



えムナウのC#
プログラミングのページ

C++、C++ / CLI、C# 適材適所 BoF の紹介

とっちゃん
επιστημη
えムナウ

高萩 俊行 Microsoft MVP Windows - SDK
福田 文紀 Microsoft MVP Visual C++
児玉 宏之 Microsoft MVP Visual C#



おすすめストリートライブ・BoF

8/22 (水) 16:40 - 17:00 わんくま同盟
「わんくまストリートライブ ショート ショート」

8/23 (木) 12:20-12:35 えムナウのプログラミングのページ
「C++・C++/CLI・C# 適材適所のBoFの紹介」

8/23 (木) 13:30-13:45 わんくま同盟
「わんくまストリートライブ ダイナミック言語のおもしろさ」

8/23 (木) 15:25-16:40 わんくま同盟
.NET Framework 3.0 をこう使いたい

8/23 (木) 17:00-18:15 えムナウのプログラミングのページ
C++・C++/CLI・C# 適材適所



2007/09/21
詳説VSセットアップ
とっちゃん

2007/10/06
WPF 第2回
えムナウ

2007/10/20
WPF 第2回
えムナウ

MS大阪支店 にて

東京 新宿OSTにて

はじめに

- C++、C++ / CLI、C# は、兄弟のような存在ではあるが、各々が現役で置き換わるような言語ではない。
- επισημηの勉強会では、C++ / CLI を native / managed の仲介役として紹介した。
- だとしたら、C++、C++ / CLI、C# はそれぞれにいいところや得意とするところがあり、多言語開発環境の良さも、技術者の皆さんも分かっているんじゃないか。
- 皆さんも我々と一緒に欠点の指摘ではない C++、C++ / CLI、C# の適材適所を話し合っていきましょう。

DEMO



3人がこう思う使いどころ

とっちゃん	高萩 俊行	Microsoft MVP Windows - SDK
επιστημη	福田 文紀	Microsoft MVP Visual C++
えムナウ	児玉 宏之	Microsoft MVP Visual C#

C++:サンプル抜粋2

```
BOOL CMainFrame::OnDeviceChange (UINT  
    nEventType, DWORD_PTR dwData)  
{  
    BOOL bResult =  
        CFrameWnd::OnDeviceChange ( nEventType,  
            dwData );  
    // メディアが挿入されたことをビューに通知  
    return bResult;  
}
```

C++:サンプル抜粋4

```
CDevChangeHint hint(nEventType,  
                    pDBCv->dbcv_unitmask,  
                    pDBCv->dbcv_flags );  
  
CDocument* pDoc = GetActiveDocument();  
pDoc->UpdateAllViews( NULL,  
    WM_DEVICECHANGE, &hint );
```

C# : WinformからWPFをホスト

```
private ElementHost ctrlHost;
private MnowTchedBofWpf.TchedBofWpf
    tchedBofWpf;

private void UserControl1_Load
(object sender, EventArgs e)
{
    ctrlHost = new ElementHost();
    ctrlHost.Dock = DockStyle.Fill;
    panelWpf.Controls.Add(ctrlHost);
    tchedBofWpf = new
        MnowTchedBofWpf.TchedBofWpf();
    tchedBofWpf.InitializeComponent();
    ctrlHost.Child = tchedBofWpf;
}
```


C# : WPFでコードからImage操作

```
public void DoEmbossBitmapEffect()  
{  
    this.image1.BitmapEffect = new  
        EmbossBitmapEffect();  
}  
public void SetBitmapFileName  
    (string source)  
{  
    this.image1.Source = source;  
}
```

C++/CLI: Native を Managed でラップ

```
class Native { ... };
```

```
ref class Managed {
```

```
    Native* native;
```

```
public:
```

```
    Managed() { native = new Native(); }
```

```
    ~Managed() { this->!Managed(); }
```

```
    !Managed() { delete native; }
```

```
    ...
```

```
};
```

- クラスの場合、ポインタに限る
- new/delete をお忘れなく
- auto_ptr<Native> 使用不可
- インスタンスのコピー時に注意!

C++/CLI: Managed を Native でラップ

```
ref class Managed { ... };
```

```
class Native {
```

```
    gcroot<Managed^> managed;
```

```
public:
```

```
    Native() { managed = gcnew Managed(); }
```

```
    ~Native() { /* delete managed; は不要 */ }
```

```
    ...
```

```
};
```

- クラスの場合、直接メンバになれない
- **gcnew** をお忘れなく
- インスタンスのコピー時に注意!

#include <vcclr.h> しませう !

Microsoft[®]

Your potential. Our passion.[™]

© 2007 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.



**Prepare to make
your mark.**

