C# Coding Style

Author: Anita & Ted
Date: 2020.10.26
Document Version: v1.0.5

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]	2020.02.10	Anita.Lee	First release.
[1.0.1]	2020.04.24	Anita.Lee	Change fields format. (Count)
[1.0.2]	2020.05.06	Anita.Lee	新增 if 多條件式判斷和空行寫法準則
[1.0.3]	2020.06.29	Anita.Lee	Add Extension.
[1.0.4]	2020.07.20	Anita.Lee	新增 using & this.規範
[1.0.5]	2020.10.26	Anita.Lee	Add using dll 順序 Add 元件縮寫命名規範

Reviewed By

Version	Issue Date	Name	Position	Review Date

TABLE OF CONTENTS

1 命名規範 4

1. Namespaces 4
2. Class & Interfaces 4
3. Enum 5
4. Methods 5
5. Fields 5
6. Properties 5
7. Parameters 6
8. Event 6
9. Names of Assemblies & DLLs 6
10. Remark 7

1.10.1 Principle 7
1.10.2 Abbreviation 7
1.10.3 Specific Name 7
1.10.4 Other 7

2 宣告規範 8

1. Access Level Modufiers 8
2. Fields & Variables 8
3. Classes 8
4. Interfaces 8
5. this 8

3 間距 9

1. Indentation 9

3.1.1 Block 9
3.1.2 Line Wraps 9

4 括號樣式 10

4.1.1 If 11
4.1.2 Switch 11

5 新行 11

5.1.1 Method 11
5.1.2 For & While 11
5.1.3 Switch 11

6 EXTENSION(擴充方法) 12
7 USING 12
8 USING DLL 13
9 元件縮寫 13

1 命名規範

On the whole, naming should follow C# standards.

- ***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軍為大寫。

  ```
  IO
  ```

- ***Capitalization Rules for Identifiers***

  當標識符由多個單詞組成時，盡量避免在單詞之間使用下劃線（例如，下劃線（_）或連字符（-））。而是使用大小寫指示每個單詞的開頭。

1. Namespaces

使用 Pascal Case 編寫, 但在 GUI 或 HUD 之類的首字母縮寫時，可以是大寫的:

AVOID:

```
com.raywenderlich.fpsgame.hud.healthbar
```

PREFER:

```
RayWenderlich.FPSGame.HUD.Healthbar
```

格式: <Company>.(<Product>|<Technology>)[.<Feature>][.<Subnamespace>]

Ex: Microsoft.WindowsMobile.DirectX

2. Class & Interfaces

- 使用 Pascal Casing 編寫. Ex: RadialSlider
- 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.

3. Enum

- 使用 Pascal Casing 編寫，並且前面加”E”.
- 列舉型別的名稱,不應該在列舉值名稱中加入列舉型別名稱.
- 請不要在列舉型別上使用 Enum 做為尾碼.

```
public enum ETeams
{
    Alpha,
    Beta,
    Delta,
}
```

4. Methods

使用 Pascal Casing 編寫.
命名請使用動詞或動詞片語.

AVOID:

```
CameraLive();
```

PREFER:

```
LiveCamera();
```

5. Fields

private 使用 camel Casing 編寫. Ex: private string _userName

private const XXX_XXX_XXX 字母均為大寫.

Static 前需加”s”. Ex: private static string s_userName

6. Properties

使用 Pascal Casing 編寫.
命名方式採名詞、名詞片語或形容詞來命名.

```
public int PageNumber
{
    get { return _pageNumber; }
    set { _pageNumber = value; }
}
```

7. Parameters

使用 camel Casing 編寫.

AVOID:

```
void DoSomething(Vector3 Location)
```

PREFER:

```
void DoSomething( Vector3 location )
```

8. Event

使用 Pascal Casing 編寫.

命名方式採動詞或動詞片語來命名.

```
public event Action<int> ValueChanged;
```

- 使用現在式和過去式為事件名稱提供過去和將來的概念.

  (1) 視窗關閉之前引發的關閉事件可稱為 Closing.

  (2) 視窗關閉之後引發的關閉事件可稱為 Closed.

- Microsoft Visual 內建 Event 採下列方式

```
btnJogUp.Click += new System.EventHandler(btnDirectionMove_Click);

private void btnDirectionMove_Click(object sender, EventArgs e)
{

}
```

9. Names of Assemblies & DLLs

- 請為程序集 DLL 選擇名稱，這些名稱建議使用大量功能. Ex: System.Data
- 根據以下模式命名 DLL：

  <Company>.<Component>.dll 其中<Component>包含一個或多個以點分隔的子句.

