Clear the barrier between window-manager and desktop

GnomeShell as window-manager plugin
- Gives the desktop more control of window management
- Avoids huge API between WM ⇔ Shell

Start from a stable base
- Mutter is based on the successful and rock-solid metacity window manager
- Mutter draws windows using Clutter
Why JavaScript?

C for low-level - JavaScript for high-level
- The GNOME platform is in C
- Some things perform better when written in a low-level language
- Writing UI is much faster in a scripting language

Choosing the scripting language
- JavaScript does not have its own platform libraries
- Lots of performance work done due to browser wars
- Known to many web developers already
**GOBJECTINTROSPECTION**

**Goal**
- Two level applications - C + *your favorite runtime*
- Use common metadata for all bindings
- Semi-automatic binding generation

** Bindings that use GOBJECTINTROSPECTION**
- JavaScript bindings
- PyGI Python bindings
- Vala
/∗∗
   * gtk_list_store_set_column_types:
   * @store: a #GtkListStore
   * @n_columns: Length of @types
   * @types: (array length=n_columns): List of types
   */

void
gtk_list_store_set_column_types (GtkListStore *list_store,
gint n_columns,
GType *types);

Metadata

- Include data that cannot be specified with C features
- in/out parameters, transfer mode, array length, type of list items, etc.
- Metadata is included in gtk-doc comments
GTK+ 3.0

The GTK+ Project

API/ABI break

- 3.0 will be API/ABI incompatible with the 2.0 series
- Removes all public struct members to ensure internal changes won’t affect ABI in the future
- All widgets that have been deprecated in the 2.0 cycle are removed
  ⇒ Make it easier to maintain GTK+

Features

- Extended Layout
- Multitouch/XInput2
Extended layout

- Widgets can give a hint on their preferred size
- Space is divided between widgets taking the preferred size into account
Multitouch
Porting applications to GTK+ 3.0

Golden rule

Anything that compiles with GTK+ 2.20 and

- \texttt{-DGSEAL\_ENABLED}
- \texttt{-DGDK\_DISABLED\_DEPRECATED}
- \texttt{-DGTK\_DISABLED\_DEPRECATED}

will work out of the box with GTK+ 3.0
dconf vs. gconf

Replacing gconf
- gconf development has stalled for years
- gconf is slow at system startup

DConf
- Optimized for reads (Login: 1000s reads, 2 writes)
- Writes are non-blocking
- dconf is a backend for GSettings
GSettings

```c
gboolean g_settings_set_string (GSettings *settings, const gchar *schema_id,
                               const gchar *key, const gchar *value);

char * g_settings_get_string (GSettings *settings, const gchar *schema_id);
```

New configuration interface for GLib

- Simple interface for key/value storage
- Store boolean, int, double, strings, lists, etc.

Backends

- dconf (GNOME)
- registry (Windows)
- keyfile (ini-style file)
- gconf (testing)
GNOME 3.0 - Summary

- New desktop experience: GnomeShell
- API/ABI Break: GTK+ 3.0
- New configuration system: dconf/GSettings
- New tutorial based help: Mallard
- ...and more...

GNOME 3.0 != GNOME 3