

BUILDER

Vaporware Galore

Christian Hergert
<christian@hergert.me>

#GUADEC2014

!WARNING!

Today this is all VAPORWARE.
(but wait till the end)

GNOME UNIVERSITY

I sort of failed at this.

Turns out we were lacking the tools to do it properly.

So now it's time to build those tools.

WHAT BUILDER ISN'T

Not a generic IDE.

If you want one of those, use MonoDevelop, Anjuta, Eclipse, IDEA, or a multitude of other options.

Builder will probably not support plugins (at least initially).

WHAT BUILDER IS

... mostly mockups and prototypes.

Builder is an attempt to create a development environment for developing GNOME applications.

HOW IS THIS THING BUILT?

Multi-Process
Project Management
Source Editor
Global Search
UI Designer
Resource Manager
D-BUS Explorer
Simulator
Documentation
Debugger
Profiler
Source Control

MULTI-PROCESS

Nothing worse than an IDE that crashes.
Except maybe a browser that crashes.

Components are broken into Services.
Services are contained in sub-processes.
Private D-BUS for IPC.

VIM/Emacs integration possible.

PROJECT MANAGEMENT

Basic autotools management.

By adhering to a strict autotools convention, we can avoid complex parsers for autoconf and automake files.

Focusing on GLib based projects.

(gresource, glade, gsettings, etc)

Will be written as library for command line tools.

SOURCE EDITOR

GtkSourceView
Code Highlighting
Auto-Completion
Cross-Reference
Search Highlighting
Change Tracking
Snippets
Auto-Formatting
Diagnostics and Fix-its
Refactoring
Distraction Free Mode

CODE HIGHLIGHTING

```
13 static gboolean
14 static GtkWidget      *
15 static GtkSourceView *
16 static GtkWidget      *
17 static GbClang         *
18 static guint
19 static GtkWindow       *
20
```

AUTO-COMPLETION

```
899
900     gtk_search_bar_
901     [M] gtk_search_bar_get_show_close_button 1
902     [M] gtk_search_bar_get_search_mode 2
903     [M] gtk_search_bar_connect_entry 3
904     [M] gtk_search_bar_new 4
905     [M] gtk_search_bar_set_show_close_button 5
906     [M] gtk_search_bar_set_search_mode
907     [M] gtk_search_bar_handle_event
908
909     pane1;
910
911
```

...ler =
... "key-press-eve
... entry_key_press_
...
... "changed",
... entry_changed_cb

AUTO-COMPLETION

Couple of options here.

Use clang, far more robust of results.

Use fuzzy search with data parsed from ctags.

Simple binary search of symbol index.

Clang will give us rich parameter completion.

AUTO-COMPLETION

```
f  typename cast_retty<X, Y>::ret_type llvm::dyn_cast<class X>(Y const &Val)  
f  typename cast_retty<X, Y *>::ret_type llvm::dyn_cast_or_null<class X>(Y *Val)  
    dynamic_cast<type>(expression)  
S  DynamicUpdate
```

SEARCH HIGHLIGHTING

```
279         NULL);  
280  
281     gutter = gtk_source_view_get_gutter(sourceview, GTK_TEXT_WINDOW_LEFT);  
282     renderer = gb_source_gutter_renderer_clang_new(clang, "main.c");  
283     gtk_source_gutter_renderer_set_padding(renderer, 2, 1);  
284     gtk_source_gutter_renderer_set_size(renderer, 14);  
285     gtk_source_gutter_insert(gutter, renderer, -30);  
286     tracker = gb_change_tracker_new(GTK_TEXT_BUFFER(buffer));  
287     renderer = gb_source_gutter_renderer_changes_new(tracker);  
288     gtk_source_gutter_renderer_set_size(renderer, 1);  
289     gtk_source_gutter_renderer_set_padding(renderer, 2, 0);  
290     gtk_source_gutter_insert(gutter, renderer, -10);  
291     g_signal_connect(sourceview,  
292                     "query-tooltip",  
293                     G_CALLBACK(query_tooltip),  
294                     NULL);  
295     g_signal_connect(buffer,
```

CHANGE TRACKING

```
27 #include "gb-  
28  
29 G_DEFINE_TYPE  
30  
31  
32  
33 struct _GbApp  
34 {  
35     /*  
36      * NOTE: S  
37      */  
38     GbWorkspac  
39 };  
40  
41 enum
```

SNIPPETS

```
12  
13 | foo_do_something ("format", param, param, param);  
14
```


AUTO-FORMATTING

DIAGNOSTICS AND FIX-ITS

Compile source editor contents in background while typing.

