering if a problem arises.

However, you probably do not want to checkout the entire subversion repository as it is a little weighty!

To checkout the gtkpod trunk:

- `svn co https://gtkpod.svn.sourceforge.net/svnroot/gtkpod/gtkpod/trunk gtkpod`

To checkout the libgpod trunk:

- `svn co https://gtkpod.svn.sourceforge.net/svnroot/gtkpod/libgpod/trunk libgpod`

- Script to find the firewire id and add to the sysinfo file.
- Using musicbrainz to find duplicates encoded differently. The basic idea is to use an open source tool to fingerprint a track and look up its **Portable Unique IDentifier** (PUID) online. See http://en.wikipedia.org/wiki/MusicDNS for example for links and info. Need to investigate existing open source tools.
- Integrate existing free tagging software into gtkpod, especially those that use acoustic fingerprinting such as musicdns/musicbrainz. Picard is an existing free tool for tagging that uses musicbrainz.
- ## After adding a file to the "Podcasts" repository in the iPod.
    1. I right-clicked then chose edit track details.
    2. I changed the Media Type to "Video Podcast" and hit okay.
    3. The following error message is displayed: Writing to video files not yet supported (the_pod_file.mp4). Couldn't change tags of file:/home/user/the_pod_file.mp4. Writing to video files not yet supported (the_pod_file.mp4). Couldn't change tags of file: /media/IPOD/iPod_Control/Music/F13/gtkpod613318.mp4
- If import of a flac file is performed then the converted mp3 file should be created in a detectable location. If both are added separately then the one is detected as a duplicate and not added a second time.
- Read APEv2 tags for track info. Right now APEv2 tags are read only for replaygain information that can be converted to soundcheck.
- Also read replaygain tags stored in id3v2 instead of APEv2 tags and convert them to soundcheck.
- Create a folder for each album on export to HDD (now all songs go into one folder)
- Support for conversion from WMA
- Write tag "CD number" on converted files

iPod Touch

- G1

iPod Classic

- G2
- G1

iPod Nano

- Fourth Generation - Video G2
- Third Generation - Video G1
- Second Generation
- First Generation

iPod Mobile Phones

- iPhone
- Nokia N95

iPod Shuffle

- Fourth Generation
- Third Generation
- Second Generation
- First Generation

iPod Mini

- Second Generation
- First Generation

iPod

- HP iPods
- Sixth Generation - Video
- Fifth Generation - Video
- Fourth Generation
- Third Generation
- Second Generation
- First Generation

iPod Touch G1

- First Touch Screen Only

Model Number	Capacity (GB)	Colour
A623	8	Black
A627	16	Black

iPod Classic G2

Model Number	Capacity (GB)	Colour
B562	120	Silver
B565	120	Black

iPod Classic G1

- First generation with "cover flow"

Model Number	Capacity (GB)	Colour
B029	80	Silver
B147	80	Black
B145	160	Silver
B150	160	Black

iPod Nano Video G2 (Fourth Nano Generation)

- Available in 8 GB or 16 GB capacity and comes in 9 colors.
- Distinguished from previous iPod nano models taller screen, curved surface, oval shape and inclusion of an accelerometer.
- If using an older version of gtkpod it is possible to detect and copy to this iPod using Third Nano Generation model numbers (thanks to AF Lezny). However, it should be noted that that treating a 4th gen nano like a 3rd gen nano will get you albumart only in the cover flow view. Other formats of the albumart (like in the album list) might not be visible. There may be other limitations not currently mentioned.

Model Number	Capacity (GB)	Colour

B480	4	Silver
B651	4	Blue
B654	4	Pink
B657	4	Purple
B660	4	Orange
B663	4	Green
B666	4	Yellow
B598	8	Silver
B732	8	Blue
B735	8	Pink
B739	8	Purple
B742	8	Orange
B745	8	Green
B748	8	Yellow
B751	8	Red
B754	8	Black
B903	16	Silver
B905	16	Blue
B907	16	Pink
B909	16	Purple
B911	16	Orange
B913	16	Green
B915	16	Yellow

B917	16	Red
B918	16	Black

iPod Nano Video G1 (Third Nano Generation)

- First generation of Video Support for Nano

Model Number	Capacity (GB)	Colour
A978	4	Silver
A980	8	Silver
B261	8	Black
B249	8	Blue
B253	8	Green
B257	8	Red

iPod Nano (Second Nano Generation)

- Pretty much identical to first generation Nano but with better display, extended battery operation time and gap-free playback

