EXHIBIT 9A

BOOKMARK THIS SITE | SEARCH | Keywords | GO | DOWNLOAD BASKET | YOUR ACCOUNT

SOFTPEDIA®
Updated one minute ago

TODAY'S NEWS: Developers Says Facebook Games Need to Adopt Hardcore Mechanics

WINDOWS | GAMES | DRIVERS | MAC | LINUX | SCRIPTS | MOBILE | HANDHELD | NEWS

PROGRAM FINDER

LINUX CATEGORIES:

- Adaptive Technologies
- Adobe AIR Apps
- Artistic Software
- Communications
- Database
- Desktop Environment
- Documentation
- Education
- Games
- Home Automation
- Information Management
- Internet
- Multimedia
- Office
- Printing
- Programming
- Religion
- Science
- Science and Engineering
- Security
- System
- Terminals
- Text Editing&Processing
- Utilities

GLOBAL PAGES >>
NEWS ARCHIVE >>
SOFTPEDIA REVIEWS >>
MEET THE EDITORS >>

I ❤ SOFTPEDIA

Find us on Google+

@softpedia

Home > Linux > Desktop Environment > Tools

GetCodecs 1.0.1

Download

DOWNLOAD

OR ADD TO DOWNLOAD BASKET

no screenshots available

Downloads: 2,146 | View global page NEW! | Tell us about an update

User Rating: NOT RATED
Rated by: 0 user(s)

Developer:	gLaNDix | More programs
License / Price:	GPL / FREE
Last Updated:	September 24th, 2005, 18:49 GMT
Category:	ROOT / Desktop Environment / Tools

Read user reviews (0) | Refer to a friend | Subscribe

CLICK HERE TO ADD YOUR COMMENT/REVIEW

GetCodecs description

GetCodecs is a simple applet written in python with a GTK2 GUI.

GetCodecs is a simple applet written in python with a GTK2 GUI that downloads, installs, and configures your system to use the various multimedia formats (MP3, DivX, DVD, etc) that are not shipped with many distributions (such as RedHat).

Misc Audio/Video Support:

The following codecs are provided by the "Misc Codecs" package. All codecs are copyright their respective owners. The author takes no responsibility in how you use them. You must agree to each of the license agreements before installing or using these codecs.

- Indeo Video 3.2/4.1/5.0/4.1 quick/5.0 (http://www.ligos.com/indeo)
- DivX ;-) video, ver. 3.11 alpha (http://divx.ctw.cc)
- DivX ;-) DirectShow decompression filter, v3.11 alpha (http://divx.ctw.cc)
- Microsoft MPEG-4 video, beta version 3.0.0.2700
- Microsoft WMV 7 (http://codecs.microsoft.com/codecs/i386/wmvds32.cab)
- Cinepak video
- Microsoft ADPCM video
- Intel 263 video (http://members.aol.com/SlavTrainr/STsPage.html)
- Morgan Multimedia Motion JPEG video v2.0 (http://www.morgan-multimedia.com)
- ATI VCR-2 video (http://www.ati.com)
- DivX ;-) WMA audio (http://divx.ctw.cc)
- Intel Music (http://members.aol.com/SlavTrainr/STsPage.html)
- Voxware Metasound audio (http://codecs.microsoft.com/codecs/i386/voxmsdec.cab)
- ACELP.Net DirectShow audio

TAGS: python applet | get codecs | panel applet | GetCodecs | python | applet

SHARE THIS
TWEET THIS

HTML code for linking to this page:

```
<a href="http://linux.softpedia.com/get/Desktop-Environment/Tools/GetCodecs-5182.shtml"><img border="0" src="http://www.softpedia.com/im
```

copy

Go to top

WEEK'S BEST

- BackTrack 5 R1
- Wine 1.2.3 / 1.3.33
- Mozilla Firefox 8....
- Ubuntu 11.04
- Angry Birds 1.1.2
- Ubuntu 10.04.3 LTS
- Linux Kernel 3.1.2
- Ubuntu Manual 10.10
- Adobe Flash Player...
- Pidgin 2.10.0

WINDOWS GAMES DRIVERS MAC LINUX SCRIPTS MOBILE HANDHELD NEWS

SUBMIT PROGRAM | ADVERTISE | GET HELP | SEND US FEEDBACK | RSS FEEDS | UPDATE YOUR SOFTWARE | ROMANIAN FORUM

Copyright Information | Privacy Policy | Terms of Use

Search Go

EXHIBIT 9B

Gnome applets with Python

Arturo González Ferrer

<arturogf@ugr.es>

Lorenzo Gil Sánchez

<lgs@sicem.biz>

Revision History

Revision 0.5	07 February 2004	
Revision 0.4	23 January 2004 (added "change-orient" event to callbacks section)	Revised by: arturogf
Revision 0.3	20 January 2004 (merge 0.2 with the new version. Some updates on bonobo and .server files. Autotools installation).	Revised by: arturogf
Revision 0.2	15 January 2004 (some grammar and structure corrections)	Revised by: lgs
Revision 0.1	07 January 2004 (first release)	Revised by: arturogf

