

# Gaim's program flow (I)

- Initialize itself
  - Read ~/.gaim/\*.xml
  - Parse command-line options
  - Draw the login window

# Gaim's program flow (II)

- After click the “Sign on/off”
  - Check if the password is written
  - Log in using the account's protocol
- Draw the buddy list
  - Load the local buddy list
  - Update it with the server's notifications
- Wait for events

# GaimPluginProtocolInfo – prpl.h

(I)

```
/**
 * A protocol plugin information structure.
 *
 * Every protocol plugin initializes this structure. It is the
 * gateway between gaim and the protocol plugin. Many of this
 * callbacks can be NULL. If a callback must be implemented, it has
 * a comment indicating so.
 */
struct _GaimPluginProtocolInfo
{
    GaimProtocolOptions options; /**< Protocol options. */

    GList *user_splits; /* A GList of GaimAccountUserSplit */
    GList *protocol_options; /* A GList of GaimAccountOption */

    GaimBuddyIconSpec icon_spec; /* The icon spec. */
}
```

# GaimPluginProtocolInfo – prpl.h

## (II)

```
/**  
 * Returns the base icon name for the given buddy and account.  
 * If buddy is NULL, it will return the name to use for the account's  
 * icon. This must be implemented.  
 */  
const char *(*list_icon)(GaimAccount *account, GaimBuddy *buddy);
```

[...]

```
/**  
 * Allows the prpl to add text to a buddy's tooltip.  
 */  
void (*tooltip_text)(GaimBuddy *buddy, GString *str, gboolean full);
```

[...]

# GaimPluginProtocolInfo – prpl.h

## (III)

[...]

```
/** This must be implemented. */  
void (*login)(GaimAccount *);
```

```
/** This must be implemented. */  
void (*close)(GaimConnection *);
```

[...]

```
void (*set_chat_topic)(GaimConnection *gc, int id,  
                      const char *topic);
```

[...]

```
};
```

# GaimPluginProtocolInfo – msn.c

(I)

```
static GaimPluginProtocolInfo prpl_info =
{
    OPT_PROTO_MAIL_CHECK,
    NULL,                /* user_splits */
    NULL,                /* protocol_options */
    {"png", 0, 0, 96, 96, GAIM_ICON_SCALE_SEND}, /* icon_spec */
}
```

# GaimPluginProtocolInfo – msn.c

## (II)

```
msn_list_icon,          /* list_icon */
```

```
[...]
```

```
msn_tooltip_text,     /* tooltip_text */
```

```
[...]
```

# GaimPluginProtocolInfo - msn.c

## (III)

[...]

```
msn_login,          /* login */  
msn_close,         /* close */
```

[...]

```
NULL,              /* set_chat_topic */
```

[...]

```
};
```



# GaimBlistNode - blist.h

```
/**
 * A Buddy list node. This can represent a group, a buddy, or
 * anything else. This is a base class for struct buddy and struct
 * group and for anything else that wants to put itself in the buddy
 * list.
 */
struct _GaimBlistNode {
    GaimBlistNodeType type;    /**< The type of node this is    */
    GaimBlistNode *prev;    /**< The sibling before this buddy. */
    GaimBlistNode *next;    /**< The sibling after this buddy. */
    GaimBlistNode *parent;    /**< The parent of this node    */
    GaimBlistNode *child;    /**< The child of this node    */
    GHashTable *settings;    /**< per-node settings    */
    void *ui_data;    /**< The UI can put data here.    */
    GaimBlistNodeFlags flags;    /**< The buddy flags    */
};
```

# GaimBlistNodeType - blist.h

```
typedef enum
{
    GAIM_BLIST_GROUP_NODE,
    GAIM_BLIST_CONTACT_NODE,
    GAIM_BLIST_BUDDY_NODE,
    GAIM_BLIST_CHAT_NODE,
    GAIM_BLIST_OTHER_NODE
} GaimBlistNodeType;
```

# GaimBuddy - blist.h

```
/**
 * A buddy. This contains everything Gaim will ever need to know
 * about someone on the buddy list. Everything.
 */
struct _GaimBuddy {
    GaimBlistNode node;/**< The node that this buddy inherits from */
    char *name;        /**< The screenname of the buddy. */
    char *alias;       /**< The user-set alias of the buddy */
    char *server_alias;/**< The server-specified alias of the
                                                                buddy. (i.e. MSN "Friendly Names") */
    void *proto_data; /**< This allows the prpl to associate
                                                                whatever data it wants with a buddy */
    GaimBuddyIcon *icon;/**< The buddy icon. */
    GaimAccount *account;/**< the account this buddy belongs to */
    GaimPresence *presence;
};
```

# How reach the code

- Find where the context menu was created
  - Search for the options Hardcoded in the source files
  - create\_group\_menu founded in gtkblist.c

```
static GtkWidget *
create_group_menu (GaimBlistNode *node, GaimGroup *g)
{
    GtkWidget *menu;
    GtkWidget *item;
    GList *accounts, *l;

    menu = gtk_menu_new();
    gaim_new_item_from_stock(menu, _("Add a _Buddy"), GTK_STOCK_ADD,
        G_CALLBACK(gaim_gtk_blist_add_buddy_cb), node, 0, 0, NULL);
    [...]
    gaim_gtk_append_blist_node_extended_menu(menu, node);
```

# Look for in the msn plugin

- Search any callbacks list
  - msn\_buddy\_menu founded in msn.c
  - it was called from msn\_blist\_node\_menu wich is one of the functions of the plugin protocol

```
msn_blist_node_menu(GaimBlistNode *node)
{
    if(GAIM_BLIST_NODE_IS_BUDDY(node))
    {
        return msn_buddy_menu((GaimBuddy *) node);
    }
    else if(GAIM_BLIST_NODE_IS_GROUP(node))
    {
        return msn_group_menu();//gwilgen
    }
    else return NULL;
}
```

# msn\_group\_menu()

```
static GList *
msn_group_menu()
{
    GList *m = NULL;
    GaimMenuItem *act;

    act = gaim_menu_action_new(_("Initiate _Chat"),
                               GAIM_CALLBACK(initiate_group_chat_cb),
                               NULL, NULL);
    m = g_list_append(m, act);

    return m;
}
```

# initiate\_group\_chat\_cb()

[...]

```
swboard->conv = serv_got_joined_chat(gc, swboard->chat_id,  
                                     "MSN Group Chat");
```

```
swboard->flag = MSN_SB_FLAG_IM;
```

```
for (aux = node->child; aux != NULL; aux = aux->next)
```

```
{
```

```
    buddy = (GaimBuddy *)aux->child;
```

```
    if (GAIM_BUDDY_IS_ONLINE(buddy))
```

```
        msn_switchboard_request_add_user(swboard, buddy->name);
```

```
}
```

[...]

# Ask the plugin for the group menu

- In the `create_group_menu` in the `gtkblis.c` file

```
for (accounts = l = gaim_accounts_get_all(); l != NULL; l = l->next){
    if (GAIM_BLIST_NODE_IS_GROUP(node))
        gaim_gtk_append_blist_node_proto_menu(menu,
            ((GaimAccount *)l->data)->gc,node);
}
```



# Coding styles

- Different coding styles between the core and the plugin
  - Use the same coding style than the context's code