

gtk-desktop-info Guide

Author: Mark Buck (Kaivalagi)



Rhythmbox

3:10 / 4:07 (76%)

Title
Frienemy Dub

Artist
Soldiers Of Jah Army

Album
Dub In A Time Of War

Album cover: **LET US A TIME OF WAR SOLDIERS OF JAH ARMY**

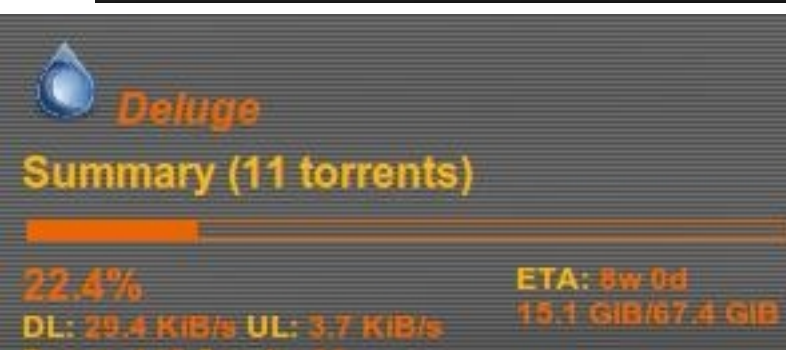
★★★★★



E-Mail

Yahoo: 0
Google: 4 New

1. Bruce & Lilliana: "Title: *Am... (unreadable)*"

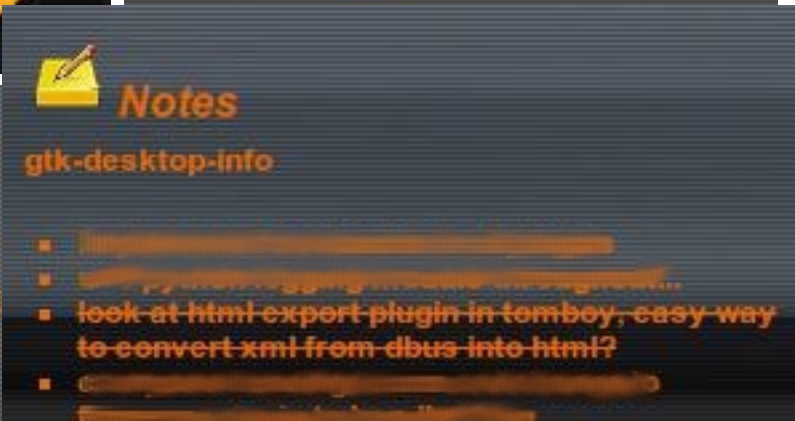


Deluge

Summary (11 torrents)

22.4%
DL: 29.4 KiB/s UL: 3.7 KiB/s
Peers: 112 Seeds: 90

ETA: 8w 0d
15.1 GiB/67.4 GiB



Notes

gtk-desktop-info

- look at html-export plugin in tomboy, easy way to convert xml from dbus into html?



74.27% Downloading
DL: 3.1 KiB/s UL: 0.0 KiB/s
ETA: 5d 13h
4.0 GiB/5.4 GiB

65.03% Downloading
DL: 14.0 KiB/s UL: 0.0 KiB/s
ETA: 1d 5h
2.7 GiB/4.1 GiB

49.85% Downloading
DL: 0.0 KiB/s UL: 0.0 KiB/s
ETA: Infinity
3.0 GiB/5.9 GiB



Calendar

Title: Restructure
Starts: Wed 03/12/08 08:30:00
Ends: Wed 03/12/08 10:00:00
Where: Bure Training Room
Who: kaivalagi

Title: Heroes
Starts: Wed 03/12/08 21:00:00
Ends: Wed 03/12/08 23:00:00
Where: Home
What: Heroes on BBC2
Who: kaivalagi

Title: Mark & Maraia's Anniversary
Starts: Thu 04/12/08
Ends: Fri 05/12/08
Who: kaivalagi

Table of Contents

Introduction.....	3
Limitations.....	3
Command Line Help.....	4
Main Functionality.....	5
Plugins (--plugin="plugin").....	5
Templates (--template="template file" , --headertemplate="header template file").....	6
CSS Styles (--css="css stylesheet file").....	6
Colours (--colour="colour").....	6
Configuration.....	7
Gotchas.....	8
Compiz Shadows.....	8
Example Usage.....	9

Introduction

gtk-desktop-info is a python tool to display various pieces of information directly on the desktop, using html templates and css style sheets for formatting.

