

libplg Reference Manual

Generated by Doxygen 1.3.5

Sat Oct 16 05:07:58 2004

Contents

1 libplg Data Structure Index	1
2 libplg File Index	1
3 libplg Page Index	1
4 libplg Data Structure Documentation	2
5 libplg File Documentation	3
6 libplg Example Documentation	9
7 libplg Page Documentation	11

1 libplg Data Structure Index

1.1 libplg Data Structures

Here are the data structures with brief descriptions:

plgdinfo (PLGD header type)	2
plginfo (PLG header type)	2

2 libplg File Index

2.1 libplg File List

Here is a list of all documented files with brief descriptions:

libplg.h (Allow use of plugin)	3
libplgd.h (Allow build of plugin)	6
plgtypes.h (Types use by plugin libraries)	7

3 libplg Page Index

3.1 libplg Related Pages

Here is a list of all related documentation pages:

Deprecated List	11
---------------------------------	----

4 libplg Data Structure Documentation

4.1 plginfo Struct Reference

PLGD header type.

```
#include <plgtypes.h>
```

Data Fields

- char [sFileName](#) [MAX_PATH_NAME_LEN]
File to use in the plgin.
- char [sExt](#) [4]
Extension of file to use in the plugin.
- char [sPlgName](#) [33]
Name of the PLG.
- char [sCallProg](#) [MAX_PATH_NAME_LEN]
Internal purpose.
- unsigned char [bTransData](#) [PLG_TRANSMIT_BLOCK_SIZE]
Internal purpose.

The documentation for this struct was generated from the following file:

- [plgtypes.h](#)

4.2 plginfo Struct Reference

PLG header type.

```
#include <plgtypes.h>
```

Data Fields

- char [sPlgFileName](#) [13]
PLG filename.
- char [sTag](#) [4]
PLG TAG must be "PLG\0".
- char [sPlgName](#) [33]
PLG name.
- char [sAuthorName](#) [33]
PLG Athor name.

- char [sIcon](#) [1024]
PLG icon.
- char [bIsSelf](#)
Is the PLG self contained or linked to a program ?
- char [bNbExt](#)
Number of filetype handled by the plugin.
- char * [psExt](#)
Null terminated string with all extension handled by the plugin.

4.2.1 Detailed Description

Presentation part 0~3 : Tag("PLG") & Major version 5~35 : Plugin name 36~67 : Author name 68~1091 : Plugin icon (same as FXE by default)

Technical part 1092 : is the plg is the plugin itself 1093 : number of extension recognized by plugin ("n" is this number) 1094~1094+3*n : extensions list

Examples:

[SampleSearch.c](#).

The documentation for this struct was generated from the following file:

- [plgtypes.h](#)

5 libplg File Documentation

5.1 libplg.h File Reference

Allow use of plugin.

```
#include <plgtypes.h>
```

Functions

- unsigned long [PlgInit](#) (void)
- void [PlgFree](#) (void)
- unsigned long [PlgGetNum](#) (void)
- [PLGINFO](#) * [PlgSearchByExt](#) (const char *sExt, unsigned long *num)
- [PLGINFO](#) * [PlgSearchPlgByName](#) (const char *sName)
- [PLGINFO](#) * [PlgSearchByFileName](#) (const char *sName)
- void [PlgLaunch](#) ([PLGINFO](#) *pPlg, char *sFilePath, char *sExt, void *vSaveData)

Variables

- [PLGINFO](#) * [pPlgList](#)
- unsigned long [dwPlgNum](#)

5.1.1 Detailed Description

Author:

DOUALOT Nicolas (mail:slubman.dndd@laposte.net , web:<http://www.slubman.linux-fan.com>)
SAAS Alain (mail:al1saas@yahoo.fr , web:<http://www.egloo.fr.st>)

Let's you get informations and launch plugins in your prgoram.

5.1.2 Function Documentation

5.1.2.1 void PlgFree (void)

Free memory allocate by library.

