

The GNOME3 Desktop and You

Guido Günther <agx@sigxcpu.org>

FrOSCon 2011

About myself

- Freelancing Free Software Developer
- Debian Developer since 2000
- Contributions to GNOME/GTK+/GObject based apps since about 2008: krb5-auth-dialog, PPM, ModemManager, gtk-vnc, virt-manager
- I'm less an author but more a user of *GNOME* APIs.

About this talk

- Covers things I came across when working on the above projects in
- Tries to give an introduction so it's simpler to dive deeper into it GNOME development

GNOME3

- GNOME is a desktop environment built on top of *GTK+/GLib/GObject*
- GNOME's user interface is shaped by *GNOME Shell*

Overview

- 1 Introduction
- 2 GObject Introspection
- 3 GSettings
- 4 Notifications
- 5 Shell Extensions
- 6 The End

GTK+, GLib, GObject

- GLib is a cross platform C library that provides:
 - Data types: Lists, Hash Tables, Trees, Caches, Strings, ...
 - Application Support: Threads, Loadable Modules, Memory Management, ...
 - Utilities: Timers, Checksums, Random Numbers, Parsers, Testing framework, ...
 - gobject: The GLib Object system
 - gio: filesystem monitoring, async I/O, networking, DBus, settings, ...
- GTK+3 is the widget toolkit
 - Based on GObject
 - Widgets, clipboard, key bindings, d'n'd, theming, ...
- Many other libraries are based on GObject: libsoup, gtk-vnc, telepathy, ...

GTK+, GLib, GObject

- GLib is a cross platform C library that provides:
 - Data types: Lists, Hash Tables, Trees, Caches, Strings, ...
 - Application Support: Threads, Loadable Modules, Memory Management, ...
 - Utilities: Timers, Checksums, Random Numbers, Parsers, Testing framework, ...
 - **gobject**: The GLib Object system
 - **gio**: filesystem monitoring, async I/O, networking, DBus, **settings**, ...
- GTK+3 is the widget toolkit
 - **Based on GObject**
 - Widgets, clipboard, key bindings, d'n'd, theming, ...
- **Many other libraries are based on GObject**: libsoup, gtk-vnc, telepathy, ...

GNOME Shell

- Workspace and window management
- Application life cycle
- Notification system
- Integrated IM
- Can be extended in JavaScript
- much more...

GNOME Shell

- Workspace and window management
- Application life cycle
- **Notification system**
- Integrated IM
- **Can be extended in JavaScript**
- much more...

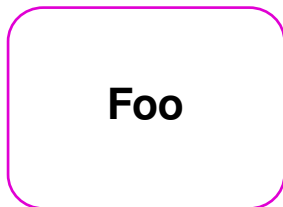
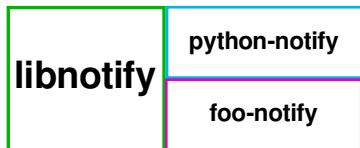
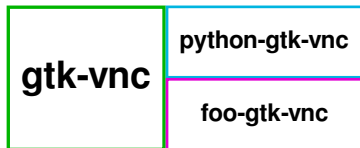
What's a GObject

- GLib has it's own dynamic type system:
 - Non classed: *numbers, pointers, ...*
 - Instantiable classed types: *objects*
 - Non-instantiable classed types: *interfaces*

What's a GObject

- GLib has it's own dynamic type system:
 - Non classed: *numbers, pointers, ...*
 - Instantiable classed types: *objects*
 - Non-instantiable classed types: *interfaces*
- GObject is the *base class* of GLib's type system
 - provides: *signals, memory management, per-object properties*
- Written in C, very binding friendly

What is it?



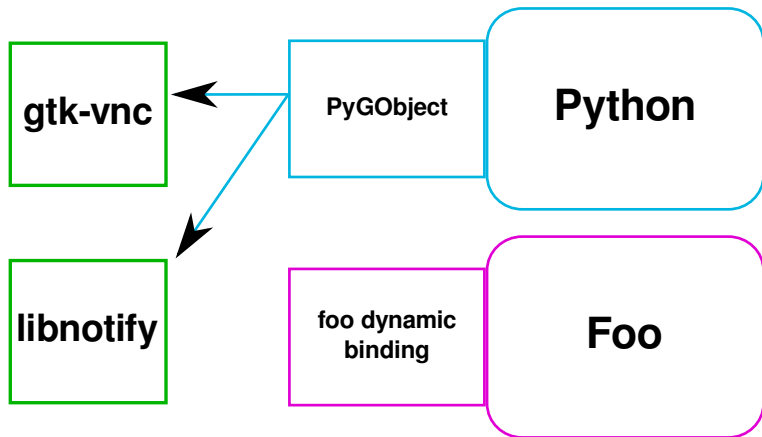
What is it?