The application has been created off the back of existing python scripts used with Conky. The reason for it's creation is a simple one, Conky is great but formatting can suck sometimes...html seemed the obvious choice of formatting giving the user the ability to construct output in a variety of styles based on understood techniques.

The app works fairly simply, it can be broken down as follows:

- The main python module "gtk_desktop_info.py" manages the creation of a gtk/webkit based browser view on the desktop. This creates a cropped background image on the fly from the system wallpaper, and defines it as the html background, simulates a transparent desktop based window. It also handles the refreshing of information based on the defined interval.
- The various python plugin modules, such as "plugin_deluge.py" provide html content to display in the main window. These all work based on:
 - A config file defining any necessary settings
 - A header template file and a main template file defining the output and it's arrangement, within a html document.

There are alternatives out there which can be used such as Conky, Screenlets, Desklets etc, and these will probably provide a simpler and suitable solution for most people. However, although this application may prove a little difficult to master, it has some key benefits:

- What ever can be done in html/flash/javascript can be delivered to the desktop
- Any content can be totally customised to suit individual needs
- More plugins can be added relatively easily, if developed and sent to the main author
- It is all encompassing, meaning all possible plugins are available in the one package
- It has a keen developer, willing to put time and effort into it. A testimony of this would be the Conky Scripts previously created.

Limitations

Currently the script only handles nautilus based wallpaper settings automatically (default Ubuntu setup). And even then, if tiling or simple background colours are used, the application will no handle these well.

I am attempting to find a better way to handle system wallpaper, independently from the window manager type, so that the only thing this application will be dependant on is gtk, webkit and python libraries. However this may be some time off. If you have issues with wallpapers and the fake transparency is not working, I suggest you use the `-wallpaper` option, as described on the next page.

Command Line Help

As the application is command line based, the most obvious place to start is with the command line options. These are detailed below and can be recalled at any time by running the main application with the `-h` or `--help` option. Most are explained in enough detail below, more information is available on some further on in the guide.

```
Usage: gtk-desktop-info [options]
Options:
-h, --help            show this help message and exit
--url=URL             If set a plugin isn't used, instead the url's html is
                    displayed
--plugin=NAME        default:[shell] The plugin to use, the following are
                    currently supported: banshee, deluge, email, exaile,
                    feedparser, forecast, googlecalendar, googlereader,
                    null, pidgin, processinfo, rhythmbox, shell,
                    sysgraph(not working), tomboy, twitter
--css=FILE           Override the css stylesheet file path to use for html
                    output
--colour=NAME        default:[default] Override the default css stylesheet
                    colour, if css is overridden this option will not
                    apply. Colour names supported are: default, white,
                    black, green, blue
--noheader           Turn off header output. This will override any header
                    template setting to be nothing
--overridehead=FILE Override the default html head construction of the
                    app, opting for a user based template instead.
--headertemplate=FILE
                    Override the html body header template for the plugin,
                    default or config based template ignored.
--template=FILE     Override the html body template for the plugin,
                    default or config based template ignored.
--config=FILE       Override the config for the plugin.
--interval=SECONDS The refresh interval of the output, If zero no refresh
                    occurs. Floats are supported so 0.25 is 250
                    milliseconds
--width=NUMBER      default:[400] The required width
--height=NUMBER     default:[300] The required height
--x=NUMBER          default:[0] The required horizontal placement from the
                    left
--y=NUMBER          default:[0] The required vertical placement from the
                    top
--allworkspaces     If set output will be shown on all workspaces, rather
                    than the current one
--internalbrowsing  Bypass the webview navigation event, loading the url
                    in the system browser instead
--wallpaper=FILE    Override the wallpaper filepath in case it isn't found
                    automatically
--noresize          Turn off resizing of the wallpaper used/found. This
                    will override default resizing to screen res, allowing
                    the app to have an alternative background just for
                    itself etc
--windowmanager=NAME
                    default:[gnome] Tell the app which windows manager is
                    in use, options are gnome, kde3, kde4, xfce4, compiz
                    (compiz option is not implemented yet!)
--backgroundblend=NUMBER
                    default:[0] The optional blending ratio for the
                    backgroundcolour which is set. The default of 0 does
                    no blending, the maximum of 100 will blend "out" all
                    the wallpaper. Use this option to give some visual
                    separation to the output from the wallpaper.
--backgroundcolour=RGBValues
                    default:[0,0,0] The optional colours to use for
                    blending into the background. They should be input in
                    the form "R,G,B". Not used if --backgroundblend=0
--backgroundcorner=NUMBER
                    default:[10] The optional corner radius (in pixels) to
                    use when applying background blending. Not used if
                    --backgroundblend=0
--verbose           Outputs verbose info to the terminal (and logfile if
                    setup) for both the main app and associated plugins
--version           Displays the version of the script.
--logfile=FILE     If a filepath is set, the main app and any associated
                    plugins log errors to the filepath. If --verbose is
                    set, verbose info will be logged to file.
```

