

C++ Coding Style

Author: Anita & Ted

Date: 2020.05.06

Document Version: v1.0.1

DOCUMENT STATUS SHEET

Author

Position	Name	Contact Phone Ext.
Engineer	Anita.Lee	#1151

Revision History

Version	Issue date	Author/Editor	Description/Summary of Changes
[1.0.0]	2020.04.10	Anita.Lee	First release.
[1.0.1]	2020.05.06	Ted.Pan	重新調整格式

Reviewed By

Version	Issue Date	Name	Position	Review Date

TABLE OF CONTENTS

1	命名規範	4
1.1	Class & Function 名稱	4
1.2	Function 內變數	5
1.3	Const	5
1.4	Static	5
1.5	Gol	5
1.6	Class member	5
1.7	Enumerator Names	5
1.8	Struct	6
1.9	Event	6
1.10	Remark	6
1.10.1	Principle	6
2	標頭檔	7
2.1.1	每一個 .cpp 檔都應該有一個對應的.h 檔	7
2.1.2	模板與行內函數的定義放在同樣的檔案作為宣告	7
2.1.3	所有的標頭檔應該要包含#define 保護，以防止多重載入。	7
2.1.4	其名稱的格式為<專案名稱>_<路徑>_<檔名>_H_。	7
3	行內函數	7
4	SCOPING	7
5	括號樣式	8
5.1.1	If	9
5.1.2	Switch	10
5.1.3	Other	10

1 命名規範

- **一般規則**

c	char
by	byte
n	int or short
x,y	the xy-coordinates of the point => (x, y)
b	bool
w	WORD
dw	DWORD
s	string
sz	the null terminator at the end of string
h	handle
p	pointer
e	enumeration
st	structure

- **Pascal Casing**

標識符中的第一個字母和每個後續串聯詞的第一個字母均大寫。

BackColor

- **Camel Casing**

標識符的第一個字母為小寫字母，每個後續串聯詞的第一個字母均大寫。

backColor

- **Uppercase**

標識符中的所有字母均為大寫。

IO

1.1 Class & Function 名稱

- 使用 **Pascal Casing** 編寫。
- Class 定義時請加前綴 **C**，但變數則不加

```
class CNode; // 類名
class CLeafNode; // 類名
void Draw( void ); // 函數名
void SetValue( int value ); // 函數名
```

1.2 Function 內變數

- 使用 **Camel Casing** 編寫.
- 命名方式採名詞或名詞片語或偶爾的形容詞片語來命名，**避免**是動詞.

```
bool      bFirstTime;  
Int       nUpdateCount;
```

1.3 Const

- 全用大寫的字母，用下劃線分割單詞.

```
const int MAX = 100;  
const int MAX_LENGTH = 100;
```

1.4 Static

- 靜態變量加前綴 **s_**.

```
static bool s_bTurnOn;  
static EWorkState s_eTask;
```

1.5 GoI

```
BOOL g_bXXXX;
```

1.6 Class member

- 類的數據成員加前綴 **m_** (表示 member)，這樣可以避免數據成員與成員函數的參數同名.

```
BOOL      m_bMove;  
Int       m_nIndexX;  
double    m_dDiePitchX;  
GVirtualTimer m_SendCmdTimer;
```

1.7 Enumerator Names

- 命名前面加 **E**
- 內容前面編寫前面加 **E_**.

```
typedef enum  
{  
    E_CAMERA_TRIG_MODE_SOFT = 0,  
    E_CAMERA_TRIG_MODE_HARD,  
} ECameraTrigMode;
```

- OP Code / Error / Alarm / Register(SDR / IR / UDR) 部分一律以 C++ 為主.

1.8 Struct

- 結構體的成員命名規則和普通變量一致.

```
typedef struct tagAxisUnit
{
    GMotorizedAxisUnit *pAxisX;
    GMotorizedAxisUnit *pAxisY;
} TAxisUnit;
```

1.9 Event

```
// Wafer scan unit.
m_IShapeWaferScanUnitConfig.ModuleName           = "<WT_IShapeWaferScan>";
m_IShapeWaferScanUnitConfig.Error.CallbackInitDataFunctionNotImplement = E_SYS_ERR_WT_INIT_DATA_FUNCTION_NOT_IMPLEMENT;
m_IShapeWaferScanUnitConfig.Delegate.InitData      = ::OnWTWaferScanUnit_InitData;
m_IShapeWaferScanUnitConfig.Delegate.CheckFixedScanBoundary = ::OnWTWaferScanUnit_CheckFixedScanBoundary;
```