Table of Contents

1. Introduction
- 1.1. The applet code skeleton
- 1.2. The bonobo .server file

2. Running the applet in its own window

3. Moving on: Object Oriented Applets
- 3.1. The class definition

4. The import section

5. The class constructor: __init__
- 5.1. Some instance variable definitions
- 5.2. The GTK widgets definition
- 5.3. Optional timeout callback method
- 5.4. Connecting the "destroy" signal and show the applet
- 5.5. Connecting signals and events to handlers
  - 5.5.1. Loading the interface with `glade.XML()`
  - 5.5.2. Using the `gtkWidget.connect()` method

6. The callback methods
- 6.1. The timeout callback
- 6.2. The panel applet "change-orient" signal

7. Autotools : a brief introduction to the installation process
- 7.1. The project structure: directories and files needed
- 7.2. Creating these files
- 7.3. Editing the files Makefile.am and configure.in

8. Example applet

Bibliography

This article will cover how to make an applet for the GNOME 2.x desktop with the Python programming language. Usually, GNOME applets are made in C, which has the big advantage of generating compiled executables (this means less memory footprint and faster programs) but in the other hand it's more

difficult to do it and it takes more time to write the code. Python programs are developed faster, they have fewer lines of code, and it's an object oriented programming language. But it's an interpreted language, so the execution speed can be worse. It's up to the reader to decide if it is worthy writing a GNOME applet in Python. One good aproach is to prototype the applet in Python and then write the final version in C. Here we'll explain the necessary steps to write and deploy a simple applet from scratch using Glade 2 and libglade for the dialogs. Look at the bibliography for an in-deep explanation. Another good source of information is the pyGTK mailing list archives.

However, with regard to the code shown at the tutorial, we advise you to better look at the example section, where you will find some releases that do like it has been explained here, so you can see it growing from a prototype to a really functional applet.

1. Introduction

GNOME 2.x applets need at least two files to get them working:

- The executable (in this case the applet code)
- The bonobo server component, that must be placed at `/usr/lib/bonobo/servers` by default. This file contains all the information about the resources the applet will use for its execution.

1.1. The applet code skeleton

The simplest applet we can write can be found in the gnome-python package documentation (`/usr/share/doc/python2.x-gnome2/examples/applet/applet.py` in DEBIAN-based distributions) or simply in the gnome-python tarball from the original GNOME ftp.

☒ Warning `place this at /usr/bin/pysample.py`

```
#!/usr/bin/env python
import pygtk
pygtk.require('2.0')

import gtk
import gnome.applet

def sample_factory(applet, iid):
    label = gtk.Label("Success!")
    applet.add(label)

    applet.show_all()
    return gtk.TRUE

gnome.applet.bonobo_factory("OAFIID:GNOME_PysampleApplet_Factory",
                             gnome.applet.Applet.__gtype__,
                             "hello", "0", sample_factory)
```

First, we need to import some Python modules that we will need. The *pyGTK* module, needed to specify the GTK version used (we'll be using 2.x in this article), the *gnome* module, that contains all the useful classes and methods about the gnome desktop, i.e. the applet class, and the *gtk* module, Python bindings for the GTK toolkit.

Then, we define the *sample factory* function to generate applet objects, with a text label "Sucess!". This function receives the object to be initializated (the applet) and the bonobo activation ID that the new factory will implement. It's all that simple. It returns gtk.TRUE if no errors were reported (see the panelapplet reference manual for details). When we call to `gnome.applet.bonobo_factory()`.

When we call the `bonobo_factory` method, we need to pass it the following arguments:

1. iid: The bonobo-activation iid of the factory.
2. type : the type of the created object.
3. description.
4. version.
5. factory callback: the name of the factory method.

1.2. The bonobo .server file

The second thing we need to get the applet running is to construct the bonobo server file. `bonobo-activation-server`, the GNOME application that tracks information about installed components and brokers components, reads and mantains the component descriptions from `/usr/lib/bonobo/servers/*.server`. These files provide an XML description of a component's capabilities which can be queried and manipulated by clients using the activation client library. `Bonobo-activation-server`, also ensures that the minimum neccessary number of servers for your display setup are running.