Main Functionality

Plugins (`--plugin="plugin"`)

All the plugins have come from my conky tinkering of the past, and have been adapted to output html based information, including weather/moon/wind icons for forecasts, and coverart for music. Plugins are defined using the `-plugin` option of the main application.

Below is a summary of each plugin.

- **"banshee"** - this plugin displays Banshee based song information, cover art, progress etc
- **"deluge"** - this plugin provides a breakdown of bit torrent information, when bit torrents are managed by Deluge. It can provide detailed and/or summary info.
- **"email"** - this plugin provides email account information, a count of unread emails and optionally details of the sender and subject of the emails.
- **"exaile"** - this plugin displays Exaile based song information, cover art, rating, progress etc
- **"feedparser"** - this plugin provides rss/atom feed information
- **"forecast"** - this plugin provides weather forecast information from weather.com for a given location. It includes weather, moon and wind images, detailing the weather. There is current support for the following locales: bg=bulgarian, cs=czech, de=german, es=spanish, en=english, es=spanish, fj=fijian, fr=french, it=italian, nl=dutch, pl=polish, ro=romanian, sk=slovak
- **"googlecalendar"** - this plugin provides Google Calendar event information.
- **"googlereader"** - this plugin provides Google Reader feed information.
- **"null"** - this plugin provides a means to use html, javascript and flash only content. By default is displays a javascript clock.
- **"pidgin"** - this plugin provides pidgin buddy information, telling you who is online and what their status is.
- **"processinfo"** - this plugin provides process information, telling you a processes id, cpu usage and memory usage. The information can be sorted by cpu or memory usage.
- **"rhythmbox"** - this plugin displays Rhythmbox based song information, cover art, rating, progress etc
- **"shell"** - this plugin provides a means to execute scripts and commands from within a template. Whatever can be done on the command line can be done here too.
- **"tomboy"** - this plugin provides information on Tomboy based notes, keeping most formatting from the notes intact in the output.
- **"twitter"** - this plugin provides information on all your twitter friends, shown newest to oldest

Templates (*--template="template file" , --headertemplate="header template file"*)

Each plugin has a default template which is used when no template option is provided. These are basically html documents with options in square brackets e.g. [...options...] , which define the information a plugin will retrieve and output. You can set your own templates up using the `--template` option.

By default the standard templates can be found here: `/usr/share/gtk-desktop-info/templates`.

Each template has detailed notes in it explaining the options available, I will not attempt to explain every little detail in this guide, open the template file and have a read!

Some of the template files will need modifying before the plugin will work as required. These are listed below for your convenience:

- **Email** - the email template holds login info for the emails to be retrieved, this will need copying and editing with the correct details entered. This new template can then be pointed to using the `--template` option of the main application.

CSS Styles (*--css="css stylesheet file"*)

Each plugin has a default css style which is used when no css option is provided. You can set your own css styles up using the `--css` option.

By default the css styles can be found here: `/usr/share/gtk-desktop-info/styles`

These would be a good place to start if wanting to customise your output format.

Colours (*--colour="colour"*)

Each plugin's default style is orange/yellow, to use the subset of alternative colour css styles the `--colours` option can be used. Currently blue, green, white and black are available. Again, if alternative options are wanted, a custom `--css` option can be used instead to utilise your own css style file.