This function scan plugin's diretory and build a "database".

See also:

[PlgInit\(\)](#).

Examples:

[SampleSearch.c](#).

5.1.2.2 unsigned long PlgGetNum (void)

Give nummber of valid plugins.

Returns:

Number of plugins found.

See also:

[PlgInit\(\)](#).

5.1.2.3 unsigned long PlgInit (void)

Init library

This function scan plugin's diretory and build a "database".

Returns:

Number of plugins found.

See also:

[PlgFree\(\)](#).

Warning:

Must be call BEFORE any use of other function in library.

Examples:

[SampleSearch.c](#).

5.1.2.4 void PlgLaunch (**PLGINFO** * *pPlg*, char * *sFilePath*, char * *sExt*, void * *vSaveData*)

Launch a PLG.

Parameters:

- pPlg* Pointer to PLGINFO of plugin you want to launch
- sFilePath* Full path of file which must be used by the plugin.
- sExt* Extension of file which must be used by the plugin.
- vSaveData* Data you want to retrieve if the plugin exits "normally".

5.1.2.5 **PLGINFO** * PlgSearchByExt (const char * *sExt*, unsigned long * *num*)

Search PLGs for a given extension.

Parameters:

- sExt* String with the file extension.
- num* Pointer to an unsigned long to store number of PLG in array.

Returns:

- An array of PLGINFO with all matched PLGs.
- NULL otherwise.

Warning:

- You MUST free the array with gm_free.

5.1.2.6 **PLGINFO** * PlgSearchByFileName (const char * *sName*)

Search PLG by filename.

Parameters:

- sName* String with the searched plugin filename.

Returns:

- PLG found: a pointer to PLGINFO of PLG filename *sName*.
- NULL otherwise.

Warning:

- You MUST free the array with gm_free.

5.1.2.7 **PLGINFO*** PlgSearchPlgByName (const char * *sName*)

Search PLG by name.

Parameters:

- sName* String with the searched plugin name.

Returns:

- PLG found: a pointer to PLGINFO of PLG name *sName*.
- NULL otherwise.

Warning:

- You MUST free the array with gm_free.

5.2 libplgd.h File Reference

Allow build of plugin.

```
#include <plgtypes.h>
```

Functions

- int [PlgdInit](#) (void)
- char * [PlgdGetPlgName](#) (void)
- char * [PlgdGetFileName](#) (void)
- char * [PlgdGetFileExt](#) (void)
- void [PlgdQuit](#) (void)
- void [AppExecute](#) (void)

5.2.1 Detailed Description

Author:

DOUALOT Nicolas (mail:slubman.dnnd@laposte.net , web:<http://www.slubman.linux-fan.com>)
SAAS Alain (mail:al1saas@yahoo.fr , web:<http://www.egloo.fr.st>)

Let's you make plugins.

5.2.2 Function Documentation

5.2.2.1 void AppExecute (void)

Quit library

This function free allocated memory, write the PLG_RETURN_FILE then launch plugin caller.

Deprecated

prefer [PlgdQuit\(\)](#) now.

5.2.2.2 char* PlgdGetFileExt (void)

Get the file to treat in plugin.

Returns:

Pointer to null terminated string with filename to treat in plugin.

Examples:

[SamplePlugin.c](#).

5.2.2.3 char* PlgdGetFileName (void)

Get the file to treat in plugin.

Returns:

Pointer to null terminated string with filename to treat in plugin.

Examples:

[SamplePlugin.c](#).

5.2.2.4 char* PlgdGetPlgName (void)

Get the PlgName

Returns:

Pointer to null terminated string with PLG name.

Examples:

[SamplePlugin.c](#).

5.2.2.5 int PlgdInit (void)

Init library

This function parse the PLG_PARAM_FILE.

Returns:

SM_OK if parsing is correct.
!SM_OK otherwise.

See also:

PlgQuit().

Warning:

Must be call BEFORE any use of other function in library.

Examples:

[SamplePlugin.c](#).

