



{ C }

使用 Windows API



Visual C# 2008

Microsoft .Net Framework 提供不少的 Class Library，但仍然可以使用 Windows API 處理 .Net Framework 所沒有提供的功能，例如：

- Windows Networking API。
- Microsoft Win32 Internet API。

在 Microsoft .Net Framework 中呼叫 Windows API (DLL)，需使用 `DllImport`，其命名空間為 `System.Runtime.InteropServices`，因此在 C# 中需加入：

```
using System.Runtime.InteropServices;
```

欲使用 Windows API，首先需瞭解 API 的宣告方式及其源自何者 DLL 檔案，此部份可查詢 MSDN 之說明。例如在 C# 中，可使用 `System.Windows.Forms.MessageBox` 類別顯示訊息，但亦可使用 Windows API 之 `MessageBox` 顯示訊息，`MessageBox` 源自 `user32.dll` 檔案，其宣告方式為：

## ✱ 【C/C++】

```
int MessageBox(int hwnd, LPCTSTR lpText,  
               LPCTSTR lpCaption, UINT nType);
```

## ✱ 【Visual Basic 6.0】

```
Declare Function MessageBox  
    Lib "user32" Alias "MessageBoxA"  
    (ByVal hwnd As Long, ByVal lpText As String,  
     ByVal lpCaption As String, ByVal wType As Long) As Long
```

接著將 Windows API 以 C# 的語法宣告，則使用 `DllImport` 呼叫其 DLL 檔案，並以 `static` 與 `extern` 修飾詞 (Modifier) 宣告，例如：

```
[DllImport("user32.dll")]  
public static extern int MessageBox(  
    int hWnd, string lpText, string lpCaption, uint wType);
```

此宣告可直接定義於所需使用的程式中，或者另外建立一類別檔案定義，例如：

```
using System;
using System.Runtime.InteropServices;

namespace WindowsAPI {
    public class Test {
        ...
        [DllImport("user32.dll")]
        public static extern int MessageBox(
            int hWnd, string lpText, string lpCaption, uint wType);
    }
}
```

最後使用 Windows API 的方式，則會因宣告方式的不同而有所差異，若是直接定義於所需使用的程式中時，則使用方式為：

```
int iResult = MessageBox(0, "...", "...", ...) ;
```

請參考範例 C-1。

```
using System;
using System.Runtime.InteropServices;
...
namespace WindowsAPI
{
    ...
    public const int MB_OKCANCEL = 1;
    public const int MB_ICONINFORMATION = 64;

    [DllImport("user32.dll")]
    public static extern int MessageBox(int hWnd, string lpText,
        string lpCaption, uint wType);

    private void btnShow_Click(object sender, System.EventArgs e)
    {
        int iResult = MessageBox(0, txtMsg.Text, txtCaption.Text,
```

```
        MB_OKCANCEL | MB_ICONINFORMATION) ;  
    }  
}
```

若是定義於外部的類別檔案時，則使用方式為：

```
int iResult = [類別名稱].MessageBox(0, "...", "...", ...) ;
```

請參考範例 C-2。

```
using System;  
using System.Runtime.InteropServices;  
...  
namespace WindowsAPI  
{  
    ...  
    private void btnShow_Click(object sender, System.EventArgs e)  
    {  
        int iResult = Test.MessageBox(0, txtMsg.Text, txtCaption.Text,  
            Test.MB_OKCANCEL | Test.MB_ICONINFORMATION) ;  
    }  
}
```

## ✳ 【參考資料】

[1] Microsoft Developer Network.