Libclang gives us diagnostics.

Some are Errors, some are Fix-its.

Fix-its are "one-click" helpers to fix a well known problem.

Missing a trailing ; is an example.

REFACTORING

Just use libclang for now (hurts me to say)

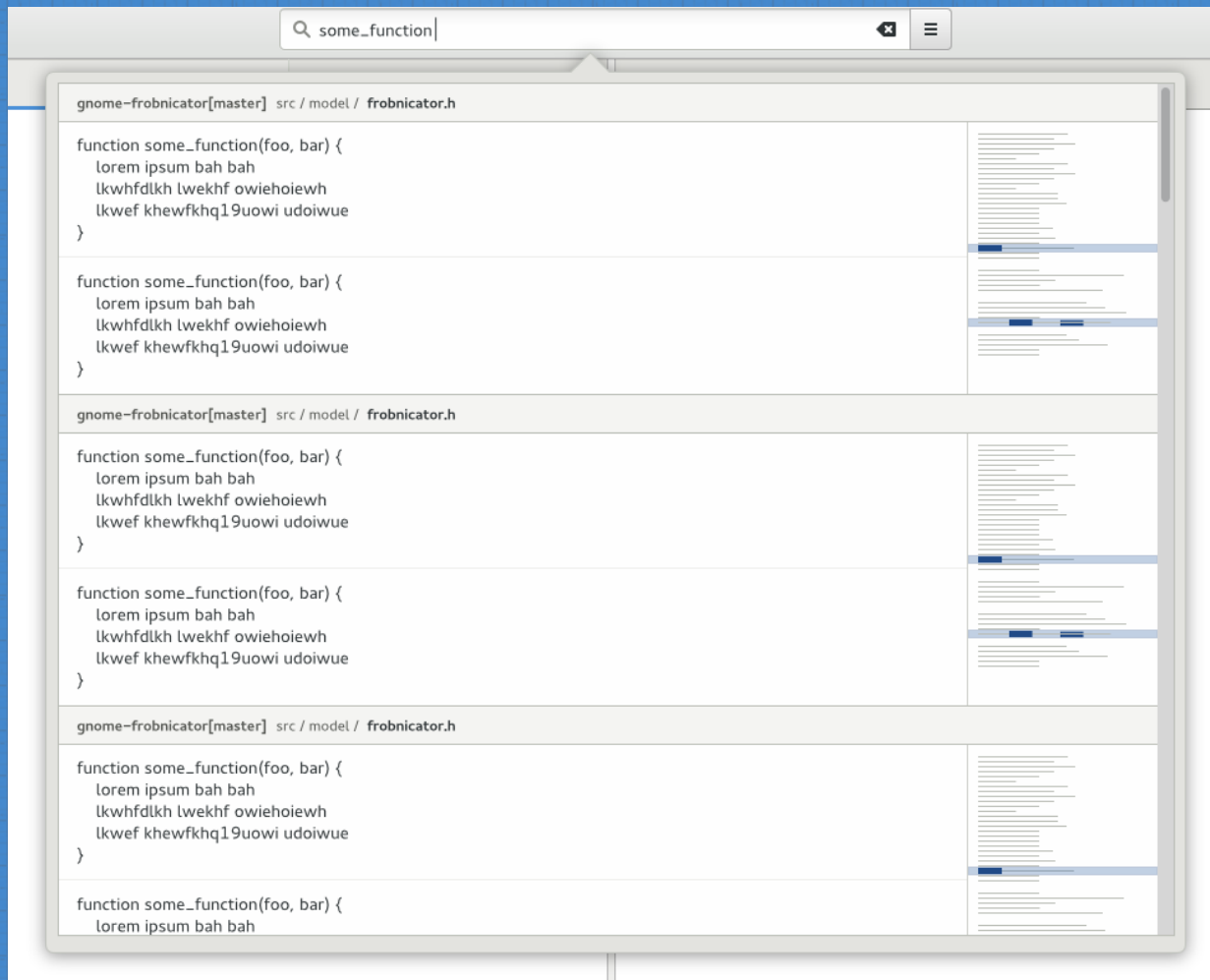
Rename locals

Extract methods

DISTRACTION FREE MODE

```
1 /*
2  * Copyright 2013 MongoDB Inc.
3  *
4  * Licensed under the Apache License, Version 2.0 (the "License");
5  * you may not use this file except in compliance with the License.
6  * You may obtain a copy of the License at
7  *
8  * http://www.apache.org/licenses/LICENSE-2.0
9  *
10 * Unless required by applicable law or agreed to in writing, software
11 * distributed under the License is distributed on an "AS IS" BASIS,
12 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 * See the License for the specific language governing permissions and
14 * limitations under the License.
15 */
16
17
18 #include <inttypes.h>
19 #include <stdarg.h>
20
21 #include "b64_ntop.h"
22 #include "bson.h"
23 #include "bson-private.h"
24
25
26 #ifndef BSON_MAX_RECURSION
27 #define BSON_MAX_RECURSION 100
28 #endif
29
30
31 typedef struct
32 {
33     bson_validate_flags_t flags;
34     ssize_t err_offset;
35 } bson_validate_state_t;
36
37
38 typedef struct
39 {
40     bson_uint32_t count;
41     bson_bool_t keys;
42     bson_uint32_t depth;
43     bson_string_t *str;
44 } bson_json_state_t;
45
46
47 static bson_bool_t
48 bson_as_json_visit_array (const bson_iter_t *iter,