- Layer to use GObject based libraries written in C from other languages like:
 - JavaScript (gjs, seed)
 - Python (PyGObject)
 - ruby
 - scheme
 - PHP
 - and more

GObject Introspection

- Only one dynamic binding per language needed to use all GObject introspection enabled libraries.
- No outdated bindings!

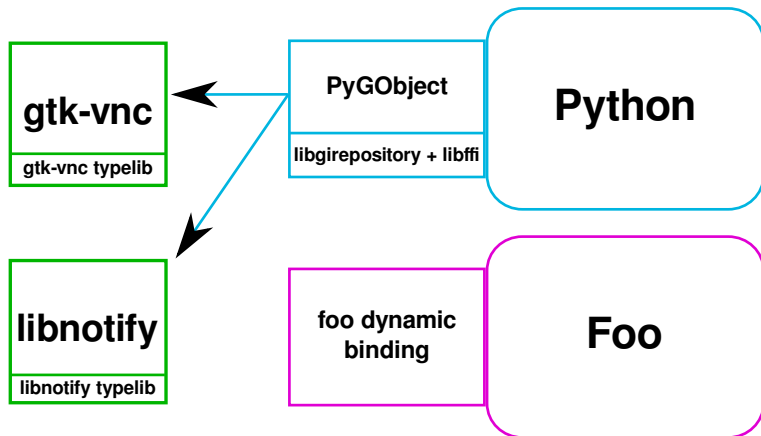
What is it?



How does it work?

- When writing the C library
 - Add annotations to the libraries source code (reference counting, allocation)
- C library build time
 - Scanner (g-ir-scanner) generates GIR (XML) from sources and built libraries
 - Debian: *-dev packages in */usr/share/gir-1.0/*.gir*
 - Compiler (g-ir-compiler) to compile GIR to typelib
 - Debian: gir1.2-* packages in */usr/lib/girepository-1.0/*.typelib*
- Runtime (e.g. gjs or Python+PyGObject)
 - libgirepository to read the introspection data
 - libffi to make library callable from other language

What is it?



How to use it from JavaScript?

JavaScript VNC viewer¹

gvncviewer.js - VNC in 10 lines

```
const Gtk = imports.gi.Gtk;
```

```
Gtk.init(0, null);
```

```
Gtk.main();
```

¹Example taken from [gtk-vnc/examples](https://github.com/any1/gtk-vnc/tree/master/examples)

How to use it from JavaScript?

JavaScript VNC viewer¹

gvncviewer.js - VNC in 10 lines

```
const Gtk = imports.gi.Gtk;

Gtk.init(0, null);
var win = new Gtk.Window({title: "GTK-VNC"});

Gtk.main();
```

¹Example taken from [gtk-vnc/examples](https://github.com/any1/gtk-vnc/tree/master/examples)

How to use it from JavaScript?

JavaScript VNC viewer¹

gvncviewer.js - VNC in 10 lines

```
const Gtk = imports.gi.Gtk;
const Vnc = imports.gi.GtkVnc;

Gtk.init(0, null);
var win = new Gtk.Window({title: "GTK-VNC"});

Gtk.main();
```

¹Example taken from [gtk-vnc/examples](#)

How to use it from JavaScript?

JavaScript VNC viewer¹

gvncviewer.js - VNC in 10 lines

```
const Gtk = imports.gi.Gtk;
const Vnc = imports.gi.GtkVnc;

Gtk.init(0, null);
var win = new Gtk.Window({title: "GTK-VNC"});
var disp = new Vnc.Display();

win.add(disp);

Gtk.main();
```

¹Example taken from [gtk-vnc/examples](#)

How to use it from JavaScript?

JavaScript VNC viewer¹

gvncviewer.js - VNC in 10 lines

```
const Gtk = imports.gi.Gtk;
const Vnc = imports.gi.GtkVnc;

Gtk.init(0, null);
var win = new Gtk.Window({title: "GTK-VNC"});
var disp = new Vnc.Display();

win.add(disp);

disp.open_host("localhost", "5901");

Gtk.main();
```

¹Example taken from [gtk-vnc/examples](#)

How to use it from JavaScript?

