Planet SoC

Week 5 Updates: Incoming Message Formatting

Sun, 07/01/2007 - 06:24 — jeff

A feature that has been on my to do list for a long time is adding text formatting support to msimprpl. MySpaceIM uses a markup language seemingly inspired by HTML, but much simpler (call it msim markup). All tags and attributes are single characters, and only a handful of tags are supported. The <f> tag not only controls the font face and size, as with HTML's <font>, but also enables bold, italic, and underline text formatting using a bit field. The HTML <b>, <i> and <u> tags are occasionally incorrectly thought of as setting a bit field in a browser (this may be how it is implemented internally, but it is not what conceptually happens), but with MySpaceIM markup, it is a reality. This probably simplified MySpaceIM's programming since you wouldn't have to handle nested tags.

Additionally, the msim markup language also supports separate tags for setting text color, background color, specifying paragraphs, and inserting emoticons.

Text formatting support in msimprpl has been on hold for a while because doing so requires knowing how to do two things:

- Parse the proprietary msim markup
- Translate it into something libpurple can understand

Fortunately, neither were too difficult. I was able to parse the proprietary markup language using libpurple's XML parsing facilities. The xmlnode_* functions, defined in the xmlnode.c module, are fairly easy to use once you get the hang of it. One trick is that none of the xmlnode functions allow you to loop over all the children of a parent. There is `xmlnode_get_child(const xmlnode *parent, const name *name)`, but it requires a child name. `xmlnode_get_next_twin(xmlnode *node)` looks promising because it does not take a child's name, but all it does is return a child with the same name as the node you pass it (useful if you have a parent with many children of all the same name, for example in `~/.purple/blist.xml`). With some help from #pidgin, I figured it out: simply access the members of the _xmlnode structure.

Next I had to find out how libpurple handles text formatting. This, too, is simple: it uses a stripped-down subset of HTML. Prior to 2001, Gaim used GtkHTML, but now conversations are rendered using the GtkIMHtml engine. There has been some talk on devel about replacing GtkIMHtml with Gtk+'s Webcore, but for now GtkIMHtml it is.

Armed with my new knowledge of libpurple's xmlnode and GtkIMHtml, I wrote msim_markup_xmlnode_to_html(). This function generates libpurple HTML for the root node, then recursively for each of the children. Incoming formatted instant messages are now partially supported--at least, the <f> tag (but there is no support for sending formatted messages, yet).

Recursion proved very valuable in solving this problem, since SGML/XML-based markup is naturally recursive.

As aside, I'd like to mention that in working on msimprpl I've gained more experience on how to make better compromises in terms of performance and complexity. Recursion greatly simplified msim markup parsing, although using

it means that each child element causes a new stack frame to be pushed and popped. Since msim markup is not deeply nested, this is a worthwhile tradeoff versus writing a more complicated iterative solution. Similarly, when parsing incoming protocol messages I first used a GHashTable. A hash table's main benefit is speed. Access is, in best case, is O(1)--each access takes the same amount of time. My custom MsimMessage implementation, which I discussed at length in earlier blog posts, uses a GList (since outgoing messages need to maintain order, and MsimMessage handles both incoming and outgoing messages). This means that accessing a field is O(n) -- proportional to the size of the list. However, this performance hit is not noticeable because of the small size of protocol messages. If it does become a problem, I can benchmark msimprpl to find the bottlenecks. Avoid premature optimization.

Just found out this week that Trillian Astra will support MySpaceIM. Cerulean Studio's ChangeLog says "MYSPACE: Plugin created for MySpaceIM interoperability" for *Trillian Astra, Changes from 3.11 -> 4.0*. . Wikipedia says that Trillian Alpha Build 21 was released to testers in November 2006, but it is not clear what build this plugin was developed for. As of this writing, Trillian Astra is in non-public alpha testing.