49                           const char *key,
50                           const bson_t *v_array,
```


GLOBAL SEARCH



UI DESIGNER

Manage templates and backing code.
Template editor by embedding Glade.
Integrate Gtk Inspector.

RESOURCE MANAGER

UI to manage GResources in your project.

- Icons (theming integration)
- Files
- Translations?

D-BUS EXPLORER

Ever try to consume a D-BUS API from your project?

- Browse to D-BUS target similar to d-feet.
- Click to generate proxy as part of build system.
- Consume from application.

No need to call `gdbus-codegen` yourself.

SIMULATOR

Look similar to Boxes.

Use OSTree to fetch GNOME releases.

Always test in a clean OSTree environment.

(Hope to solve the "Works For Me" mentality)

Also, run code on remote laptop/phone/tablet in similar way.

DOCUMENTATION

Start by using and improving Devhelp.
Access documentation during auto-completion.

Perhaps we should also work on making it easy to write
documentation.

Markdown Editor.

DEBUGGER

Reuse as much of Nemiver as possible.

Visualizers (show GdkPixbuf upon cursor hover).

Remote GDB into Simulator or Device.

Maybe Mozilla RR for recording session.

Always run debugger, so you can explore upon failure without having to re-run your experiment.

PROFILER

Linux perf tool is actually quite good these days.
operf (oprofile), sysprof, and others are useful.
Basic tracing support with GLib/Gtk and systemtap.
strace.
uber-graph.