Model Number	Capacity (GB)	Colour
A477	2	Silver
A426	4	Silver
A428	4	Blue
A487	4	Green
A489	4	Pink
A497	8	Black

iPod Nano (First Nano Generation)

- Buttons are integrated into the touch wheel

Model Number	Capacity (GB)	Colour
A350	1	White
A352	1	Black
A004	2	White
A099	2	Black
A005	4	White
A107	4	Black

iPod Mobile Phones

Model Number	Capacity (GB)	Colour
A501	4	Black
A712	8	Black
iPhone	-	Black
First Mobile (N95?)	-	Silver

iPod Shuffle (Fourth Generation)

- Bar, button-less, speaking
- Note: Can be identified but not fully supported yet due to a different database format. Full support available soon.

Model Number	Capacity (GB)	Colour
B867	4	Silver
C164	4	Black

iPod Shuffle (Third Generation)

- Square connected to computer via cable. looks identicaly to Sixth Generation

Model Number	Capacity (GB)	Colour

B225	1	Silver
B233	1	Purple
B231	1	Red
B227	1	Blue
B228	1	Blue
B229	1	Green
B518	2	Silver
B520	2	Blue
B522	2	Green
B524	2	Red
B526	2	Purple

iPod Shuffle (Second Generation)

- Square connected to computer via cable

Model Number	Capacity (GB)	Colour
A564	1	Silver
A947	1	Pink
A949	1	Blue
A951	1	Green
A953	1	Orange

iPod Shuffle (First Generation)

Model Number	Capacity (GB)	Colour
9724	0.5	White
9725	1	White

iPod Mini (Second Generation)

Model Number	Capacity (GB)	Colour
9800	4	White
9802	4	Blue
9804	4	Pink
9806	4	Green
9801	6	White
9803	6	Blue
9805	6	Pink
9807	6	Green

iPod Mini (First Generation)

Model Number	Capacity (GB)	Colour
9160	4	White
9436	4	Blue
9435	4	Pink
9434	4	Green
9437	4	Gold

iPod Photo

- Buttons are integrated into the touch wheel

Model Number	Capacity (GB)	Colour
A079	20	White
A127	20	U2

9829	30	White
9585	40	White
9830	60	White
9586	60	White
9830	60	White

HP iPods

- Need contributions for this table. Buttons are integrated into the touch wheel.

Model Number	Capacity (GB)	Colour
E436	40	Regular
S492	30	COLOR

iPod Video (Sixth Generation)

- Pretty much identical to fifth generation with better display, extended battery operation time and gap-free playback
- upon trouble with "invisible" podcasts, upgrade your firmware (using e.g. itunes)

Model Number	Capacity (GB)	Colour
A444	30	White
A446	30	Black
A664	30	U2
A448	80	White
A450	80	Black

iPod Video (Fifth Generation)

- Buttons are integrated into the touch wheel

Model Number	Capacity (GB)	Colour

A002	30	White
A146	30	Black
A003	60	White
A147	60	Black

iPod (Fourth Generation)

- Buttons are now integrated into the touch wheel

Model Number	Capacity (GB)	Colour
9282	20	Regular
9787	25	U2
9268	40	Regular

iPod (Third Generation)

- Touch sensitive buttons and arranged in a line above touch wheel. Docking connector was introduced here same models for Mac and PC from now on

Model Number	Capacity (GB)	Colour
8976	10	Regular
8946	15	Regular
9460	15	Regular
9244	20	Regular
8948	30	Regular
9245	40	Regular

iPod (Second Generation)

- Same buttons as First Generation but are around the touch-sensitive touch wheel.
- 8737 and 8738 are Mac types
- 8740 and 8741 are PC types

Model Number	Capacity (GB)	Colour
8737	10	Regular
8740	10	Regular
8738	20	Regular
8741	20	Regular

iPod (First Generation)

- Mechanical buttons arranged around rotating "scroll wheel".
- 8513, 8541 and 8709 are Mac types
- 8697 is PC

Model Number	Capacity (GB)	Colour
8513	5	Regular
8541	5	Regular
8697	5	Regular
8709	10	Regular

Starting with the 2007 generation of iPods, libgpod needs an additional configuration step to correctly modify the iPod content. libgpod needs to know the so-called iPod "firewire id", otherwise the iPod won't recognize what libgpod wrote to it and will behave as if it's empty.

Contents:

The iPhone and iTouch

Credits Go First

This tutorial would not have been possible without the tremendous help of the people from #gtkpod. Thank you!