Configuration

A plugins config file holding the default settings is copied into `~/.config/gtk-desktop-info` when a plugin is first run. These are named based on the plugin names. You can set your own configs up using the `-config` option.

If you want to open a config before a plugin has been run, you can find them here: ***/usr/share/gtk-desktop-info/config***

Each config has detailed notes in it explaining the reason for each setting, I will not attempt to explain every little detail in this guide, open the config file and have a read!

Some of the config file will need modifying before the plugin will work as required. These are listed below for your convenience:

- **GoogleCalendar** - the google calendar config will need editing to hold the login details for the google account used.
- **GoogleReader** - the google reader config will need editing to hold the login details for the google account used.
- **Forecast** - the forecast config will need editing to detail the location code required for weather output.

Gotchas

Compiz Shadows

Compiz shadows can cause some unwanted side effects effectively losing the transparent view of the windows on the desktop. To remove shadows from any output from this application, follow these instructions:

1. Open the “CompizConfig Settings...” dialog from the preferences menu or run “ccsm”
2. Go to the “Window Decoration” settings
3. Place the following text into the “Shadow windows” setting:
!(class=gtk_desktop_info.py)
e.g.



Example Usage

I have included an example script in `/usr/share/gtk-desktop-info/example` called `gtk-desktop-info.sh`. It is based on a screen resolution of 1680x1050. If used with a desktop where the resolution is not high enough errors will occur.

Notice below that the width, height, x and y values are defining the location and sizes of each plugins output. I am using the `--allworkspaces` option to display on all my virtual desktops, and a common `--logfile` setting too so all logging happens in the one place.

```
#!/bin/sh
# example use of gtk-desktop-info, for a screen resolution of 1680x1050
# will display on all desktops, and log to a single logfile
# Note: that if used on a smaller screen bad cropping of wallpaper can take play
# causing unwanted side effects
#####

# kill all instances of the app before starting new ones
pkill -f gtk_desktop_info.py

# wait for a sec
sleep 1

# use the shell plugin to display the default output of date and time
# update every second
gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=1 --width=130 --height=50 --x=1560
--y=30 --plugin=shell &

# use the null plugin to display the default output of an analog clock
# No interval as no refresh needed (coolclock: http://randomibis.com/coolclock/)
gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --width=120 --height=120 --x=1570 --y=80
--plugin=null &

# use the email plugin to display email details
# This will error until proper details are provided in a template file (see guide for info), update every 5
minutes
gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=300 --width=280 --height=170
--x=1280 --y=30 --plugin=email &

# use the googlecalendar plugin to display the default of the next 7 days events
# This will error until proper details are provided in a config file (see guide for info), update every 30 minutes
gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=1800 --width=400 --height=780
--x=1280 --y=200 --plugin=googlecalendar &

# use the rhythmbox plugin to display song information
# update every second
gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=1 --width=300 --height=300 --x=980
--y=30 --plugin=rhythmbox &

# use the forecast plugin to display weather info
# The default location is mine, the config file requires updating for custom locations (see guide for info),
update every 30 minutes
gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=1800 --width=300 --height=400
--x=980 --y=330 --plugin=forecast &

# use the pidgin plugin to display buddy information
# update every minute
gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=60 --width=300 --height=250
--x=980 --y=730 --plugin=pidgin &

# use the deluge plugin to display bit torrent information
# update every 10 seconds
gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=10 --width=300 --height=500
--x=680 --y=30 --plugin=deluge &

# use the tomboy plugin to display current notes
# update every 30 minutes
gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=1800 --width=300 --height=400
--x=680 --y=530 --plugin=tomboy &
```

PTO

```
# NOT ENOUGH ROOM FOR THE BELOW, SO COMMENTED OUT :)

# use the feedparser plugin to display rss feed info
# update every 30 minutes
#gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=1800 --width=300 --height=400
--x=680 --y=530 --plugin=feedparser &

# use the twitter plugin to display statuses of friends
# update every 30 minutes
#gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=1800 --width=300 --height=400
--x=680 --y=530 --plugin=twitter &

# use the processinfo plugin to display process information
# update every 5 seconds
#gtk-desktop-info --allworkspaces --logfile=/tmp/gtk-desktop-info.log --interval=5 --width=300 --height=400
--x=680 --y=530 --plugin=processinfo &
```