

# Recent Files And Bookmarks

Or: How I Stopped Worrying About A  
File System

Emmanuele Bassi <[ebassi@openedhand.com](mailto:ebassi@openedhand.com)>



OpenedHand Ltd., <http://www.o-hand.com>



Ha!

Yes, it's a spinning  
cube



# Who am I?

- employed at OpenedHand
- gnome-utils co-maintainer (yes, you can blame me for gnome-dictionary)
- gtk2-perl bindings contributor
- GLib/GTK+ contributor



# Why this tutorial

- Understand how bookmarks/recent files work
- Support bookmarks in applications
- Support recent files in applications



# Problem Space/o

Define bookmarks and recent files in terms of requests.

- **Bookmark := link for a resource pointed by a URI**
- **Recent file := document used within an (arbitrary) amount of time**



# Problem Space/1

## Bookmarks

- user defined, long lived
- per-application or global
- possibly shared among applications



# Problem Space/2

## Recent Files

- application defined, short lived
- per-application or global
- globally shared but possibly hidden



# Problem Space/3

- Bookmarks and recently used files share the same solution space: both are placeholders for resources (pointed by a URI) with additional meta-data (for caching purposes).
- Recently used files are a special case of bookmarks automatically created by applications and usually are completely transparent to the user





# Solution Space/1

## Bookmarks in GTK+: `${HOME}/.gtk-bookmarks`

- Format hidden inside the FileChooser
- Only URI + name
- No API to access it from outside GtkFileChooser
- More users with custom-made parsers mean fewer chances to expand or change the format later on



# Solution Space/2

## Recent Files in GNOME: EggRecent

- Not maintained for 3+ years
- No clear position on the stack
- Spec missing meta-data
- Spec implemented only by GNOME



# Solution Space/2

## Recent Files in GNOME: EggRecent

- Inefficient, hacky, leaky
- Limited widgets (only menu, which was a hack)
- Model/View not really separated (views changed the model)



