

Gaim project contribution

Jon Igual Brun

Guillermo Barbero Maiz

Table of Contents

Choosing the project.....	1
Understanding the code.....	2
Founding the functions that we wanted to change.....	3
Watch the effects of the changes.....	4
The new code.....	6
Compiling the project.....	7
Mailing developers.....	8

Choosing the project

As the project had to be open source one we went to sourceforge and we began reading witch projects were in the web. After a while we decided to develop a program that we already used, so we could see if we could add options we miss, and the point that we are also the users of that program would make it easier, as we are suppose to know what we want.

Our election was the Gaim program, witch works for instant messaging with different protocols, such as jabber, irc, or msn.

So the next step was to take a look around the messages of the people about the project, so we could see if there is something we also like, and so we could go over it. After reading the suggestions log we decided to get focused on the msn plugin, after all that's the feature we use.

Once done this we had to download the latest version of the code, the usual way would be to download the CVS, but all the time we tried the server couldn't support all the connections it had at that moment (or that was we deducted) and as we haven't got 24 hours Internet access we decided to download the latest version trough the web.

Understanding the code

The first point we began with was reading the documentation there was about the project, the README, INSTALL, HACKING, PROGRAMMING_NOTES files, and the /doc folder (but in this folder the data is the same that can be found in the code, just the comments, made by doxygen). There is a small beginning tutorial in HACKING file, where is explained the program flow of Gaim, while reading this file we read the code as well, beginning from the file witch had the main() function and understanding the skeleton of the program.

Gaim is a GTK project, so we had to know how GTK works, fortunately it wasn't so difficult to understand as we knew that the code we had to write was going to be similar to the code that was written. Basically it is a group of callbacks, that are going to be called due to some specific events, and so long and so forth it just had to be found the function that is called to show contextual menus in the program. That means that when the program is running it just executes few lines of code, the drawing of the main window and the specification of the main callbacks (as button_pressed or button_clicked), and after that just runs the main gtk loop, that is no more than wait for events.

Founding the functions that we wanted to change

The menus we were looking for had some options as “Add a Buddy” that we could look for in the code, and effectively that options where found in the core of the program (gtkblis.c), but the option we wanted to change in this case, was specific of the msn plugin. Fortunately in this function in gtkblis.c there was called a method to get extra callbacks, all we had to make was assure that the callbacks of the plugin are being added. But in the contextual menu of a group (that was where we actually were) there weren't plugin specific callbacks being added, as a group can have different plugin's buddies. So we had to go through the active accounts (or active plugins by the way) and enqueue the callbacks. Once done this we had to specify what was going to return the plugin when the core asks for group callbacks.

We had to take care of where we were writing, as in the core the loops where as appears at the left of the fig. 1 and in the msn plugin was as appears at the right.

<pre>for (i = 0; i < max; i++){ [...] }</pre>	<pre>for (i = 0; i < max; i++) { [...] }</pre>
--	--

fig. 1 - Code styles, core and plugin

Watch the effects of the changes

After coding what we suppose was enough for adding a msn option to the group contextual menu, we runned the program to see if it worked as we wanted and to see if it didn't work as we didn't want. So there was a new option in the menu, it opened a chat room, but there was everybody but you, and that was a bit strange as they could read messages of someone that was not suppose to be in the chat room, and worse of that was that after a while the program exited suddenly.

Hopefully gaim has a -d option when you run it from console that allows you to watch was happening in debug mode, and this way we could see that it halted after trying to loop over the accounts. The problem was the functions that give you the accounts give you a GList that sometimes must be destroyed, and sometimes not (problem that doesn't happen with Java), so the halt was due to a null pointer exception (what in Java would just inform and keep on running possibly). And about not appearing in the list of people that are in the chat room was because the program doesn't see itself as another user, so it had to be treated differently to add to the list of the people witch are in the msn chat room.

