

Class 09: GIMP Internals (1): GIMP Plug-Ins

Held: Thursday, 4 April 2013

Summary: We begin to explore plug-ins for the GNU Image Manipulation Program.

Related Pages:

- EBoard.

Notes:

- Homework: Write an “interesting” plug-in for the GIMP. (Something that makes the "No" sign (circle with slash) counts as interesting.)
- We will start class by going over homework.
- Most of today’s notes are in the sample code. Let me know when you have questions.

Overview:

- An overview.
- Examples.

Some References

- <http://www.gimp.org/docs/plugin/sect-essentials.html>
- <http://developer.gimp.org/plugin-ins.html>
- Headers in `/usr/include/gimp-2.0`

An Overview of GIMP Plugins

- In the early days of the GIMP, the designers realized that the GIMP would be much more powerful if it included ways for other programmers to easily add features to the GIMP without having to learn too many GIMP internals or to recompile.
- Their solution (and it’s a typical solution)
 - Permit the creation of “plug-ins”, small pieces of code that can be added to the GIMP.
 - Provide appropriate libraries that permit the programmer to do interesting things (not everything, but lots of interesting things).
- Some additional aspects
 - Try to provide mechanisms that allow plugins to learn about each other.
 - Try to provide mechanisms that allow plugins to learn what features are generally available.
 - Include a multi-typed data value to support generic calls. (Just like the GValues we used in GDBus.) In GIMP, these are called `GimpParams`.
- So, what do you need to do to implement a GIMP Plug-in (once you’ve decided what it’s going to

do).

- Include the GIMP libraries `<libgimp/gimp.h>`
 - Design information on your plugin
 - `char* name`
 - `char* blurb`
 - `char* help`
 - `char* author`
 - `char* copyright`
 - `char* date`
 - `char* menu_path` - begins with `<Toolbox>`, `<Image>`, `<Load>` or `<Save>`
 - `char* image_types` - some plugins only deal with certain image types
 - `GimpPDBProcType` type - usually `GIMP_PLUGIN`, sometimes `GIMP_EXTENSION`
 - `int nparams` - the number of parameters
 - `int nreturn_vals` - the number of returned values
 - `GimpParamDef*` `params` - descriptions of the parameters
 - `GimpParamDef*` `return_vals` - descriptions of the return values
 - Provide four functions:
 - Initialization (optional) - called when GIMP starts up
 - Quit (optional) - called when the GIMP shuts down
 - Query (required) - called the first time the plug-in is used (or when it's updated, I think)
 - Run (required) - does the real work.
 - Add lots of boilerplate
 - Compile with `gimptool-2.0 --install plugin.c`
 - There are other ways to compile, but we'll deal with those another day.
 - Run it through the GIMP
- Note: You can see a list of many GIMP procedures with the PDB browser.
 - Note: If MediaScheme isn't installed, you can still get a simple Scheme console through Script-Fu

Example 1: The Skeleton of a Global Plugin

Example 2: The Skeleton of an Image Plugin

Example 3: A Simple Image Plugin

Copyright © 2012 Samuel A. Rebelsky. This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.