static void _f_do_barnacle_install_properties(GObjectClass *gobject_class)
{
    GParamSpec *pspec;

    /* Party code attribute */
    pspec = g_param_spec_uint64
            (F_DO_BARNACLE_CODE,
             "Barnacle code.",
             "Barnacle code",
             0,
             G_MAXUINT64,
             G_MAXUINT64 /* default value */,
             G_PARAM_READABLE | G_PARAM_WRITABLE |
             G_PARAM_PRIVATE);

    g_object_class_install_property (gobject_class,
                                      F_DO_BARNACLE_PROP_CODE,
                                      pspec);

    g_object_class_install_property (gobject_class,
                                      F_DO_BARNACLE_PROP_CODE,
                                      pspec);
}

Embedded Linux Conference Europe
Cambridge, October 2010

Iago Toral Quiroga
itoral@igalia.com
Index

- Media Integration
- Overview of Grilo
- Demo
- Grilo for application developers
- Grilo for backend developers
Media Integration
Media Integration: services

- YouTube
- Jamendo: Open your ears.
- SHOUTcast Radio
- Vimeo
- UPnP
- Last.fm: the social music revolution
- Podcast
Media Integration: hardware
Media Integration: Present

➔ What are the challenges?
   ➔ Homogeneous, integrated user experience.
   ➔ Easy to use.
   ➔ Service scalability.
   ➔ Decrease development and maintenance effort.
Media Integration: Present

→ What are the problems?
  → Lots of heterogeneous services.
  → Lots of APIs and protocols.
  → Lots of technologies.
Media Integration: Present

➔ What are we doing now?
  ➔ Individual makers developing in-house solutions.
  ➔ Not efficient.
  ➔ Slow development.
  ➔ Not scalable.
  ➔ Expensive maintenance.
Media Integration: Future

➔ What can we do about this?
  ➔ Platform level solution for accessing media content.
  ➔ Stop reinventing the wheel, start reusing code.
  ➔ A place where interested actors can collaborate.
Media Integration: Future

➔ What are the benefits?
  ➔ Reduce and share maintenance effort.
  ➔ Focus on added value.
  ➔ Faster development.
  ➔ Scalability.
Grilo: Overview
Grilo: Overview

- A framework for easing access to multimedia content.
- Application developers want to browse / search content from many services...
- ...but they don't want to know how they work internally (APIs, protocols, technologies, limitations, ...)
- Single API to access media content, hiding differences among media providers.
- Application developers write their solution once and it will work for any service supported in Grilo.
Grilo: Overview
Demo
Grilo for application developers
Grilo for plugin developers
Resources

→ Wiki:
  → http://live.gnome.org/Grilo

→ Git repositories:
  → git://git.gnome.org/grilo
  → git://git.gnome.org/grilo-plugins

→ IRC:
  → grilo @ GIMPNet

→ Mailing list:
  → http://mail.gnome.org/mailman/listinfo/grilo-list

→ Bugzilla:
  → http://bugzilla.gnome.org

• Category: Other, Product: grilo
Image Credits

- **Easy button**: Jason Gulledge, www.flickr.com
  - http://www.flickr.com/photos/ramdac/373881476

- **Dead end**: AJC1, www.flickr.com
  - http://www.flickr.com/photos/ajc1/3019611194

- **Collaboration**: thinkpublic, www.flickr.com
  - http://www.flickr.com/photos/thinkpublic/3042777307/sizes/m/in/photostream

- **Headache**: www.freeclipartnow.com

- **Thumbs up**: Stefano Valle, www.freedigitalphotos.net
  - http://www.freedigitalphotos.net/images/Gestures_g185-Two_Hands_Thumbs_Up_p20593.html