Global Assumptions and Scope of this Tutorial

touch_ip -- denotes the IP address that is assigned to your iPod Touch
(e.g. 192.168.0.27).

touch_mnt -- denotes the mount point to your iPod Touch (i.e. the
directory the iPod is mounted on).

To make effective use of the following you have to jailbreak your iPod Touch first (in order to access it via SSH). This document won't go into detail on this topic. One possibility to do this using a computer that runs Microsoft Windows (XP) can be found at:

http://iphone.cricblogs.com/one-click-ipod-touch-jailbreak-for-windows-by-msbasher/

Compiling libgpod & gtkpod (subversion versions)

Install and compile libgpod and gtkpod from subversion. See here for details.

Configure libgpod. The '--prefix' option will make sure that is not installed globally but locally within your home directory (note: replace '~' by the absolute path to your home directory):

~/libgpod$ ./autogen.sh --prefix=~/local

Otherwise make sure all dependencies are met ('gtk-doc' is a candidate likely to be missing, proper version of automake is another).

Repeat this process for gtkpod ('PKG_CONFIG_PATH' make sure it builds on the freshly compiled version of libgpod; again, replace '~' by absolute path to your home directory):

```
~/gtkpod$ PKG_CONFIG_PATH=~/local/lib/pkgconfig ./autogen.sh --prefix=
~/local

~/gtkpod$ make && make install
```

Assuming that everything went well you should be able to start gtkpod:

```
~$ LD_LIBRARY_PATH=~/local/lib ~/local/bin/gtkpod
```

You won't be able to fully use it at this point so close it for now.

Determining the Firewire GUID

Connect your iPod Touch via USB (to your Linux machine) and issue the command:

```
sudo lsusb -v | grep -i iSerial
```

This should print something similar to this:

```
iSerial 3 fd98145617c113dc15ab151601001ff12354b139
[ ... ]
```

The first 16 characters constitute the FirewireGUID, i.e. in this example the FwGUID is:

```
fd98 1456 17c1 13dc
```

For FreeBSD, there is a tool for this, since usbdevs does not provide enough output:
http://50hz.ws/dev/getserial.c

Making libgpod Aware of the Firewire GUID

Mount your iPod Touch via:

```
sudo sshfs root@touch_ip:Media touch_mnt/ -o allow_other
```

Go to the directory 'iTunes_Control' and create (if it does not already exist) a directory named 'Device':

```
~/touch_mnt/iTunesControl$ mkdir Device; cd Device
~/touch_mnt/iTunesControl/Device$ echo "FirewireGuid: 0xfd98145617c113
dc" > SysInfo
```

Test if libgpod is able to retrieve the FWGUID by changing into the 'tests' subdirectory of libgpod's source directory and runnning:

```
./test-firewire-id touch_mnt/
```

gtkpod Revisited

With your iPod Touch still mounted start up gtkpod once more:

```
~$ LD_LIBRARY_PATH=~/local/lib ~/local/bin/gtkpod
```

You should now be able to sync songs to your iPod Touch. After syncing is complete go into "Music" on your iPod Touch: The songs you have just synced won't show up at this point. To make them appear press and hold the "Home"-Button until the springboard is reloaded. Going to "Music" now will show your music.

Don't forget to unmount it after you're completely done.

Tutorial compiled by staciemonroe (tobias kreisel gmail com)

The Classic and Nano3g

There are two ways to set up the iPod to make libgpod able to find its firewire id.

The 1st one is mostly automated. First, make sure you have libsgutils installed before running

configure/autogen.sh. If you built libgpod without it, install it and run configure/make/make install. You should now have an ipod-read-sysinfo-extended tool available. Run it with the iPod device path (eg /dev/sda) and the iPod mount point (eg /mnt/ipod) as arguments. This may require root privileges. ipod-read-sysinfo-extended will read an XMLfile from the iPod and write it as /mnt/ipod/iPod_Control/Device/SysInfoExtended. See http://ipodlinux.org/Device_Information for more details about the method used.

Having that file is enough for libgpod to figure out the iPod firewire id.

The 2nd method requires more manual intervention. First, you need to get your firewire id manually. To do that, run "sudo lsusb -v | grep -i Serial" (without the "") with your iPod plugged in, this should print a 16 character long string like 00A1234567891231. For an iPod Touch, this number will be much longer than 16 characters, the firewire ID is constituted by the first 16 characters.

