

My 名前空間の件について

とりこびと

本題に入る前に自己紹介（40分）

- とりこびと
- 山奥に住んでるといふ噂。
- ジェットトリコビトを操り、
- 日夜正義のために戦う！
- 今日はわんくま勉強会のお手伝い。

アジェンダ

1. My 機能の概要
2. My 名前空間の拡張
3. My 拡張のパッケージ化と配置
4. 考察

1. My 機能の概要

- アプリケーションおよびその実行時環境に関連する情報や、既定のオブジェクトのインスタンスへのアクセスを実現。
- Visual Basic 2005 から追加。

1. My 機能の概要

- My.Computer
- My.Application
- My.Forms
- My.Log
- My.Request
- My.Responce
- My.Resource
- My.Settings
- My.WebServices

1. My 機能の概要

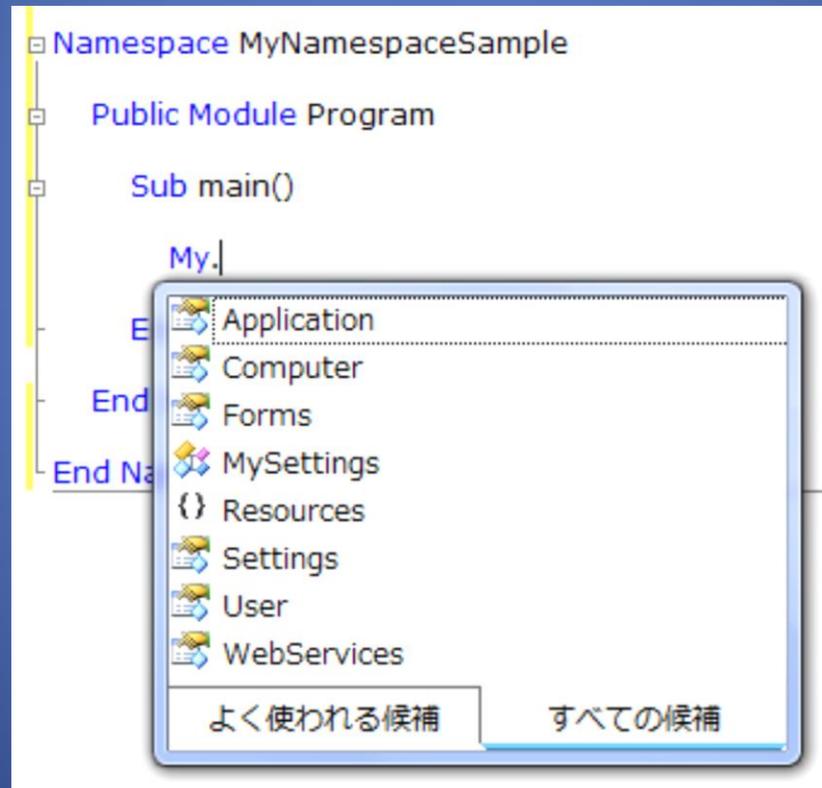
- Microsoft.VisualBasic.Dll に My 機能をサポートするクラスが定義されている。

My.Computer オブジェクト =
Microsoft.VisualBasic.Devices.Computer
の派生クラス。

My.Application オブジェクト =
Microsoft.VisualBasic.ApplicationServices
.WindowsFormsApplicationBase の派生クラス。

1. My 機能の概要

- IntelliSense によるサポート。



1. My 機能の概要

- コンパイラによる生成コード。

→ コンパイル定数を指定することにより生成を制御。

`__MYTYPE` と なかまたち。

1. My 機能の概要

プロジェクトの種類	_MYTYPE 値
クラス ライブラリ	"Windows"
コンソール アプリケーション	"Console"
Web	"Web"
Web コントロール ライブラリ	"WebControl"
Windows アプリケーション	"WindowsForms"
カスタムの Sub Main で開始される Windows アプリケーション	"WindowsFormsWithCustomSubMain"
Windows コントロール ライブラリ	"Windows"
Windows サービス	"Console"
空	"Empty"