Bonobo activation server is nothing but a daemon implementing a set of CORBA interfaces. These CORBA interfaces implement a name service for the set of CORBA servers installed on your system. GNOME Object Activation Framework daemon (OAFD) knows about all the CORBA servers in your system, running or not. The OAF daemon will activate those servers if you ask for them. Each server is described by its .server file which contains among other things the IDL interfaces it implements, some specific properties and an IID (Implementation ID). Each IID has to be globally unique, and its format is pretty simple:

```
OAFIID:program_name:UUID
```

The bonobo factories are CORBA objects that allow the creation of other new CORBA objects. This is a common practice in GNOME. The use of factories will allow us to use only an executable to get several instances of the component.

So we need to define the .server file and place it at the location mentioned before:

```
<oaf_info>
<oaf_server iid="OAFIID:GNOME_PysampleApplet_Factory"
            type="exe" location="/usr/bin/pysample.py">

        <oaf_attribute name="repo_ids" type="stringv">
                <item value="IDL:Bonobo/GenericFactory:1.0"/>
                <item value="IDL:Bonobo/Unknown:1.0"/>
        </oaf_attribute>
        <oaf_attribute name="name" type="string" value="Python applet example"/>
        <oaf_attribute name="description" type="string" value="Python applet example"/>
</oaf_server>

<oaf_server iid="OAFIID:GNOME_PysampleApplet"
            type="factory" location="OAFIID:GNOME_PysampleApplet_Factory">

        <oaf_attribute name="repo_ids" type="stringv">
                <item value="IDL:GNOME/Vertigo/PanelAppletShell:1.0"/>
                <item value="IDL:Bonobo/Control:1.0"/>
                <item value="IDL:Bonobo/Unknown:1.0"/>
        </oaf_attribute>
        <oaf_attribute name="name" type="string" value="Python applet example"/>
        <oaf_attribute name="description" type="string" value="Python applet example"/>
        <oaf_attribute name="panel:category" type="string" value="Utility"/>
        <oaf_attribute name="panel:icon" type="string" value="bug-buddy.png"/>
</oaf_server>
</oaf_info>
```

The .server file registers two unique OAF identifiers, and give the activation-server the description of how the objects must be created. The first one is the Factory, that is created when the .py script is executed. The second is the applet itself, and is created asking the Factory how to do it. When we do the "add to panel" action in GNOME, it takes the OAF identifiers that are supposed to implement the "IDL:GNOME/Vertigo/PanelAppletShell:1.0" in its "repo_ids" attribute. The submenu is determined by the "panel:category" attribute, the icon from "panel:icon", the text displayed at the menu comes from "name" and the tooltip text from "description".

The `sample_factory()` function is called when a new applet is created. It receives a container that should be populated with the applet contents.

Next
Running the applet in its own window

EXHIBIT 9C

New applet: Desktop Drawers

Ubuntu Dev Link Forum
This forum allows developers to ask questions and have the user base reply.

New Reply

Thread Tools Display Modes

September 3rd, 2008 #1

ratl3
First Cup of Ubuntu

Join Date: Jun 2005
Beans: 5

New applet: Desktop Drawers

Hello, I have created a gnome panel applet to solve an issue that has bugged me for a while now. I was wanting to know if anyone else would like to use it or has any input.

It is an extension of the show desktop applet with some added features. It is a drop down menu that includes the ability to show the desktop, show the contents of the desktop, change desktop to a different folder, add a new desktop, and manage current desktops.

My current drawers drop down looks like this:

- Show Desktop
- Contents >
- -------------
- Desktop Drawers
- Kids
- Temp
- Website
- ----------
- Add Desktop
- Manage Desktops

I created this because I noticed that I was using the desktop for the projects I was currently working on. I needed a way to change the files on the desktop according to the project being worked on and originally would just link the desktop folder to a project folder. I created this panel applet to automate this process.

(Update)
NEW IN 0.4:
Addition of multiple new features and bug fixes. The desktop is now changed using the ~/.config/user-dirs.dirs file instead of symlinking the Desktop dir to the folder. This fixes Recent Document displays in user programs as well as allowing copy and paste between desktops. Recent Documents Desktop added to the list. This allows one to have the recently accessed documents show up on their desktop. Also, a Preferences window has been added to allow one to customize Drawers to their liking.

I have created a launchpad page and a PPA for this project at:

https://launchpad.net/desktopdrawers
https://launchpad.net/~ryanhjefferson/+archive

To add this to the list of apt repositories add the following entry to your /etc/apt/sources.list file:

Code:

```
deb http://ppa.launchpad.net/ryanhjefferson/ubuntu hardy main
```

After this has been added to the sources.list file and everything has been updated one can just apt-get install desktopdrawers to install Desktop Drawers.