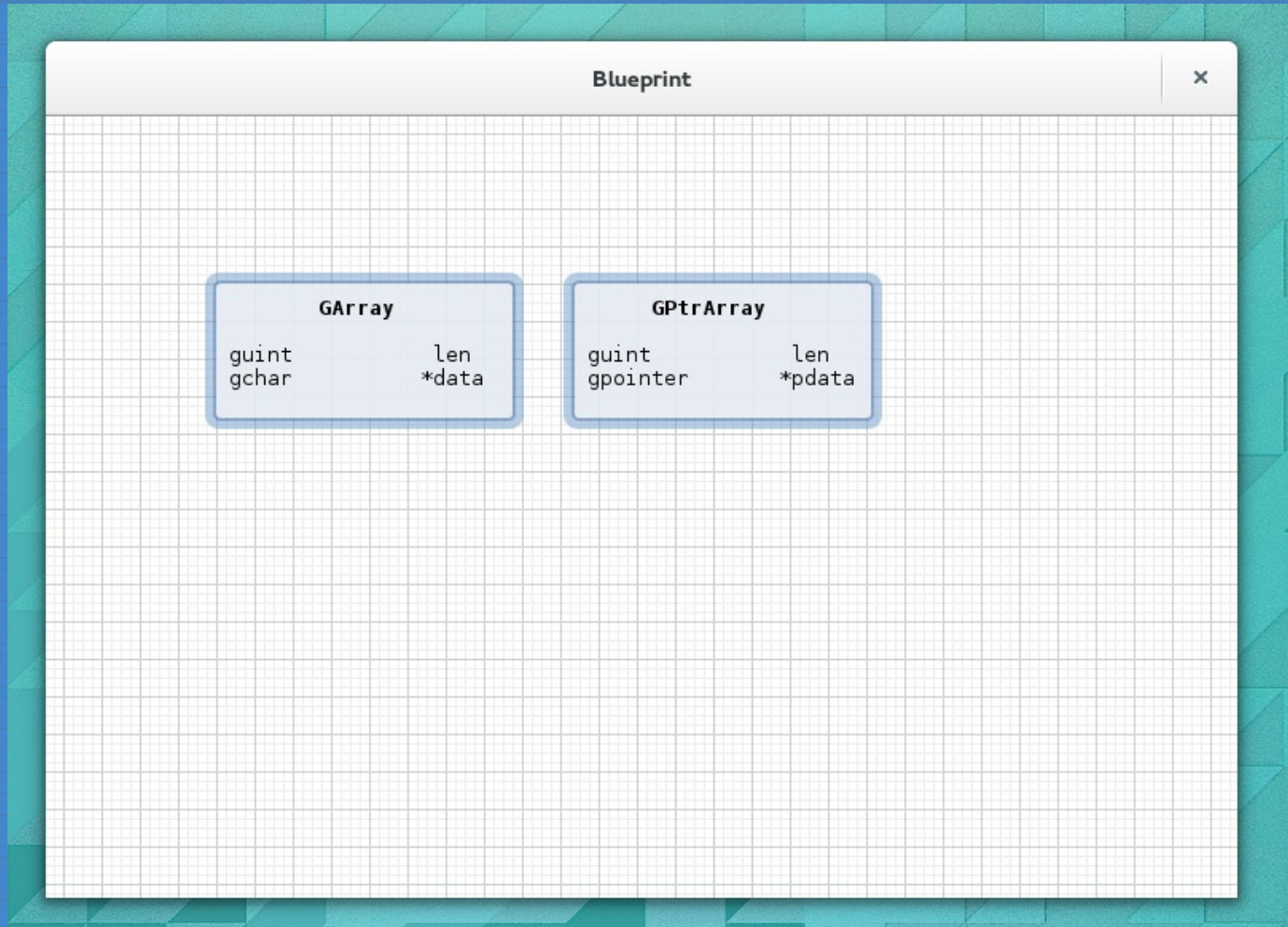
REMEMBER PERFKIT?

Yeah, I wrote this back around 2010.

Abandoned it because Linux perf came out and was clearly a better way to go.

Possibly time to revisit that work and integrate into Builder.

CLASS DIAGRAMS



WHERE ARE WE NOW?

Lots of small prototypes to prove designs.
Basic prototype of putting things together.
Time to make it real.

DEMO TIME

SO WHEN CAN WE EXPECT IT?

SOON!

FUNEMPLOYMENT

Resigned from MongoDB last week
to work on Builder full time for one year.

However, doing so while living off my savings.
Indiegogo or similar around November.

Starting September!

PATCHES ACCEPTED



QUESTIONS?