  Ex: Contoso.WebControls.dll

10. Remark

1.10.1 Principle

- 選擇能展現意圖的名稱，不需要額外的註解.
- 能夠被唸出來、被搜尋的命名.
- 每個概念只使用一種字詞，避免使用多個類似的字詞.
- 使用解決方案領域命名，例如演算法名稱、模式(Pattern)名稱.
- 可讀性優先於簡潔性. (CanScrollHorizontally 優於 ScrollableX)，且需參考常用的命名 (HorizontalAlignment 優於 AlignmentHorizontal)
- 避免使用匈牙利命名法(將變數型別寫入命名)，列如 strName
- 避免使用誤導的命名. Ex: name1, name2

1.10.2 Abbreviation

- 請勿將縮寫或縮寫用作標識符名稱的一部分.

  Ex: 使用 OnButtonClick 而不是 OnBtnClick.
- 正確使用縮寫 (沒事不要用縮寫)
  - 兩字皆為大寫(如 : "IO" , "DB" , system.IO)
  - 三個字以上 (如 : "Xml")

1.10.3 Specific Name

對於類型名稱，請使用語義上有意義的名稱，而不要使用特定於語言的關鍵字。例如，GetLength 是比 GetInt 更好的名稱.

1.10.4 Other

- Bool 可使用 ”Can” , “Is” , “Has” 當前致詞 (Ex: bool isPlay)
- 使用後致詞
  - System.EventArgs : AppleEventArgs
  - System.Exception : CmdException
  - System.Attribute : LoginAttribute

2 宣告規範

1. Access Level Modufiers

明確定義 Classes, Methods, Member variables 的訪問級別.
public / protected / internal / private

2. Fields & Variables

每行只宣告一項.
AVOID:

```
string username, twitterHandle;
```

PREFER:

```
string username;
string twitterHandle;
```

3. Classes

每個 Class.cs 內盡量只存在一個 Class.

4. Interfaces

Interfaces 前面需加 “I”.
AVOID:

```
RadialSlider
```

PREFER:

```
IRadialSlider
```

5. this

說明: public & property 請前面加上 this.，而 private 則不需加.
Ex:

this.XXX <= (public & property)

_XXX <= (private)

3 間距

1. Indentation

3.1.1 Block

使用 Tab 間距(預設為 4 空格)，提升可讀性.

AVOID:

```
for ( int i = 0; i < 10; i++ )
{
  Debug.Log( "index=" + i );
}
```

PREFER:

```
for ( int i = 0; i < 10; i++ )
{
    Debug.Log( "index=" + i );
}
```

3.1.2 Line Wraps

當 code 過長時，在換行時請在等號後使用 Tab 間距(預設為 4 空格).

AVOID:

```
string message = String.Format("This function:<btnPreAlignerVac_Click> is not supported in this machine type:<0>.", Const.Machine.Information.MACHINE_TYPE);
```

PREFER:

```
string message = String.Format("This function:<btnPreAlignerVac_Click> is not supported in this machine type:<0>.",
                    Const.Machine.Information.MACHINE_TYPE);
```

4 括號樣式

AVOID:

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

PREFER:

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

4.1.1 If

AVOID:

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

PREFER:

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

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

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

4.1.2 Switch

PREFER:

```
switch ( somTest )
{
    case ( int )EWTControlServerCommand.OpenMappingMap:
        return OpenMappingMap();
    case ( int )EWTControlServerCommand.MatchMappingMap:
        return MatchMappingMap();
    default:
        iMPI.Log.Logger.Fatal( "WT ControlServer recivied exception command:[{0}]", command );
        return ( int )EWTControlServerResult.Fail;
}
```

5 新行

5.1.1 Method

若編排方以 Block 為一區塊的情況下，return & break 前需新增一行.

5.1.2 For & While

break & continue & return 一般建議前面需新增一行(視情況).

5.1.3 Switch

break 無需新增一行.

6 EXTENSION(擴充方法)

使用時機：當使用靜態類別可以優先考慮使用.

