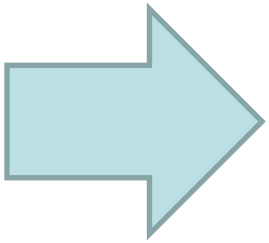
System.AddInを利用したアプリケーション拡張 - アドインの開発 -

Ｊ Ｚ ５（松江祐輔）@わんくま
http://katamari.jp
http://katamari.wankuma.com

2008/9/13

What's System.AddIn

- System.AddIn名前空間
  - Visual Studio Orcusから利用可能
  - アプリケーションに拡張機能を提供
  - なんかいろいろ特長が？



とりあえず使ってみるしか！

1. とりあえずアドインを作る
2. とりあえずアドインを使う
3. 特長/特徴をみてみる

用語について

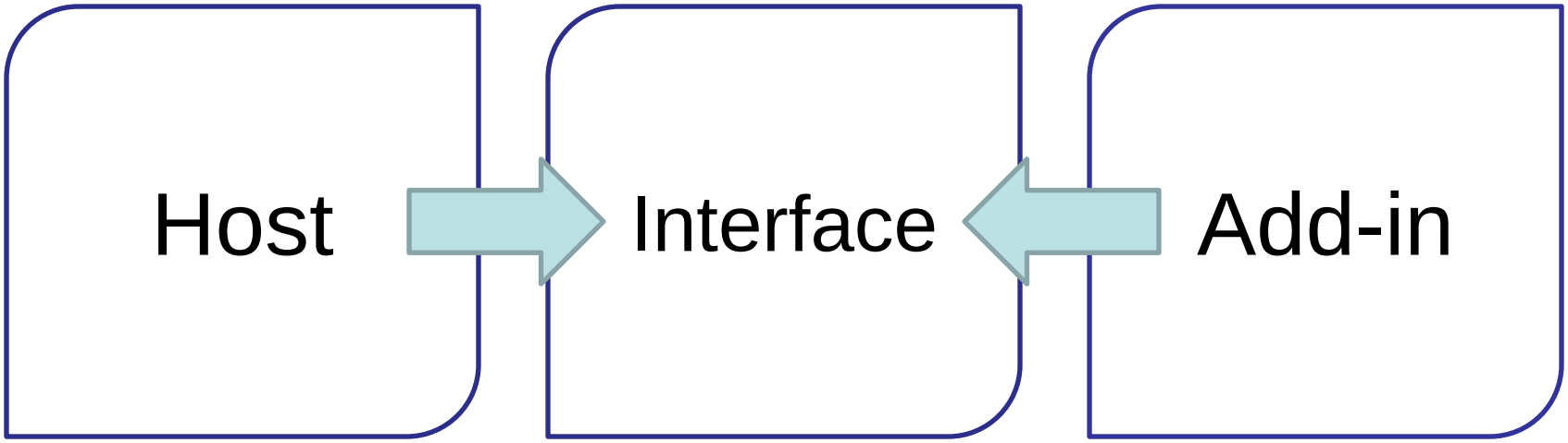
- アドイン: アプリケーションに追加される拡張機能
  - アドオン、プラグイン、スナップインとも
  - 動的に読み込まれる
- ホスト: アドインを使用する（拡張性のある）アプリケーション

サンプルケース

- Calcアドイン
  - 2個の引数を受け取り計算結果を返すメソッド
    - Calc(x As Integer, y As Integer) As Integer

これまでのアドイン

- Interfaceを作って参照・実装
- Reflectionで探索



コードで書いてみた(1/2)

- Interface

```
Public Interface ICalcAddIn
    Function Calc(ByVal x As Integer, _
                  ByVal y As Integer) As Integer
End Interface
```

- Add-in

```
Public Class Adder
    Implements Wankuma.ICalcAddIn
    Public Function Calc(ByVal x As Integer, _
                         ByVal y As Integer) As Integer _
                         Implements ICalcAddIn.Calc
        Return x + y
    End Function
End Class
```