5.2.2.6 void PlgdQuit (void)

Quit library

This function free allocated memory, write the PLG_RETURN_FILE then launch plugin caller.

Examples:

[SamplePlugin.c](#).

5.3 plgtypes.h File Reference

Types use by plugin libraries.

Data Structures

- struct [plgdinfo](#)
PLGD header type.
- struct [plginfo](#)
PLG header type.

Defines

- `#define PLG_TAG "PLG"`
Firts part of PLG TAG.
- `#define PLG_VERSION 0`
Major version.
- `#define PLG_BASE_PATH "gp:\\gpsys\\plugins\\"`
Directory where plugins are.
- `#define PLG_PARAM_PATH PLG_BASE_PATH`
Unused.
- `#define PLG_BINARY_PATH PLG_BASE_PATH`
Unused.
- `#define PLG_PARAM_FILE "gp:\\gpsys\\param.tmp"`
Name of the file to trasnmit parameters to plugin.
- `#define PLG_RETURN_FILE "gp:\\gpsys\\exit.tmp"`
Name of the plugin write on normal exit (no power down).
- `#define PLG_TRANSMIT_BLOCK_SIZE 4 * 1024`
Size of data the caller can pass to retrieve on plugin exit.

Typedefs

- `typedef plginfo PLGINFO`
PLG header type.
- `typedef plgdinfo PLGDINFO`
PLGD header type.

5.3.1 Detailed Description**Author:**

DOUALOT Nicolas (mail:slubman.dndd@laposte.net , web:<http://www.slubman.linux-fan.com>)
 SAAS Alain (mail:al1saas@yahoo.fr , web:<http://www.egloo.fr.st>)

Let's you make plugins.

5.3.2 Typedef Documentation**5.3.2.1 typedef struct `plginfo PLGINFO`**

Presentation part 0~3 : Tag("PLG") & Major version 5~35 : Plugin name 36~67 : Author name 68~1091 : Plugin icon (same as FXE by default)

Technical part 1092 : is the plg is the plugin itself 1093 : number of extension recognized by plugin ("n" is this number) 1094~1094+3*n : extensions list

6 libplg Example Documentation

6.1 SamplePlugin.c

An example on how to create a plugin.

This example is extracted from one of my plugins.

```

/*****
 * Project Informations:
 * Project:
 * Version: 1.00b
 * Plateforme: GP 32
 * Copyright: Slubman DevSoft
 * Date: 25/08/2004
 *
 * File Informations:
 * Name: SamplePlugin.c
 * Description: Plugin example
 *
 * Author informations:
 * Author: DOUALOT Nicolas
 * E-Mail: slubman@laposte.net
 * site: http://www.slubman.linux-fan.com
 *****/

/*****
 */
GP32 API
/*****

#include <gpstdlib.h>
#include <gpstdio.h>
#include <gpgraphic.h>
#include <gpfont.h>

#include <libplgd.h>

/*****
 */
GLOBAL VARIABLES
/*****
GPDRAWSURFACE screen[1];
int nflip = 0; // Active screen buffer
int key; // Key pressed
int tick; // Variable for time issue

void GpMain(void *arg)
{
    /* Get the plugin name
    May be usefull if a plugin is an fxe, an have several PLG linking to him
    */
    char *sPlgName = PlgdGetPlgName();

    // Get the file to work on
    char *sFileName = PlgdGetFileName();

    // Get the file extension
    char *sExt = PlgdGetFileExt();

    // Registring screen buffers
    GpLcdSurfaceGet(&screen[0], 0);

    // Init plugin and get parameters
    if (PlgdInit() != SM_OK)
        GpAppExit();

    // Check for the correct extension(s)

```

```

    if (gm_compare(sExt, "XXX") || gm_compare(sExt, "YYY"))
        GpAppExit();

    // I do what i want on sFileName

    // Exit plugin
    PlgdQuit();
}

```

6.2 SampleSearch.c

An example on how to search for plugins.