# Solution Space/3

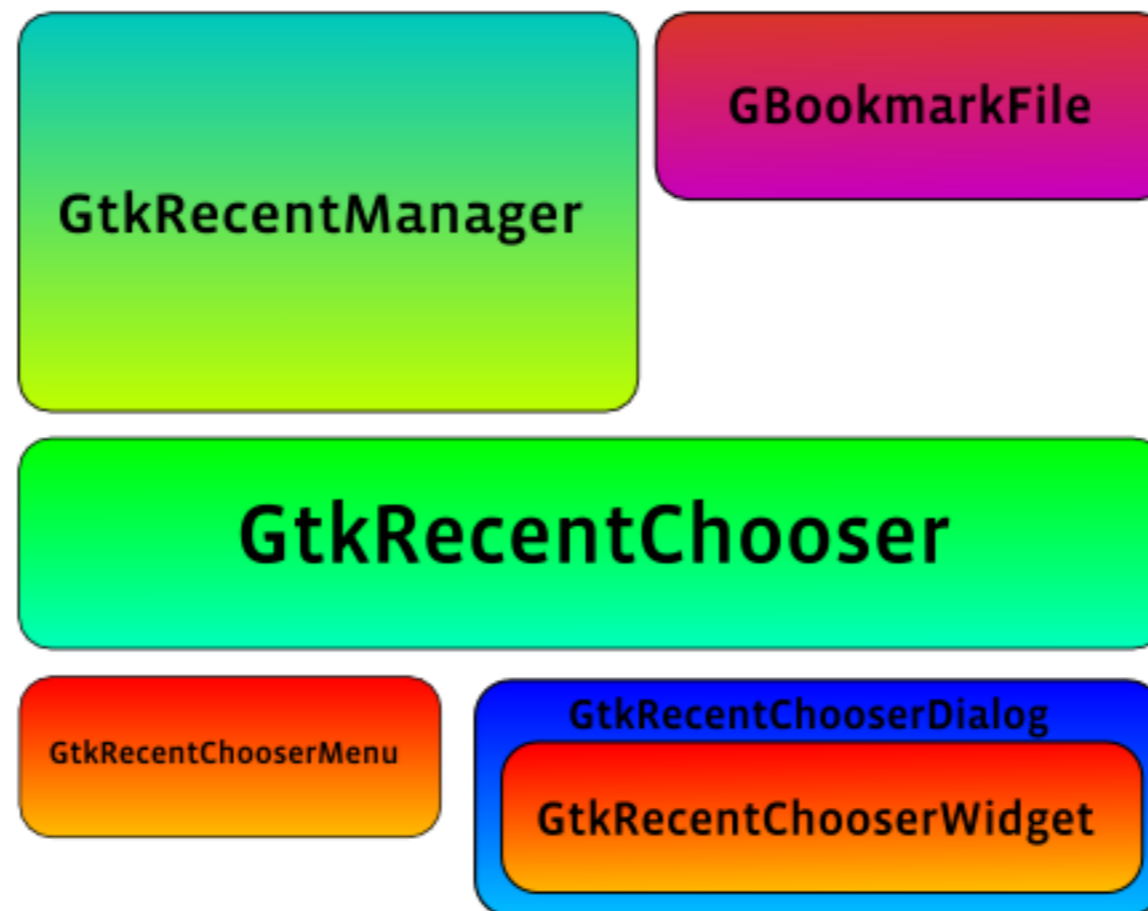
## Fixing Bookmarks and Recent Files

- Part of Project Ridley
- Re-design of the specification for the storage protocol and the relative meta-data, using XBEL as back-end
- Create reference implementation, using GLib only
- Lower the bookmarks and recent files support on the stack, and expose API for every application to use them
- Implement the widgetry for recent files inside GTK



# Solution Space/4

## Recent Files and Bookmarks





# An now...

- ... a totally pointless, unrelated and pretty embarrassing photo of mallum



# Storage Spec and API/1

## Storage specification: Desktop Bookmark Specification

- Based on XML Bookmark Exchange Language (XBEL)
- XBEL is used by other projects (KDE, Galeon, Epiphany, ...)
- Easily parsable (GMarkup, expat, libxml)
- Easily extensible (meta-data encapsulated)



# Storage Spec and API/2

Reference implementation: **GBookmarkFile**

- Inside GLib (>= 2.11.0)
- API similar to GKeyFile
- Fast (but improvable)





# GBookmarkFile

Using GBookmarkFile is possible to store bookmarks to resources, as long as the resource has valid URI and MIME type

- Bookmarks are accessed by URI
- Each bookmark is unique
- Each bookmark must have a MIME type
- Each bookmark must have an application registered



# GBookmarkFile/Add

- Adding a bookmark

```
GBookmarkFile *bookmarks = g_bookmark_file_new ();
const gchar *uri = "file:///tmp/recent-files.xml";

g_bookmark_file_set_title (bookmarks,
                           uri,
                           "Recent Files And Bookmarks");
g_bookmark_file_set_description (bookmarks,
                                 uri,
                                 "GUADEC 2006 Tutorial");
g_bookmark_file_set_mime_type (bookmarks,
                               uri,
                               "text/xml");
```



# GBookmarkFile/Set

- **Setting the bookmark's meta-data**

```
g_bookmark_file_add_application (bookmarks,  
                                uri,  
                                "Gedit", "gedit %u");  
g_bookmark_file_add_group (bookmarks,  
                            uri,  
                            "Presentation");  
g_bookmark_file_add_group (bookmarks,  
                            uri,  
                            "TextEditor");  
g_bookmark_file_set_private_hint (bookmarks, uri, TRUE);
```



# Desktop Places

- Applications should install bookmarks to standard places
- If "Foo" wants to export a bookmark "My Foo" for similar applications to use, it should save a bookmark file for that location inside `$XDG_DATA_DIRS/desktop-bookmarks`
- "Bar" can now use "My Foo" as a bookmark too
- `g_bookmark_file_load_from_data_dirs()`



# GtkRecent/1

- **Deprecates EggRecent (finally!)**
- **Better Model/View/Controller design**
  - **GtkRecentManager: model**
  - **GtkRecentChooser: viewer/controller**



# GtkRecent/2

- **More widgets**
  - **GtkRecentChooserDialog**
  - **GtkRecentChooserWidget**
  - **GtkRecentChooserMenu**
  - **(GtkRecentAction (2.12))**
- **(Integration with GtkFileChooser (2.12))**



# GtkRecentManager/1

- Database for recently used resources
- Using GBookmarkFile as storage (low-level access)
- xdgmime for getting MIME type
- Simple API





# GtkRecentManager/2

- Adding a resource

```
GtkRecentManager *manager;
```

```
manager = gtk_recent_manager_get_default ();  
gtk_recent_manager_add_item (manager, uri);
```

- But you won't often use it





# GtkRecentManager/3

- Usually, you know meta-data about your own application and you may want to set inside the list
- GtkRecentData and `gtk_recent_manager_add_full()`
- Title, description, MIME type, application name and command line, "private" bit, groups



# GtkRecentManager/4

- API for removing, moving, clear and look up
- You can fetch a single item or a list
- Each item is wrapped inside a `GtkRecentInfo` boxed type
- Each `GtkRecentInfo` is created on the fly: it's up to you to destroy it (use `gtk_recent_info_unref()`)