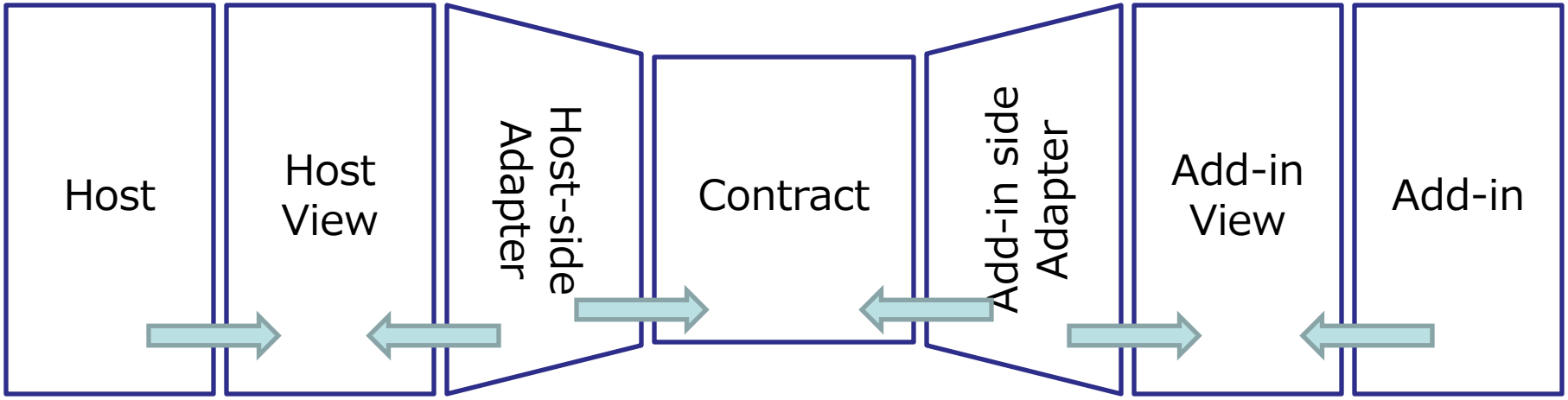
コードで書いてみた(2/2)

Host application

```
Imports System.IO
Imports System.Reflection
Module MainModule
    Sub Main()
        Dim files = Directory.GetFiles( _
            Path.GetDirectoryName(Assembly.GetExecutingAssembly.Location), "*.dll")
        For Each file In files
            For Each t In Assembly.LoadFrom(file).GetTypes
                If t.IsClass AndAlso _
                   t.IsPublic AndAlso _
                   Not t.IsAbstract AndAlso _
                   t.GetInterface("Wankuma.ICalcAddIn") IsNot Nothing Then
                    Dim addinAssembly = Assembly.LoadFrom(file)
                    Dim addin = DirectCast(addinAssembly.CreateInstance(t.FullName),  _
                        Wankuma.ICalcAddIn)
                    Console.WriteLine(t.FullName)
                    ' アドイン呼び出し
                    Console.WriteLine("Calc({0}, {1}) = {2}", 160, 2, addin.Calc(160, 2))
                End If
            Next
        Next
        Console.ReadLine()
    End Sub
End Module
```

