DoxygenExample.cpp File Reference

File containing example of doxygen usage for quick reference. More...

```
#include <systemheader1.h>
#include <systemheader2.h>
#include <box/header1.h>
#include <box/header2.h>
#include "local_header1.h"
#include "local_header2.h"
```

Classes

struct	**BoxStruct_struct** Use brief, otherwise the index won't have a brief explanation. More...

Typedefs

typedef enum **BoxEnum_enum**	**BoxEnum** Use brief, otherwise the index won't have a brief explanation. More...
typedef struct **BoxStruct_struct**	**BoxStruct** Use brief, otherwise the index won't have a brief explanation. More...

Enumerations

enum	**BoxEnum_enum** { **BOXENUM_FIRST**, **BOXENUM_SECOND**, **BOXENUM_ETC** } Use brief, otherwise the index won't have a brief explanation. More...

Functions

BOXEXPORT **BoxStruct** *	**Box_The_Function_Name** (BoxParamType1 param1, BoxParamType2 param2) Example showing how to document a function with Doxygen. More...
BOXEXPORT void *	**Box_The_Second_Function** (void) A simple stub function to show how links do work. More...
BOXEXPORT void	**Box_The_Last_One** (void)

Detailed Description

File containing example of doxygen usage for quick reference.

Author

My Self

Date

9 Sep 2012 Here typically goes a more extensive explanation of what the header defines. Doxygens tags are words preceeded by either a backslash \ or by an at symbol @.

See also

http://www.stack.nl/~dimitri/doxygen/docblocks.html

http://www.stack.nl/~dimitri/doxygen/commands.html

Typedef Documentation

◆ BoxEnum

typedef enum **BoxEnum_enum BoxEnum**

Use brief, otherwise the index won't have a brief explanation.

Detailed explanation.

◆ BoxStruct

typedef struct **BoxStruct_struct BoxStruct**

Use brief, otherwise the index won't have a brief explanation.

Detailed explanation.

Enumeration Type Documentation

◆ BoxEnum_enum

enum **BoxEnum_enum**

Use brief, otherwise the index won't have a brief explanation.

Detailed explanation.

Enumerator	
BOXENUM_FIRST	Some documentation for first.
BOXENUM_SECOND	Some documentation for second.
BOXENUM_ETC	Etc.

Function Documentation

◆ Box_The_Function_Name()

BOXEXPORT **BoxStruct*** Box_The_Function_Name (BoxParamType1 param1,
BoxParamType2 param2
)

Example showing how to document a function with Doxygen.

Description of what the function does. This part may refer to the parameters of the function, like `param1` or `param2`. A word of code can also be inserted like `this` which is equivalent to `this` and can be useful to say that the function returns a `void` or an `int`. If you want to have more than one word in typewriter font, then just use <tt>. We can also include text verbatim,

```
like this
```

Sometimes it is also convenient to include an example of usage:

```
BoxStruct *out = Box_The_Function_Name(param1, param2);
printf("something...\n");
```

Or,

```
pyval = python_func(arg1, arg2)
print pyval
```

when the language is not the one used in the current source file (but **be careful** as this may be supported only by recent versions of Doxygen). By the way, **this is how you write bold text** or, if it is just one word, then you can just do **this**.

Parameters

param1 Description of the first parameter of the function.

param2 The second one, which follows `param1`.

Returns

Describe what the function returns.

See also

Box_The_Second_Function

Box_The_Last_One

http://website/

Note

Something to note.

Warning

Warning.

◆ Box_The_Last_One()

BOXEXPORT void Box_The_Last_One (void)

Brief can be omitted. If you configure Doxygen with `JAVADOC_AUTOBRIEF=YES`, then the first Line of the comment is used instead. In this function this would be as if

```
@brief Brief can be omitted.
```

was used instead.

◆ Box_The_Second_Function()

BOXEXPORT void* Box_The_Second_Function (void)

A simple stub function to show how links do work.

Links are generated automatically for webpages (like http://www.google.co.uk) and for structures, like **BoxStruct_struct**. For typedef-ed types use **BoxStruct**. For functions, automatic links are generated when the parenthesis () follow the name of the function, like **Box_The_Function_Name()**. Alternatively, you can use **Box_The_Function_Name**.

Returns

`NULL` is always returned.