I have to wonder if Cerulean Studios used my MySpaceIM Protocol Specification to help develop their plugin. As far as I know, I was the first to crack the authentication scheme. I wouldn't mind if Cerulean used the information from my document; I'm just curious. In the past, Trillian has helped Gaim, so I wouldn't mind having us Pidgin developers help Trillian. We'll still get our msimprpl.

Update 2007-06-10: Scott from Cerulean posted below to set the record straight. Trillian developers have cracked the msim protocol in September 2006, so they were there first. However, msimprpl continues to be the first open source msim protocol plugin; the first public disclosure of the MySpaceIM login scheme. -Jeff

Speaking of other IM clients, I tried out Adium on an OS X machine. A pretty nice looking IM client. Since Adium uses libpurple, it should be possible to use MySpaceIM on Adium with some effort, but I have yet to learn Objective C. Adium has their own Google Summer of Code for this year, and to be clear, I'm working for Pidgin's Google Summer of Code so that is where my priorities are, but I'd still like to see msimprpl under Adium some day. And maybe if I'm lucky, msimprpl will be added to the libpurple-based meebo.

I want for msimprpl to be useful to the largest number of people possible, so I inquired about including it in the official Pidgin builds. Mark Doliner responded that it could be merged into the official tree, but not built by default until it is stable. I'll look into this more after I do some more testing of msimprpl on Windows, since I've been getting a lot of bug reports on that platform. Of course, it is rock solid for me on Linux. Seriously, porting to another platform can reveal many bugs which I need to fix.

I released msimprpl v0.8 this week, but couldn't release the Win32 binaries until later because my personal site's quota, graciously hosted by the SDF Public Access UNIX System, has been exceeded (both disk and bandwidth). I tried posting the releases to the developer.pidgin.im wiki but attachments are limited to 256 KiB, and the site was not really intended for this. Mark Doliner suggested using SourceForge's file release system or a personal site; I opted for the latter and Marc, owner of darkthoughts.net, stepped up the plate to offer hosting for msimprpl (thanks, Marc). The new official site for msimprpl file releases is now http://msimprpl.darkthoughts.net/. Information will continue to be posted on the Pidgin wiki at http://developer.pidgin.im/wiki/MySpaceIM.

Maybe I should make these blog posts a little shorter... but not this time. I only have a few additional things to mention. I finally managed to install Windows, so now I will be able to do more (greatly needed) testing there. By the way, if you dual boot Windows and Linux, you may be interested in my RebootInto open source program. Just installed it. It makes selecting the next boot option (Windows, Linux, Linux safe mode, memtest86, etc.) easily accessible within both Windows and Linux. Instead of rebooting, waiting for the GRUB menu to load, and selecting the next OS to load, you simply make your selection within the OS, and RebootInto takes it from there. This will make the switching between

testing and developing msimprpl in Windows and Linux much easier.

Next up, I plan to work on additional testing on Windows, bug fixes, sending of formatted text, setting your status, and more. I've been getting a handful of messages to my MySpaceIM `msimprpl` account (feel free to IM me, even if I'm offline--I'll get it when I sign on) so I can tell msimprpl usage is growing. Thanks to everyone for their support.

Tags: html libpurple markup msim msimprpl pidgin Pidgin text trillian XML
Source: jeff's blog
Login or register to post comments

Comments

Trillian.

Sat, 07/07/2007 - 00:47 — Anonymous

Hi Jeff - I did not use your documents, and I'm pretty sure I beat you to cracking their silly scheme - our plugin was first released September 2006. I did find your documents around 2 weeks ago and was pretty upset that I hadn't been able to use them when I wrote our MySpace plugin, as they are very nice and you've done a great job. The MySpace IM protocol is absolutely disgusting and I credit you for all the work you've done thus far. :)

Scott @ Cerulean

Login or register to post comments

adium

Mon, 08/13/2007 - 02:30 — Anonymous

I really hope you will be willing to lend a hand or a twist of the ear to get the folks working on adium to msimprpl. It is a feature that is really needed as there is currently no support for OSX by the official client and adium is the de facto libpurple implementation on mac.

Great job By the way!

Cheers,
Kevin

Login or register to post comments