格式: <On><Module>_<Function name>

1.10 Remark

1.10.1 Principle

- 標識符應當直觀且可以拼讀，可望文知意，不必進行“解碼”。
- 程序中不要出現標識符完全相同的局部變量和全局變量，儘管兩者的作用域不同而不會發生語法錯誤，但會使人誤解。
- 變量的名字應當使用“名詞”或者“形容詞 + 名詞”。
- 全局函數的名字應當使用“動詞”或者“動詞 + 名詞”。類的成員函數應當只使用“動詞”，被省略掉的名詞就是對象本身。
- 儘量避免名字中出現數字編號，如 Value1, Value2 等，除非邏輯上的確需要編號。這是為了防止程序員偷懶，不肯為命名動腦筋而導致產生無意義的名字

2 標頭檔

2.1.1 每一個 .cpp 檔都應該有一個對應的.h 檔

2.1.2 模板與行內函數的定義放在同樣的檔案作為宣告

2.1.3 所有的標頭檔應該要包含#define 保護，以防止多重載入。

2.1.4 其名稱的格式為<專案名稱>_<路徑>_<檔名>_H_。

```
#ifndef FOO_BAR_BAZ_H_
#define FOO_BAR_BAZ_H_

...

#endif // FOO_BAR_BAZ_H_
```

3 行內函數

只有在函式程式碼少於或等於 10 行時才將它宣告為行內函式。

4 SCOPING

一個變數的 scope 盡量越小越好；若一個變數的 scope 越大，在它的生命週期(lifetime)結束之前該變數被存取的機會就越大

5 括號樣式

AVOID:

```
class MyClass {  
    void DoSomething() {  
        if ( someTest ) {  
            // ...  
        } else {  
            // ...  
        }  
    }  
}
```

PREFER:

```
class MyClass  
{  
    void DoSomething()  
    {  
        if ( someTest )  
        {  
            // ...  
        }  
        else  
        {  
            // ...  
        }  
    }  
}
```


5.1.1 If

AVOID:

```
if ( somTest ) doSomethingElse();
```

PREFER:

```
if ( somTest )  
    doSomethingElse();
```

```
if ( somTest )  
    doSomething();  
else  
    doSomethingElse();
```

```
if ( somTest )  
{  
    doSomething();  
}  
else  
{  
    doSomethingElse();  
    UpdateData();  
}
```

bFirstTime 格式(強制加大括號{}):

```
if ( bFirstTime )  
{  
    .....  
    .....  
}
```

多個判斷式:

```
if ( ( E_WORK_OK == eSynTask1 ) && ( E_WORK_OK == eSynTask2 ) && ( E_WORK_OK == eSynTask3 ) &&  
( E_WORK_OK == eSynTask4 ) && ( E_WORK_OK == eSynTask5 ) && ( E_WORK_OK == eSynTask6 ) )  
    return TRUE;
```

這樣的判斷式需要用多行表示，如下：

```
if ( ( E_WORK_OK == eSynTask1 )  
    && ( E_WORK_OK == eSynTask2 )  
    && ( E_WORK_OK == eSynTask3 )  
    && ( E_WORK_OK == eSynTask4 )  
    && ( E_WORK_OK == eSynTask5 )  
    && ( E_WORK_OK == eSynTask6 ) )  
{  
    return TRUE;  
}
```

且一定要加上{}

5.1.2 Switch

PREFER:

```
switch ( somTest )
{
    case ( int )EWTCtrlServerCommand.OpenMappingMap:
        return OpenMappingMap();
    case ( int )EWTCtrlServerCommand.MatchMappingMap:
        return MatchMappingMap();
    default:
        iMPI.Log.Logger.Fatal( "WT CtrlServer recieved exception command:[{ 0 }]", command );
        return ( int )EWTCtrlServerResult.Fail;
}
```

5.1.3 Other

特殊符號要空格 =, ==, (,) , + , - , * , /

nSeq 建議寫法:

AVOID:

```
nSeq = m_bRgnGrabExpDoneMov ? E_WT_PRESCAN_SEND_VISION_REGION_GRAB : E_WT_PRESCAN_SEND_VISION_REGION_GRAB_BEGIN;
```

PREFER:

```
if ( m_bRgnGrabExpDoneMov )
    nSeq = E_WT_PRESCAN_SEND_VISION_REGION_GRAB;
else
    nSeq = E_WT_PRESCAN_SEND_VISION_REGION_GRAB_BEGIN;
```