1. My 機能の概要

<code>_MYTYPE</code>	<code>_MYAPPLICATIONTYPE</code>	<code>_MYCOMPUTERTYPE</code>
"Console"	"Console"	"Windows"
"Custom"	未定義	未定義
"Empty"	未定義	未定義
"Web"	未定義	"Web"
"WebControl"	未定義	"Web"
"Windows" または ""	"Windows"	"Windows"
"WindowsForms"	"WindowsForms"	"Windows"
"WindowsFormsWithCustom SubMain"	"Console"	"Windows"

未定義の条件付きコンパイル定数は FALSE

1. My 機能の概要

_MYTYPE	_MYFORMS	_MYUSERTYPE	_MYWEBSERVICES
"Console"	未定義	"Windows"	TRUE
"Custom"	未定義	未定義	未定義
"Empty"	未定義	未定義	未定義
"Web"	FALSE	"Web"	FALSE
"WebControl"	FALSE	"Web"	TRUE
"Windows" または ""	未定義	"Windows"	TRUE
"WindowsForms"	TRUE	"Windows"	TRUE
"WindowsFormsWithCustom SubMain"	TRUE	"Windows"	TRUE

未定義の条件付きコンパイル定数は FALSE

1. My 機能の概要

- ThreadSafeObjectProvider(Of T) クラス
 - My 名前空間のオブジェクトのスレッドセーフにするプロバイダクラス。
 - My.Project 名前空間。
 - Friend NotInheritable Class

2. My 名前空間の拡張

- 既存の My 名前空間メンバのカスタマイズ
→ Partial Class の利用。
- My 名前空間へのカスタム オブジェクトの追加
→ HideModuleName 属性 を使用して
IntelliSense が My 名前空間のメンバを
表示するときに、モジュール名を表示させない。

2. My 名前空間の拡張

```
Namespace My
```

```
  <HideModuleName()> _
```

```
  Module MyCustomExtensions
```

```
  End Module
```

```
End Namespace
```

2. My 名前空間の拡張

- My 名前空間に拡張機能開発の
デザインガイドライン
 - 拡張機能のロジックだけを含めるようにする。
 - プロジェクトの前提を最小にする。
 - 拡張コードを分離する。

3. My 拡張のパッケージ化と配置

- Visual Studio テンプレートを使用して、カスタムの My Namespace 拡張を簡単に配置できる方法が用意されている。
 - Visual Basic プロジェクト デザイナーの [マイ拡張] から、ユーザーがカスタム My 拡張を管理できる。
 - 特定のアセンブリへの参照がプロジェクトに追加されたときに、カスタム My 拡張を自動的に追加されるようにできる。

3. My 拡張のパッケージ化と配置

- CustomData ファイル。

```
<VBMyExtensionTemplate  
  ID="Sample.MyExtensions "  
  Version="1.0.0.0"  
  AssemblyFullName="Microsoft.VisualBasic"  
>
```

3. My 拡張のパッケージ化と配置

- .vstemplate ファイルへの
<CustomDataSignature> 要素の追加

```
<CustomDataSignature>  
    Microsoft.VisualBasic.MyExtension  
</CustomDataSignature>
```

4. 考察

- 本格的な開発で使える？
(Visual Basic プロジェクトの場合)
 - クラスライブラリ(Dll)での共通機能の提供の代替として使用できる？
 - 名前空間が ” My” 。
どのプロジェクトでも “My” 。
誰が作っても ” My” 。
 - あくまでコードファイルレベルでの拡張。
コンパイルは提供先で。

4. 考察

- C# からのMyの利用は？
 - Microsoft.VisualBasic.Dll の参照と、定義されたクラスへの直接のアクセス。
(MyApplication クラス, MyComputer クラス)
 - 互換性のないクラスも存在するため、注意が必要。

4. 考察

- 提供する側は？
 - クラス ライブラリ を作成して配布するほうが簡単…？

4. 考察

- 提供される側は？
 - テンプレートとしてインストールしてしまえばそれ以外の作業なくすべてのプロジェクトに任意に追加できる。

4. 考察

- 結局どうなの？
 - オリジナリティあふれる機能ですね。
 -
 -
 -
 -
 -