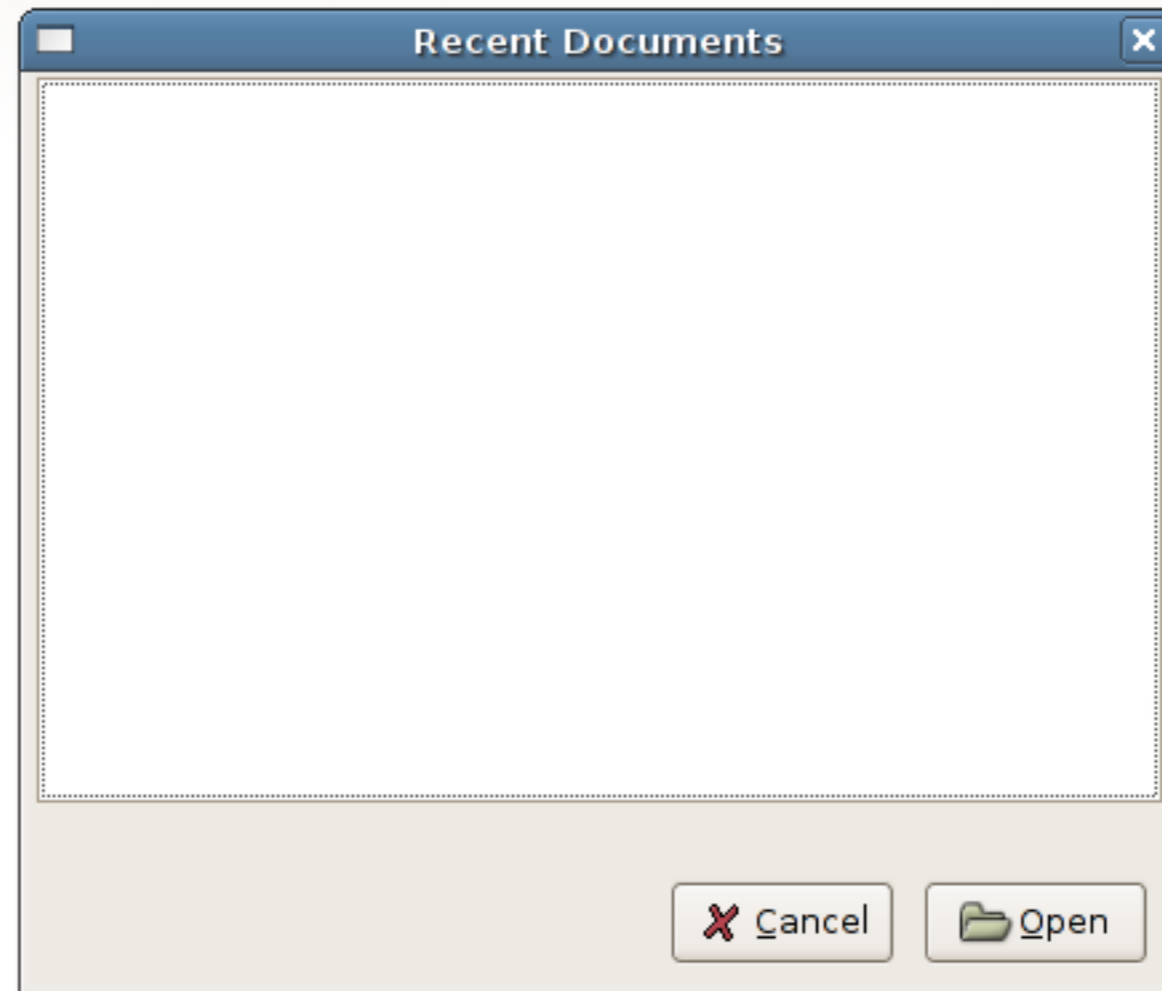


# GtkRecent Widget Zoo

- Each widget is an implementation of `GtkRecentChooser` `GInterface`
- The interface gives us uniform API, signals and properties
- If you need to create your own widget, use `GtkRecentChooser`



# GtkRecentChooserDialog



# GtkRecentChooserDialog

- Simple dialog, similar to `GtkFileChooserDialog`
- Drag and drop (`text/uri-list`)
- Multiple filters

