

一、Key File Structure

1.Glib中的key file类似于windows下的.ini。数据被分成组，小组名使用[]表示。

例如：

[username]表示一个名为username的组；

以"#"开头的为注释。

2.key files的类型为GKeyfile, 以下为一个key file的例子：

Listing 1. Key-value file (keyfile.conf)

This is an example key-value file

[username]

Name=Foundations of GTK+ Development

#A list of string that are localized

Hi=Hello

Hi[fr]=Bonjour

Hi[de]=Hallo

Hi[es]=Hola

Bool=true;true>false

Nums=1;1;2;3;5;8

Strings=One;One;Two;Three;Five;Eight

Int=42

Doubles=0.0;1.0;50.4

3.Key file有良好的易用性和易读性，可以用来做理想的配置文件；另一种建立配置文件的方法是使用Glib的xml解析。

二、Creating a New Key File

1.GKeyFile对象由g_key_file_new()函数创建，可以手动向GKeyFile中添加组合数据或者直接从文件中导入数据。

```
GKeyFile * g_key_file_new();
```

2.添加或读取数据列表时，常使用分号作为数据间的分隔符，在数据列表中要使用分号时，要在前面加一个反斜杠'/'。

3.使用g_key_file_set_list_separator函数可以更改用作分隔符的字符。

```
void g_key_file_set_list_separator ( GKeyFile *file, gchar separator);
```

4.完成对GKeyfile的编辑后，可通过g_key_file_to_data()函数将数据保存到string中，string空间可由用户自行分配，不再使用时，通过g_free释放。

```
gchar* g_key_file_to_data(GKeyFile *file, gsize *length, GError *error);
```

5.GKeyFile使用完成后，使用g_key_file_free()函数对资源进行释放。

```
void g_key_file_free(GKeyFile *file);
```

三、Setting and Retrieving Data

1.g_key_file_get_groups函数返回一系列以空字符串结尾的字符串集合，每个字符串对应一个组，这个函数同时返回整个数组的长度。

```
gchar** g_key_file_get_groups(GKeyFile *file, gsize *length);
```

2.要访问一条数据，必须先指定它所属的组（这样，每个组就可以包含相同的数据类型了）。具体操作函数见帮助文档

```
g_key_file_get_*((),
g_key_file_set_*(());
```

四、Removing Entries

1.除了可以对key file进行添加、编辑、检索操作外，还可以删除元素。下面三个函数用于在key file 中删除信息：

g_key_file_remove_group()用于删除文件中的特定group,这个group必须是已经存在的；

```
void g_key_file_remove_group(GKeyFile *file,
                             const gchar *group,
                             GError **error);
```

g_key_file_remove_key()用于删除key值，使用时需指定key值所在的组；

```
void g_key_file_remove_key(GKeyFile *file,
                           const gchar *group,
                           const gchar *key,
                           GError **error);
```

g_key_file_remove_comment()用于删除注释，若group和key值均未指定，则删除文件上部的comment，若指定了group未指定key,则删除group上部的comment,同时给出了key和group,则会删除指定的key值的comment。

```
void g_key_file_remove_comment(GKeyFile *file,
                               const gchar *group,
                               const gchar *key,
                               GError **error);
```

五、Parsing an Existing Key File

1.Glib提供了g_key_file_load_from_file()函数对已有的key file文件进行解析，函数调用成功返回TRUE。

```
gboolean g_key_file_load_from_file(GKeyFile *file,
                                   const gchar *filename,
                                   GKeyFileFlags flags,
                                   GError **error);
```

函数中的第三个参数是按位运算的GKeyFileFlags,如下：

G_KEY_FILE_NONE: No flags for the key file will be set.

G_KEY_FILE_KEEP_COMMENTS: All of the comments in the key file should be read so that they can remain in the same position when the file saved. If you do not set thie flag, all comments will be lost.

G_KEY_FILE_KEEP_TRANSLATIONS: All of the translations in the key file should be read so that they will not be lost when you save the file.

2.打开了key file后，使用g_key_file_load_from_data()函数对字符串进行解析，函数中的第三个参数用于指定字符串的长度。

```
gboolean g_key_file_load_from_data(GKeyFile *file,
                                   const gchar *data,
                                   gsize length,
                                   GKeyFileFlags flags,
                                   GError **error);
```

六、A Simple Example

1.Listing1作为一个key-value文件，程序首先创建了一个包含文件中所有信息的结构体setting。为保证注释和translations的存在，在gboolean g_key_file_load_from_file()函数中使用了

G_KEY_FILE_KEEP_COMMENTS and G_KEY_FILE_KEEP_TRANSLATIONS这两个参数

```
#include <glib.h>
```

```
typedef struct
{
```

```

gchar *name, *hello;
gboolean *boolean;
int *nums;
gchar **strings;
int meaning_of_lift;
gdouble *doubles;
}Settings;

int main()
{
    Settings *conf;
    GKeyFile *keyfile;
    GKeyFileFileFlags flags;
    GError *error = NULL;
    gsize length;

    /* Create a new GKeyFile object and a bitwise list of flags */
    keyfile= g_key_file_new();
    flag = G_KEY_FILE_KEEP_COMMENTS | G_KEY_FILE_KEEP_TRANSLATIONS;

    /* Load the GKeyFile from keyfile.conf or return */
    if (!g_key_file_load_from_file(keyfile, "keyfile.conf", flags, &error))
    {
        g_printf("%s\n",error->message);
        g_error_free(error);
        return -1;
    }

    /* Create a new Setting object .If you areusing GTK+2.8 or below, you should use g_new() or g_malloc instead! */
    conf = g_slice_new(Settings);

    /* Read in data from the key file from the group "username". */
    conf->name = g_key_file_get_string(keyfile, "username", "Name", NULL);
    conf->hello = g_key_file_get_locale_string(keyfile, "username", "Hi", "es", NULL);
    conf->boolean = g_key_file_get_boolean_list(keyfile, "username", "Bool", &length, NULL);
    conf->nums = g_key_file_get_integer_list(keyfile, "username", "Nums", &length, NULL);
    conf->strings = g_key_file_get_string_list(keyfile, "username", "Strings", &length, NULL);
    conf->meaning_of_lift = g_key_file_get_integer(keyfile, "username", "Int", NULL);
    conf->doubles = g_key_file_get_double_list(keyfile, "username", "Double", &length, NULL);
    return 0;
}

```

七、编译

```
gcc main.c -o main `pkg-config --cflags --libs glib-2.0`
```