

对外头文件及 API 文档自动生成设计

背景

本案是软件模块化设计框架下的一个配套工具。

用户使用一个软件模块，至少需要三样东西，分别是 `module.lib`, `module.h` 以及 `module.doc`，即模块的库（或源码）文件，头文件和说明文档。在实际工作中，基于模块的功能开发，可能会遇到一些问题：

- 1、模块对内和对外使用相同的头文件，使得用户可能使用到并不希望对外的声明或函数；并且，内外声明信息混合，用户使用的学习成本也高。
- 2、大量的头文件，开发时不知道要引用哪一个；
- 3、文档依赖人工维护，模块代码更新后，API 说明文档却没有同步更新。

解决方案是，将模块里每个头文件的对外声明内容标识出来，通过工具汇总至同一个头文件里，并提取对外声明中的函数说明信息，自动生成 API 使用说明文档。这样，既解决了头文件的引用问题，也解决了 API 文档的同步更新问题。

详细设计

对外声明约定

对外数据声明的约定：

```
//-----EXTERN_PARAMETER_START-----//  
//-----EXTERN_PARAMETER_END-----//  
分别标识对外数据声明的起始和结束
```

对外函数声明的约定：

```
//-----EXTERN_FUNCTION_START-----//  
//-----EXTERN_FUNCTION_END-----//  
分别标识对外函数声明的起始和结束
```

对外数据和函数声明区间中的内容，会被统一汇总到一个对外头文件中，供用户引用。API 文档自动生成工具，会直接拷贝对外数据声明部分的内容，并将对外函数声明区间的内容解析后生成 API 说明表格。

函数注释格式约定

以 `SetTimer` 函数为例，函数注释内容为：

```
/******
```

```

* Procedure:      SetTimer
* Function:       设置一个 Timer
* Parameter:      hParent: Timer 附属的窗口/创建者;
*                 wID: Timer 的 ID 值;
*                 wStyle: Timer 的风格;
*                 wElapse: Timer 的溢出值, 单位 ms
* Result:         TRUE 成功 | FALSE 失败
* Description:    同一个窗口下的 timer ID 不可重复。hParent 不能为空
*                 wElapse 不能小于最小步进
*****
* LOG    DATE    AUTHOR    ACTION
*****
* 2011-8-30                                Create this procedure
*****/
INT SetTimer(HWND hParent, WNDID bID, WNDSTYLE wStyle, WNDTIME wElapse);

```

api 表格内容与注释内容的对应关系为:

API 项	注释项	说明
标题	<i>Procedure</i>	用于生成 API 对应的标题项
函数	紧接注释的 API 声明内容	
功能	<i>Function</i>	
参数	<i>Parameter</i>	参数项与注释之间也用:号分隔 一个参数项对应表格中一行
返回	<i>Result</i>	返回值内容与说明之间也用:号分隔 一个注释行对应表格中的一行
说明	<i>Description</i>	

支持头文件包含

模块在开发维护的过程中, 可能也会用到统一的头文件, 文件内会包含各个子模块。工具会自动识别 `include` 语句, 并逐项展开并提取其中对外声明的内容。

以 `gui` 模块为例, 其中可能有 `gui.h` 文件, 文件内会包含各个子模块的头文件, 如

```
#include "GuiMgrEx.h"
```

工具在解析 `gui.h` 的时候, 会识别到此包含, 并将 `GuiMgrEx.h` 文件展开, 根据对外声明约定, 获取内容。最终生成的对外版本 `gui.h`, 不会再有 `include` 这类语句, 仅所有文件对外声明的内容。

对于规模稍大点的模块, 每个包含进来的子模块, 都会在文档中为其分配一个独立的标题项, 使得文档内容的分类更清晰。

实现方案

头文件

为了使源文件配置起来方便，可以对模块头文件中 include 语句的使用进行约定，即要求 include 的文件名要包含相对路径。以 Gui.h 为例：

```
#ifndef    __GUI_H
#define    __GUI_H

#include "../logic/Include/GuiMgrEx.h"
#include "../logic/Include/GuiTimerEx.h"
.....

#endif
```