For FreeBSD, there is a tool for this, since usbdevs does not provide enough output: http://50hz.ws/dev/getserial.c

Once you have that number, create/edit /mnt/ipod/iPod_Control/Device/SysInfo (if your iPod is mounted at /mnt/ipod). Add to that file the line below:

```
FirewireGuid: 0xffffffffffffffff
```

(replace ffffffffffffffff with the string you obtained at the previous step and don't forget the trailing 0x before the string)

Save that file, and you should be all set. Be careful when using apps which lets you manually specify which iPod model you own, they may overwrite that file when you do that. So if after doing that libgpod still seems to write invalid content to the iPod, double-check the content of that SysInfo file to make sure the FirewireGuid line you added isn't gone. If that happens, read it to the end of the file, and make sure libgpod rewrite the iPod content.

Once that is done, if you compiled libgpod from source, you can test that libgpod can find the firewire ID on your iPod by running

```
libgpod/tests/test-firewire-id /ipod/mount/point
```

Edit Track Details

The Edit Track Details window can be displayed from a couple places.

- Select a track from the Track window, right-click and select "Edit Track Details" from the popup.
- Make a selection in any of the categories, eg. Album, Genre, and right-clicking "Edit Track Details will display the window and allow all tracks in that category to be edited.
- Right-clicking "Edit Track Details" from the popup menu on the CoverArt Display displays the window , allowing all tracks belonging to the album to be edited.

Edit Track Details Window

- The window allows multiple tracks to be edited at the same time. By selecting the "Change all Tracks Simulaneously" check box **BEFORE** making any edits, all the selected tracks can be updated with the new information.
- The coverart can be dragged n dropped on from a web browser in the same way as the Cover Art Display. However, dropping a new cover onto this window will update the cover art of the tracks and not the album! This is an important distinction as it is possible to have different cover art for tracks in the same album! (why anyone would but its possible)!

Refresh (Update) Track Info From File

If you have changed the ID3 tags of your original file, you can update the ID3 tags stored in gtkpod without removing and re-adding the track. Simply select the track to be updated and press "Ctrl-u" or choose "Update Track Info From File" from the Edit menu. Since gtkpod needs to know the filename of the track, the "Extended Information File" (see here) is needed.

"Update" will also re-read the replay-gain tags from disk, if available, and set the soundcheck field. If no replay-gain tag is available, the soundcheck field will be erased. You can also select entries in the filter tab or entire playlists for refresh.

Synchronize with Directories

If you have added files to directories or changed files in directories you have previously added tracks from, you can use the "Synchronize Dirs" utility to update your iTunesDB.

"Synchronize with Dir(s)" will use the selected tracks to make a list of directories to update, so you should activate the "Write Extended Information" option in the export section of the Preferences dialogue.

It will then add all non-existing tracks in those directories and update all existing tracks. The tracks are also added to the currently selected playlist, if they aren't already a member.

Tracks that have been removed from the directory will only be removed from the iTunesDB if this option

is checked in the option dialogue.

For best results you should also activate duplicate detection. This avoids unnecessary copying of unchanged tracks.

Volume Normalization

There are two fields in the iTunes Database (iTunesDB) that allow for the adjustment of the volume of an individual track:

- the volume field (a signed integer)
- the soundcheck field (in dB).

The volume is always active, whereas the soundcheck field is only active when you select 'soundcheck' on the iPod. Further, the soundcheck field only affects the earphone output but not the line output of the iPod.

gtkpod will set the soundcheck according to the replay-gain tag set in your mp3 file. Newer versions of 'lame' automatically include the replay-gain tag when encoding. In that case the soundcheck field will be filled in when you first import a track.

If no replay-gain tag is set, you can use the function 'Normalize Volume' to call mp3gain (mp3gain.sourceforge.net) to calculate the gain and write a replay-gain tag. Since this procedure is very time consuming, it is not done automatically during import. You need to install mp3gain in the default path or set the full path in the 'Tools' section of the preferences dialog. If the iPod is connected, the tag is written to the file stored on the iPod.

At this time "album gain" functionality is not supported. "Album gain" means that the volume of all tracks of one album is adjusted by the same gain, such that the relative volume level remains the same. It is planned to realize this in one of the next versions.

Also, please be aware that tracks are not normalized on a 'per playlist' fashion. If a track is normalized, it's normalized in all playlists it is a member of.

Once the replay-gain tag has been read, it is stored in the extended information file 'iTunesDB.ext'. When you call 'Normalize volume' again, the stored value will be used to re-populate the soundcheck field. Use 'Update Track' to re-read the tag from the file.