

Apéndice A

Appendice A Conexion.c

```
#include "galaxy/galaxy_all.h"
#define SERVER_FUNCTIONS_INCLUDE "conexion_server.h"
#define USE_SERVER_DATA
#include "galaxy/server_functions.h"

#include "MITRE_galaxy.h"

static int Uttdx = 0;

#define DefaultPrompt "Type a sentence: "
#define DefaultProgram "rec"

static char *ProgName;
static char *Prompt;

static char *OAS[] = {
    "-prompt <str>", "use this prompt", DefaultPrompt,
    "-prog_name <name>", "name of program to send messages to", DefaultProgram,
    NULL
};

static Gal_Frame CreateTextFrame(char *str, MGal_StdinPoll *poll_obj)
{
    return MGal_CreateFullFrame((char *) MGal_GetStdinPollData(poll_obj),
                                GAL_CLAUSE,
                                4,
                                ":input_string", Gal_StringObject(str),
                                GAL_UTTERANCE_ID_FRAME_KEY, Gal_IntObject(Uttdx),
                                ":hyp_id", Gal_IntObject(0),
                                ":mode", Gal_StringObject("typed"));
}

Gal_Frame enable_input(Gal_Frame f, void *server_data)
{
    Gal_Object o;

    if (Gal_GetObject(f, GAL_UTTERANCE_ID_FRAME_KEY)) {
        Uttdx = Gal_GetInt(f, GAL_UTTERANCE_ID_FRAME_KEY);
    } else {
        Uttdx = 0;
    }
    if (Gal_GetObject(f, ":paraphrase_string")) {
        GalUtil_Print(-1, "[Input] %s\n", Gal_GetString(f, ":paraphrase_string"));
    }
    if (Gal_GetObject(f, ":reply_string")) {
        GalUtil_Print(-1, "[Output] %s\n", Gal_GetString(f, ":reply_string"));
    }
    if ((o = Gal_GetObject(f, ":object"))) {
        GalUtil_Print(-1, "[Output]\n");
        /* Now, we test for what kind of object it is. */
        if (Gal_Framep(o)) {
            Gal_PPFrame(Gal_FrameValue(o));
        } else {
            Gal_PrObject(o);
        }
    }
    MGal_ActivateStdinPoll((MGal_StdinPoll *) GalSS_EnvGetCommData((GalSS_Environment *) server_data));
    return (Gal_Frame) f;
}

void *_GalSS_init_server(GalIO_ServerStruct *s, int argc, char **argv)
{
    int i;
```

```

if (!GalUtil_OACheckUsage(argc, argv, OAS, &i))
    exit(1);
if (!GalUtil_OAExtract(argc, argv, OAS, "-prompt", GAL_OA_STRING, &Prompt))
    Prompt = DefaultPrompt;
if (!GalUtil_OAExtract(argc, argv, OAS, "-prog_name", GAL_OA_STRING, &ProgName))
    ProgName = DefaultProgram;
GallO_SetServerMaxConnections(s, 1);
return (void *) NULL;
}

static void FreePoll(void *server_data)
{
    MGal_FreeStdinPoll((MGal_StdinPoll *) server_data);
}

Gal_Frame reinitialize(Gal_Frame f, void *server_data)
{
    GalSS_Environment *env = (GalSS_Environment *) server_data;
    MGal_StdinPoll *p = MGalSS_EnvCreateStdinPoll(Prompt, env,
                                                    CreateTextFrame, 500, 0);

    if (Gal_GetObject(f, ":greeting")) {
        GalUtil_Print(-1, "[Greeting] %s\n", Gal_GetString(f, ":greeting"));
    }

    GalSS_EnvSetCommData(env, (void *) p, FreePoll);
    MGal_SetStdinPollData(p, (void *) ProgName);
    /* set up poll for stdin */
    MGal_ActivateStdinPoll(p);
    return f;
}

```