Please leave a message if you use Desktop Drawers.

Thanks

Last edited by ratl3; September 20th, 2008 at 12:30 AM..

QUOTE

September 5th, 2008 #2

mujambee
Just Give Me the Beans!

Join Date: Aug 2008
Location: Madrid
Beans: 52

Re: New applet: Desktop Drawers

I like the concept. May I have a look at it?

QUOTE

September 5th, 2008 #3

ratl3
First Cup of Ubuntu

Join Date: Jun 2005
Beans: 5

Re: New applet: Desktop Drawers

I have attached two files that are needed to run Drawers, drawers.py and desktop_drawers.server.txt. The python file needs to be copied to the /usr/local/bin directory and the desktop_drawers.server.txt needs to be renamed to desktop_drawers.server and copied to the /usr/lib/bonobo/servers/ directory. One will also need to install some software to run this applet. The requirements for Drawers are python, wmctrl, and xmacroplay. All of these are in the ubuntu repositories. If it seems like I am receiving any more interest I will set up a launchpad account with a deb repository for Drawers. I hope you find this as useful as I do.

Attached Files

- drawers.py (9.1 KB, 8 views)
- desktop_drawers.server.txt (1.1 KB, 7 views)

QUOTE

September 5th, 2008 #4

ratl3
First Cup of Ubuntu

Join Date: Jun 2005
Beans: 5

Re: New applet: Desktop Drawers

I should give a warning before anyone uses this applet. When it is first run it will move your current desktop to the folder ~/.Desktops/Original Desktop and link ~/Desktop to that folder. One should back up their current Desktop directory just in case something goes wrong.

QUOTE

September 11th, 2008 #5

ratl3
First Cup of Ubuntu

Join Date: Jun 2005
Beans: 5

Re: New applet: Desktop Drawers

I have created a PPA for this project at:

https://launchpad.net/~ryanhjefferson/+archive

To add this to the list of apt repositories add the following entry to your /etc/apt/sources.list file:

Code:

```
deb http://ppa.launchpad.net/ryanhjefferson/ubuntu hardy main
```

After this has been added to the sources.list file and everything has been updated one can just apt-get install desktopdrawers to install Desktop Drawers.

In the PPA as of this moment is release 0.3a which includes a default home folder switcher, showing of windows after desktop has been shown, and fixing of a bug where folders that have been switched to can not be selected.

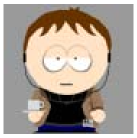
Please leave a message if you use Desktop Drawers.

Last edited by ratl3; September 11th, 2008 at 02:23 PM..

QUOTE

October 2nd, 2008 #6

izm81
5 Cups of Ubuntu



Join Date: Nov 2005
Beans: 23
Dapper Drake Testing/

Re: New applet: Desktop Drawers

Hi, Ryan! I've tried out "Desktop Drawers" for a day and it is interesting! Nice little project. 🙂 I've filed a few bugs on the project page. 😉

I think it's a great start! But in general, I think the most compelling use-case is not made clear enough, neither through the UI nor the documentation / description:

1. Install the Desktop Drawers applet on your panel.
2. Select Manage Desktops to open the ~/.Desktops directory in Nautilus.
3. In this folder, create folders for different "workspaces," ("Project 1", "Project 2", etc)
4. Then create links to related folders and files in each of these workspaces.
5. Projects/Workspaces can now be switched between using the applet.

Although it's not quite addressing the issues that I wanted to with the "Intelligent Desktop," (from my blog) I'm going to continue using Desktop Drawers for a while more, I think. 🙂 Thanks!

QUOTE

November 17th, 2008 #7

zhocchao
A Carafe of Ubuntu

Join Date: Nov 2008
Beans: 91

Re: New applet: Desktop Drawers

hi
it's interesting. What about shortcuts. As I am switching workspaces with crl + alt + left/right i could switch desktops with ctr + alt + up/down.
greetings
z

QUOTE

New Reply

Bookmarks

Digg del.icio.us StumbleUpon Google

Tags

applet, desktop, drawers, panel

« Previous Thread | Next Thread »

Posting Rules

You **may not** post new threads
You **may not** post replies
You **may not** post attachments
You **may not** edit your posts

BB code is **On**
Smilies are **On**
[IMG] code is **On**
HTML code is **Off**

Forum Rules

Forum Jump
Ubuntu Dev Link Forum Go

-- CleanV2

ubuntu forums - archive - top