JavaScript VNC viewer¹

gvncviewer.js - VNC in 10 lines

```
const Gtk = imports.gi.Gtk;
const Vnc = imports.gi.GtkVnc;

Gtk.init(0, null);
var win = new Gtk.Window({title: "GTK-VNC"});
var disp = new Vnc.Display();

win.add(disp);

disp.open_host("localhost", "5901");
win.show_all();
Gtk.main();
```

¹Example taken from [gtk-vnc/examples](#)

How to use it from JavaScript?

JavaScript VNC viewer¹

gvncviewer.js - VNC in 10 lines

```
const Gtk = imports.gi.Gtk;
const Vnc = imports.gi.GtkVnc;

Gtk.init(0, null);
var win = new Gtk.Window({title: "GTK-VNC"});
var disp = new Vnc.Display();

win.add(disp);
win.connect('delete-event', Gtk.main_quit);
disp.open_host("localhost", "5901");
win.show_all();
Gtk.main();
```

¹Example taken from [gtk-vnc/examples](#)

How to use it from Python?

gvncviewer.py - VNC in 9 lines

```
from gi.repository import GtkVnc, Gtk

Gtk.init(None)
win = Gtk.Window(title="GTK-VNC with Python")
disp = GtkVnc.Display()

win.add(disp)
win.connect('delete-event', Gtk.main_quit)
disp.open_host("localhost", "5901")
win.show_all()
Gtk.main()
```


What do I need?

Programming language, dynamic bindings, typelib files

JavaScript example

```
apt-get install gjs \
gir1.2-gtk-3.0 \
gir1.2-gtk-vnc-2.0
```

Python example

```
apt-get install python-gobject \
gir1.2-gtk-3.0 \
gir1.2-gtk-vnc-2.0
```

Note: no python-gtk-vnc, python-gtk2!

GObjects in JS, Python and C

- Constructors

```
var win = new Gtk.Window({title: "foo"});  
win = Gtk.Window(title="foo")  
win = g_object_new (GTK_TYPE_WINDOW,  
                    "title", "foo", NULL);
```

- Signals

```
win.connect("delete-event", callable);  
win.connect("delete-event", callable);  
g_signal_connect (win, "delete-event",  
                  gcallback, NULL);
```

- Properties

```
win["title"]; win.title  
win.props.title  
g_object_get (win, "title", &title, NULL);
```

What is it?

- API to retrieve and store configuration settings
- Easy to bind to GObject properties to settings
- Change notification via signals
- Supports different backends
 - DConf: store settings key based on disk
 - Memory
- XML schema describes location and types of keys
- Vendor overrides possible
- Delay mode
- Complex types possible using GVariant
- obsoletes gconf, gconf-bridge

How to use it?

Shell example

```
gsettings get \
  org.gnome.system.proxy.http enabled
gsettings set \
  org.gnome.system.proxy.http enabled true
```

Python example

```
python proxysettings.py
```

GUI

```
dconf-editor
```

What do I need?

Example

```
apt-get install libglib2.0-bin \  
                dconf dconf-tools
```

What is it?

- GNOME Shell's message tray displays and manages notifications to the user
- Notifications:
 - Provide feedback to the user
 - Displayed for a short period of time at the bottom of the screen
 - By default *persistent*: saved until interacted with or application is opened
 - Less distracting since no need to interact instantly
 - Can be globally disabled
 - *Resident* and *transient* notifications possible

How to use it?

gvncviewer.notify.py

```
from gi.repository import Notify
```

How to use it?

gvncviewer.notify.py

```
from gi.repository import Notify

def notify(obj, v):
    n = Notify.Notification(
        summary="Connected to %(host)s" % v,
        body="Made VNC connection to "
            "server %(host)s at port "
            "%(port)s" % v)
    n.set_hint("resident",
              GLib.Variant('b', True))
    n.show()
```


How to use it?

gvncviewer.notify.py

```
from gi.repository import Notify

def notify(obj, v):
    n = Notify.Notification(
        summary="Connected to %(host)s" % v,
        body="Made VNC connection to "
            "server %(host)s at port "
            "%(port)s" % v)
    n.set_hint("resident",
              GLib.Variant('b', True))
    n.show()
    ...
disp.connect("vnc-connected", notify,
            vncserver)
```

What do I need

Python example

```
apt-get install gir1.2-notify-0.7
```

GNOME Shell

- GNOME3 desktop shell written in C and JavaScript
- Heavily uses Clutter
- Extendable via JavaScript and GObject Introspection
- Available Extensions:
<https://live.gnome.org/GnomeShell/Extensions>
- The shell has a built JavaScript inspector/debugger
(ALT-F2 → lg)

How to use them?