So at the end we made the changes, and the program (our version) has this new options¹:



fig. 2 Group menu

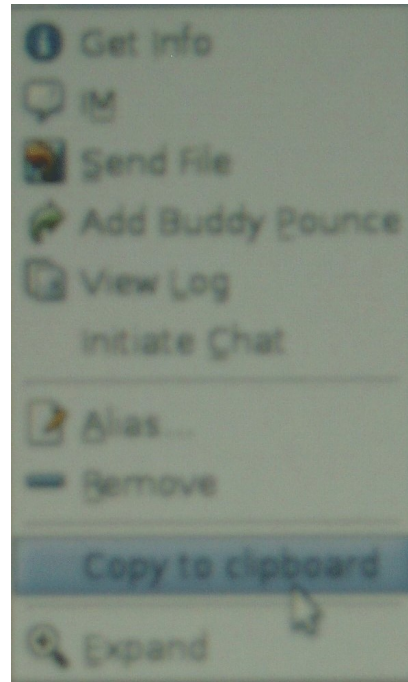


fig. 3 Buddy menu

¹ The blur in the image is due to it wasn't possible to take a screenshot of gaim within a contextual menu, as if it doesn't pass the unhandled events (`print_button_pressed`) to the container (in this case the gnome window manager)

The new code

Here comes part of the new code that we have added:

In `create_group_menu` **in** `gtkblist.c` :

```
//Gets the extended menu for all actives accounts
for (accounts = l = gaim_accounts_get_all_active(); l != NULL; l = l->next){
    gaim_gtk_append_blist_node_proto_menu(menu,((GaimAccount*)l->data)->gc,node);
}
g_list_free(accounts);
```

In `msn_blist_node_menu` **in** `msn.c` :

```
if(GAIM_BLIST_NODE_IS_GROUP(node))
    return msn_group_menu();
```

In `msn_group_menu` **in** `msn.c` :

```
act = gaim_menu_action_new_("Initiate _Chat"),
    GAIM_CALLBACK(initiate_group_chat_cb),
    NULL, NULL);
return g_list_append(NULL, act);
```

In `gaim_gtk_blist_make_buddy_menu` **in** `gtkblist.c` :

```
gaim_new_item_from_stock(menu, _("Copy to clipboard"), NULL,
    G_CALLBACK(gtk_copy_to_clipboard_cb), buddy, 0, 0, NULL);
```

In `gtk_copy_to_clipboard_cb` **in** `gtkblist.c` :

```
GtkClipboard *clipboard = gtk_clipboard_get(GDK_SELECTION_CLIPBOARD);
//We know for sure that the node is a buddy
gtk_clipboard_set_text(clipboard, ((GaimBuddy *)node)->name, -1);
```


Compiling the project

For compiling the project it just was needed a gcc (c compiler, we used the gnu c complier) and some gtk libraries, and as we where in Ubuntu, that uses gnome, that uses gtk, we already had those, and some ssl libraries for the secure communications of some plugins (and msn was among those ones, even though it can be sniffed the dialogs of the people of your own network).

At the beginning we tried to use some environments as Anjuta, an IDE to make easier the develop of the project, but it had to be added all the source files of the project, and there where too many, probably there was some automatic way of doing it (as download the CVS, that Anjuta supports that), but it was impossible to do it, and as the automake was not so bad option we decided to go through other ways.

So we used the gedit text editor and every time we wanted to compile the project we ran `./make` and `./make install` in the source folder. After that just type `gaim` and the program was launched, and we could see the `printf` function in the console.

Mailing developers

We tried to communicate with the main administrator of the project, and as it was no response we mailed the developer of the msn plugin, and still there is no answer, so probably if they answer, it will be after this course finishes. Anyway the mails are the following ones:

mail to seanegan@users.sourceforge.net:

Hi 199625!

I write you in order to offer development help for some aspects of the gaim project, I am with a colleague (we are computer engineers) and we would like to improve the following points:

- Make a chat room with a group
- Copy the email with right click
- Show the nick in the chat room

We have begun to do some development, but we would like to discuss some points before doing more changes, we are doing this for a university project, we'll be glad to work with you

--

This message has been sent to you, a registered SourceForge.net user, by another site user, through the SourceForge.net site. This message has been delivered to your SourceForge.net mail alias. You may reply to this message using the "Reply" feature of your email client, or using the messaging facility of SourceForge.net at:

<https://sourceforge.net/sendmessage.php?touser=1501382>

mail to chipx86@chipx86.com:

Hello there!

I was developing some stuff for gaim for a project for the university, and I have a couple of questions, also I would like to join the developers group, at least I would be glad of seeing that my code is being used!

What my changes concern are about the msn protocol, I'm trying to add an option to the buddy groups to make a chat room with all the buddies that are in it, and I noticed that `gaim_gtk_append_blist_node_proto_menu` in `gtkblist.c` is not being called when the user clicks on a group. Why is that, just because gaim doesn't know if the group contains irc-chats or msn-buddies or anything else?

The new callback I located in `protocols/msn/msn.c`, but I'm afraid that `gtkblist.c` must also be changed for working this new feature.

See you, Guillermo.

P.D. Congratulations for a great code!