工具根据相对路径找到 GuiMgrEx.h 和 GuiTimerEx.h 后，将内容拷贝到新的 Gui.h 中。如果找不到文件，则 PC 工具报错。

配置文件

用于说明输入输出信息。

采用 ini 文件格式，默认文件名为 config.ini，与工具放在同个目录下。

内容示例：

```
[INPUT]
GUI = "D:\code\program\python\api\file\output\include\GuiTest.h"

[OUTPUT]
DOCX = "D:\code\program\python\api\api.docx"
DIR = "D:\code\program\python\api\sdk"
```

段名称 INPUT 标识输入文件，即待解析的头文件列表

段名称 OUTPUT 标识输出文件，即“DIR”标识头文件展开后存放的目录，“DOCX”标识 API 使用说明内容存储至哪个.docx 文件。**.docx 文件要事先做好文档格式。**

补充

- 展开后的文件宏

文件宏使用文件名的大写形式。如：

```
GUI = "D:\code\program\python\api\file\output\include\GuiTest.h"
```

展开后，GuiTest.h 文件头部的内容为：

- ```
#ifndef __GUITEST_H
#define __GUITEST_H
```
- 头文件展开后，子头文件的标识
- ```
//-----GuiTimerEx.h-----//
```
- 标识后续内容来自 GuiTimerEx.h

函数说明

- 函数宏的声明方式
- 为了使用户能直观了解函数宏的使用方式，为函数宏做的函数说明后，紧跟函数宏的虚拟函数原型，如：

```
/******
.....
*****/

//INT DlistSetTitle(HWND hCtrl, CHAR *pcTitile);
```

函数宏的虚拟原型使用“//”单行注释开头，以“;”符号结尾。这样，工具就能直接获取到函数原型，并填入 API 说明表格内。

- 函数注释示例

■ 原型函数

```
/******
* Procedure:      KillTimer
* Function:       删除一个 Timer
* Parameter:      hParent: Timer 附属的窗口/创建者;
*                wID: Timer 的 ID 值
* Result:         1 成功 | 0 失败
* Description:    如果 hParent 为 0xFFFFFFFF, 则函数不检查 hParent 的匹配;
*                如果 bID 为 0xFF 则不检查 bID 的匹配
*****
* LOG      DATE      AUTHOR      ACTION
*****
* 2011-8-30                                Create this procedure
*****/

INT KillTimer(HWND hParent, WNDID bID);
```

■ 函数宏

```
/******
* Procedure:      DlistCreate
* Function:       创建双列表控件
* Parameter:      ptDlistData:数据结构体变量指针。NULL 代表先创建,
*                数据后续提供
* Result:         控件句柄
```

```
* Description: 无
*****

* LOG      DATE      AUTHOR      ACTION
*****

* 2018-12-11      ccq      Create this procedure
*****/

//HWND DlistCreate(CDlistData *ptDlistData);
```

文档示例

以 GuiTimerEx.h 为例，生成的使用说明文档大致是以下格式：

数据类型说明

```
#define TS_HOLD      1
#define TS_TOPARENT  2
```

函数接口说明

1.1.1 SetTimer

函数	STDCALL INT SetTimer(HWND hParent, WNDID bID, WNDSTYLE wStyle, WNDTIME wElapse)	
功能	设置一个定时器	
参数	hParent	Timer 附属的窗口/创建者
	bID	Time ID，用于区分不同的 Timer
	wStyle	Timer 类型
	wElapse	定时器超时时间，单位 ms
返回	TRUE 成功 FALSE 失败	
说明	同一个窗口下的 timer ID 不可重复。hParent 不能为空，wElapse 不能小于最小步进	
示例		

1.1.2 KillTimer

函数	STDCALL INT KillTimer(HWND hParent, WNDID bID)
功能	删除定时器

输入	hParent	Timer 附属的窗口/创建者
	bID	Time ID，用于区分不同的 Timer
输出	无	
返回	TRUE 成功 FALSE 失败	
说明	如果 hParent 为-1，则函数不检查 hParent 的匹配。如果 bID 为-1 则不检查 bID 的匹配	
示例		

hehuoxuetang