Create new extension

```
gnome-shell-extension-tool --create-extension
```

Example

Example [Launch_Iceowl@sigxcpu.org/](mailto:Launch_Iceowl@sigxcpu.org)

What do I need?

Example

```
apt-get install -t experimental gnome-shell
```

Your own extension

```
git clone git://git.gnome.org/gnome-shell.git
```

- gjs style guide
- An idea about GObject Introspection (see above)

Porting to GTK+3, GDBus, GSettings, PyGObject

- GNOME3 porting guide:
<http://live.gnome.org/Gnome3PortingGuide>
- GTK+2 → GTK+3:
<http://developer.gnome.org/gtk3/3.0/migrating.html>
- GConf → GSettings
<http://developer.gnome.org/gio/stable/ch28.html>
- dbus-glib → GDBus
<http://developer.gnome.org/gio/unstable/ch29.html>
- python-gtk → PyGObject:
<http://git.gnome.org/browse/pygobject/tree/pygi-convert.sh>

API documentation

- API Documentation for PyGObject and gjs:
 - Look at the the C API documentation, it's well documented
 - You can infer Python or JavaScript calls form there

Example

```
gtk_widget_show_all (GtkWidget* widget);  
widget.show_all ();  
widget.show_all ();
```

- If in doubt consult the GIR file

API documentation

- Generating language specific docs from the introspection information is in the works:
 - <http://www.j5live.com/2011/08/15/gobjects-in-berlin-the-search-for-more-documentation/>
 - <https://live.gnome.org/GObjectIntrospection/Doctools>

Development with vim

```
apt-get install vim-syntax-gtk \  
                devhelp libgtk-3-doc
```

.vimrc

```
autocmd Filetype c nmap <silent> <C-K>  
      :! devhelp -s "<cword>" &<CR><CR>
```


Thank you

- Thanks!
- Questions?

- Source code and examples are at:

```
git clone git://honk.sigxcpu.org/git/talks/2011-08-gnome3-froscon.git
```

CC BY-SA 3.0 — Creative Commons Attribution-ShareAlike 3.0

Changes in GTK+, GObject, GLib

- GObject Introspection
- GSettings: Settings DB now included
- GDBus: API to access DBus now included

- GtkBuilder: UI XML now included
- GTK+3 uses Cairo
- Multiple GDK backend support (Wayland, HTML5)
- Themes use CSS

Changes in GTK+, GObject, GLib

- GObject Introspection → obsoletes writing language bindings for each library and language
- GSettings: Settings DB now included → obsoletes GConf
- GDBus: API to access DBus now included → obsoletes dbus-glib

- GtkBuilder: UI XML now included → obsoletes libglade
- GTK+3 uses Cairo → obsoletes GDK drawing API
- Multiple GDK backend support (Wayland, HTML5)
- Themes use CSS

Changes in GTK+, GObject, GLib

- **GObject Introspection** → obsoletes writing language bindings for each library and language
- **GSettings**: Settings DB now included → obsoletes GConf
- **GDBus**: API to access DBus now included → obsoletes dbus-glib

- **GtkBuilder**: UI XML now included → obsoletes libglade
- **GTK+3** uses Cairo → obsoletes GDK drawing API
- Multiple GDK backend support (Wayland, HTML5)
- Themes use CSS

Annotations example

```
/**
 * soup_message_body_append_take:
 * @body: a #SoupMessageBody
 * @data: (array length=length) (transfer full): data to append
 * @length: length of @data
 *
 * Appends @length bytes from @data to @body.
 *
 * This function is exactly equivalent to soup_message_body_append()
 * with %SOUP_MEMORY_TAKE as second argument; it exists mainly for
 * convenience and simplifying language bindings.
 *
 * Since: 2.32
 * Rename to: soup_message_body_append
 **/
void
soup_message_body_append_take (SoupMessageBody *body,
                              guchar *data, gsize length)
{
    soup_message_body_append(body, SOUP_MEMORY_TAKE, data, length);
}
```