規則：

(1) Class 名稱： XXXExtensions

(2) Method：

欲擴充的類別必須放在參數列的第一個位置。也就是說，關鍵字 this 只能用在第一個參數型別。如果你將 this 加在第二個或後面的參數，程式將無法過編譯。

```
/// <summary>
/// Extension methods for Image buffer class.
/// </summary>
public static class ImageBufferExtensions
{
    #region >>> Drawing Method <<<

    /// <summary>
    /// Draw loop.
    /// </summary>
    public static void Draw( this ImageBuffer image, Action<Graphics> action )
    {
        if ( action == null )
            return;

        using ( Bitmap bmp = image.ContainToBitmap() )
        using ( Graphics gx = Graphics.FromImage( bmp ) )
        {
            action( gx );
        }
    }

    #endregion

    >>> Pixel Manipulating <<<

    >>> Intensity <<<

    >>> Buffer Conversion <<<

    >>> Image Processing <<<
}
```

7 USING

當使用到以下元件時，請使用 using 額外包起來使用:

a. Bitmap
b. Pen
c. Brush
d. TCP / IP Port
e. RS232
f. OpenFileDialog
g. Read / write

8 USING DLL

Using dll 順序原則:

.net

.third-party

MPI

Project

9 元件縮寫

Control #	Control Category	Control Type	Prefix	Example
1W	WINDOWS	Binding	bdg	bdgName
2W	WINDOWS	Bitmap	bmp	bmpName
3W	WINDOWS	Brush	brh	brhName
4W	WINDOWS	Button	btn	btnName
5W	WINDOWS	CheckBox	cb	cbName
6W	WINDOWS	CheckedBoxList	cbl	cblName
7W	WINDOWS	Color	clr	clrName
8W	WINDOWS	ColorPalette	clrp	clrpName
9W	WINDOWS	ComboBox	cb	cbName
10W	WINDOWS	ContextMenu	ctm	ctmName
11W	WINDOWS	CrystalReportViewer	crv	crvName
12W	WINDOWS	Cursor	csr	csrName
13W	WINDOWS	DataGrid	dg	dgName
14W	WINDOWS	DataGridColumn	dgc	dgcName
15W	WINDOWS	DateTimePicker	dtp	dtpName
16W	WINDOWS	DialogControl	dc	dcName
17W	WINDOWS	DirectoryEntry	de	deName
18W	WINDOWS	DirectorySearcher	ds	dsName
19W	WINDOWS	DomainDropDown	ddd	dddName
20W	WINDOWS	ErrorProvider	ep	epName
21W	WINDOWS	EventLog	el	elName
22W	WINDOWS	FileSystemWatcher	fsw	fswName
23W	WINDOWS	Font	fnt	fntName
24W	WINDOWS	Form	frm	frmName
25W	WINDOWS	Graphics	gps	gpsName
26W	WINDOWS	GraphicsPath	gp	gpName
27W	WINDOWS	GroupBox	gb	gbName

28W	WINDOWS	HelpProvider	hp	hpName
29W	WINDOWS	HorizontalScrollBar	hsr	hsrName
30W	WINDOWS	Icon	ico	icoName
31W	WINDOWS	Image	img	imgName
32W	WINDOWS	ImageList	imgl	imglName
33W	WINDOWS	Label	lbl	lblName
34W	WINDOWS	LinkLabel	lnkl	lnklName
35W	WINDOWS	ListBox	lb	lbName
36W	WINDOWS	ListView	lv	lvName
37W	WINDOWS	ListViewItem	lvi	lviName
38W	WINDOWS	MainMenu	mmnu	mmnuName
39W	WINDOWS	MenuItem	mnui	mnuiName
40W	WINDOWS	MaskEditBox	meb	mebName
41W	WINDOWS	MessageQueue	msq	msqName
42W	WINDOWS	MetaFile	mf	mfName
43W	WINDOWS	MonthCalendar	mclr	mclrName
44W	WINDOWS	NotifyIcon	nico	nicoName
45W	WINDOWS	NumericUpDown	nud	nudName
46W	WINDOWS	PageSettings	pstg	pstgName
47W	WINDOWS	Panel	pnl	pnlName
48W	WINDOWS	Pen	pen	penname
49W	WINDOWS	PeformanceCounter	pfmc	pfmcName
50W	WINDOWS	PictureBox	picb	picbName
51W	WINDOWS	Point	pnt	pntName
52W	WINDOWS	PrintController	prtc	prtcName
53W	WINDOWS	PrintDocument	prtd	prtdName
54W	WINDOWS	PrinterSettings	prts	prtsName
55W	WINDOWS	Process	pcs	pcsName
56W	WINDOWS	Rectangle	rec	recName
57W	WINDOWS	Region	rgn	rgnName
58W	WINDOWS	ReportDocument	rptd	rptdName
59W	WINDOWS	ServiceController	srvc	srvcName
60W	WINDOWS	Size	sze	szeName
61W	WINDOWS	Timer	tmr	tmrName