This example is part of my firmware source code.

```

/*****
*   Informations projet:
*   Projet:
*   Version:
*   Plateforme: GP32
*   Copyright: Slubman DevSoft
*   Date:      25/08/2004
*
*   Informations fichier:
*   Nom:        SampleSearch.c
*   Description:How to use plugi in your programs
*
*   Informations auteur:
*   Auteur:     DOUALOT Nicolas
*   E-Mail:     slubman@linux-fan.com
*   site:       http://www.slubman.linuxfan.com
*****/

/*****
*                                     L'API GP32                                     *
*****/
#include <gpstdlib.h>
#include <gpstdio.h>
#include <gpgraphic.h>
#include <gpfont.h>
#include <libplg.h>

void GpMain(void *arg)
{
    // Local variables
    F_HANDLE fState;
    char *sFile;

    // Program begining
    .
    .
    .

    // Come back from a plugin
    GpFileOpen(PLG_RETURN_FILE, OPEN_R, &fState);
    if (fState)
    {
        // If the caller is me ?
        GpFileRead(fState, tag, gm_strlen("slubFW" VERSION), &i);
        if (!gm_compare(tag, "slubFW" VERSION))
        {
            // Get the information i send in vSaveData
            .
            .
        }
    }
}

```

```

    }
}

.
.
.
// I get a file name for a plugin
sFile = .... //for example(gp:\\GPM\\TRY.BMP)

// Is There installed plugins
if (PlgInit())
{
    PLGINFO *PlgInfo = NULL;
    unsigned long numPlg;
    char *ext = &sfile[gm_lstrlen(sFile) - 3];
    char *Data2Sav = gm_malloc(PLG_TRANSMIT_BLOCK_SIZE);

    PlgInfo = PlgSearchByExt(ext, &numPlg);

    // I let the user choose for a plugin
    .
    .
    .

    // Save program state
    gm_memcpy(Data2Sav + offset, "A_TAG_", gm_lstrlen("A_TAG_"));
    offset += gm_lstrlen("A_TAG_");
    .
    .
    .

    // Launch selected plugin
    PlgLaunch(&PlgInfo[selectPlg], filePath, ext, Data2Sav);

    // If i am here, the plugin launch failed so ifree the list
    gm_free(Data2Sav);
    if (PlgInfo)
        gm_free(PlgInfo);
}

.
.
.

// I free plug-in suystem memory
PlgFree();

.
.
.
}

```

7 libplg Page Documentation

7.1 Deprecated List

Global [AppExecute\(void\)](#) prefer [PlgdQuit\(\)](#) now.

Index

AppExecute
libplgd.h, [6](#)

libplg.h, [3](#)
PlgFree, [4](#)
PlgGetNum, [4](#)
PlgInit, [4](#)
PlgLaunch, [4](#)
PlgSearchByExt, [5](#)
PlgSearchByFileName, [5](#)
PlgSearchPlgByName, [5](#)

libplgd.h, [6](#)
AppExecute, [6](#)
PlgdGetFileExt, [6](#)
PlgdGetFileName, [6](#)
PlgdGetPlgName, [6](#)
PlgdInit, [7](#)
PlgdQuit, [7](#)

PlgdGetFileExt
libplgd.h, [6](#)

PlgdGetFileName
libplgd.h, [6](#)

PlgdGetPlgName
libplgd.h, [6](#)

plginfo, [2](#)

PlgdInit
libplgd.h, [7](#)

PlgdQuit
libplgd.h, [7](#)

PlgFree
libplg.h, [4](#)

PlgGetNum
libplg.h, [4](#)

PLGINFO
plgtypes.h, [8](#)

plginfo, [2](#)

PlgInit
libplg.h, [4](#)

PlgLaunch
libplg.h, [4](#)

PlgSearchByExt
libplg.h, [5](#)

PlgSearchByFileName
libplg.h, [5](#)

PlgSearchPlgByName
libplg.h, [5](#)

plgtypes.h, [7](#)
PLGINFO, [8](#)