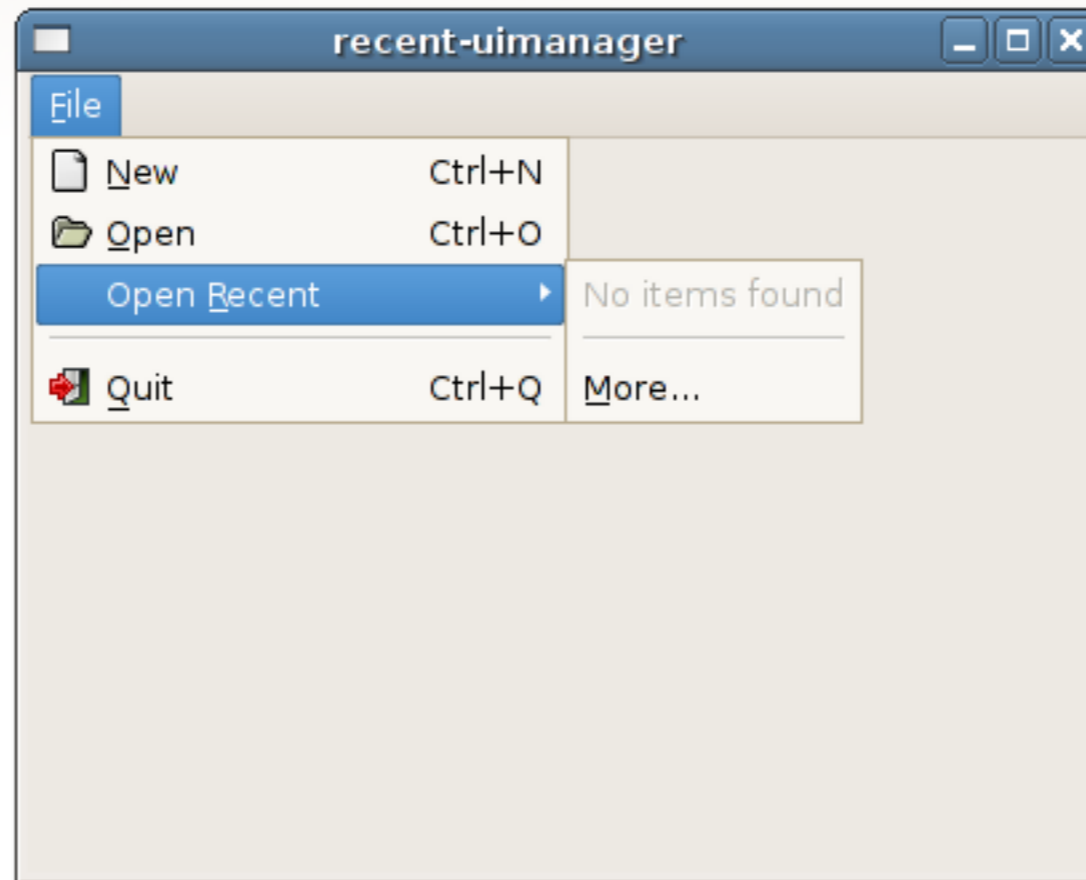


# GtkRecentChooserWidget

- **Embeddable widget**
- **Used by GtkRecentChooserDialog**
- **Good for side panes**



# GtkRecentChooserMenu



# GtkRecentChooserMenu

- **GtkMenu sub-class**
- **Appending items supported**
- **Use as a sub-menu for a GtkMenuItem**
- **Use as a sub-menu for a GtkMenuToolButton**





# GtkRecentChooserMenu

- No inline menu, sorry
- Adding inlined recent files is possible with menus built in code
- But you really don't want to do that



# GtkUIManager

- **GtkRecentAction didn't make it before GTK+ 2.10 API freeze**
- **Sub-class of GtkAction**
- **See example**



# Sorting and Filtering/1

- EggRecent supported poor sorting and filtering
- Sorting and filtering affected the list itself



# Sorting and Filtering/2

- **GtkRecent moves sorting and filtering to the GtkRecentChooser interface implementations**
- **Each view can sort and/or filter without interfering with each other**
- **This allows multiple views sharing a single GtkRecentManager**



# Sorting and Filtering/3

- GtkRecent has four types of sorting: none, most recently used, least recently used, custom
- MRU and LRU are built-in



# Sorting and Filtering/3

- Custom sorting using a comparison function:

```
static gint
my_sorting_function (GtkRecentInfo *a,
                    GtkRecentInfo *b,
                    gpointer          user_data)
{
    /* < 0 if a > b
       * = 0 if a == b
       * > 0 if b > a
       */
    return retval;
}
```



# Sorting and Filtering/4

- Filtering is done by `GtkRecentFilter`
- Similar to `GtkFileFilter`
- Adds support for `GtkRecentInfo` meta-data





# Example

- Example of program using:
- `GtkRecentFilter`
- `GtkRecentChooserMenu`
- `GtkRecentChooserDialog`
- `GtkUIManager`



# In Summary/1

- **GBookmarkFile should be used for providing a uniform storage protocol for locations inside GNOME applications (Desktop Places)**
- **Also, it provides a storage for user-defined locations**



# In Summary/2

- **GtkRecent is the code for handling recently used files**
- **Applications should register every successfully opened file and every successfully saved file using GtkRecentManager**
- **Applications should use the available widgetry**
- **If needed, implement the GtkRecentChooser GInterface**



# Thank You For Listening!

## Any questions?