System.AddInの場合

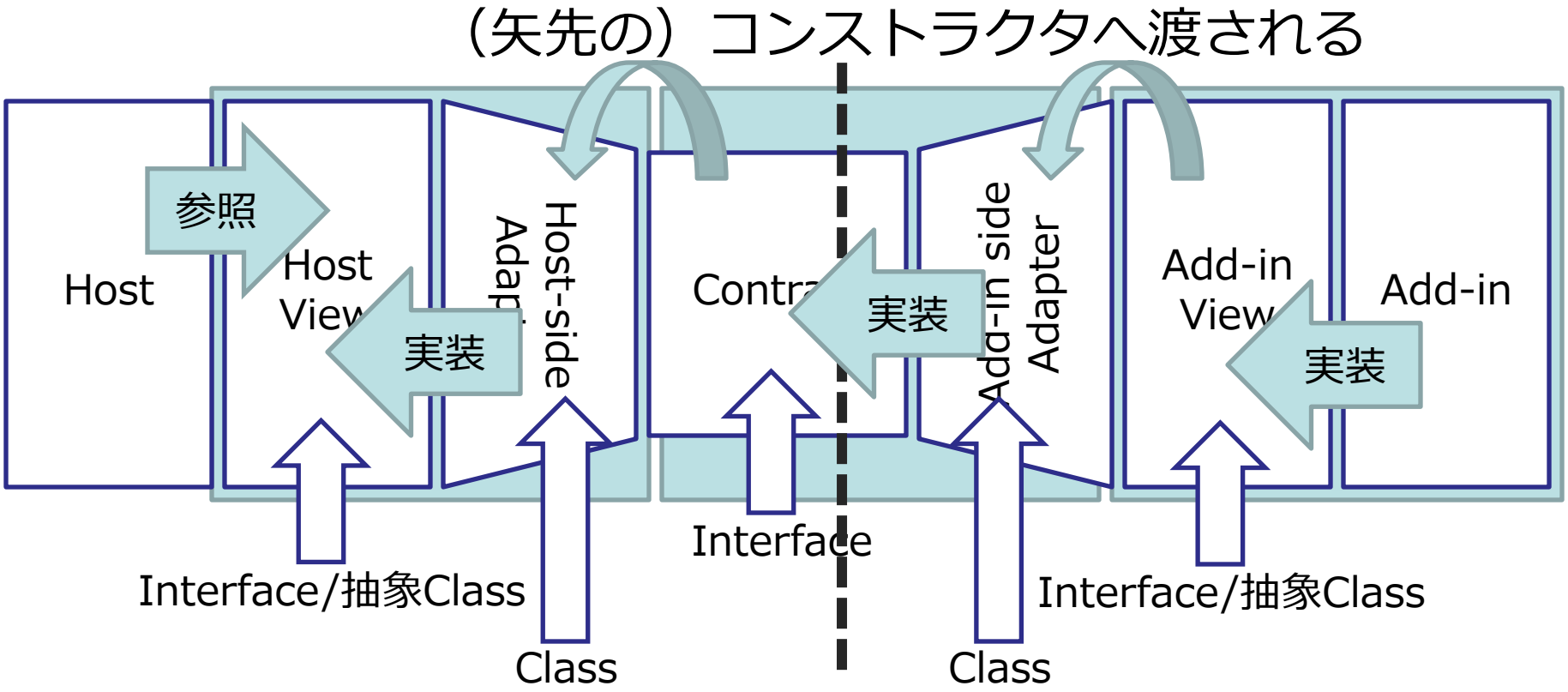
- アドインモデルとパイプライン



アドインパイプライン（通信パイプライン）：
ホスト-アドイン間通信のためのアドイン モデル

: 参照

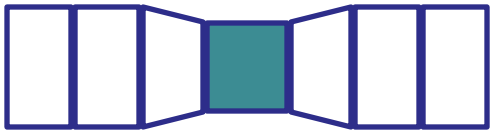
セグメントの関係



※7個のアセンブリ

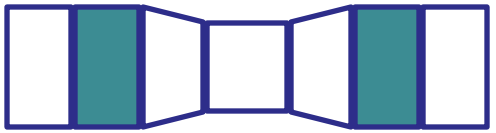
- HostとAdd-inは別AppDomain
- Contractは両Domainに読み込まれる

コントラクト



- ホストとアドイン間のプロトコルを定義
- IContractを実装したInterface
- AddInContract属性を付ける

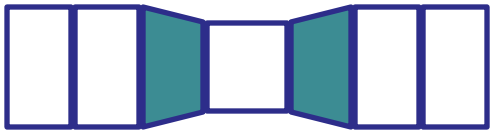
ビュー



- アドイン/ホストで
  使用する型・メソッドを
  持ったInterface または抽象クラス
- アドイン/ホストをコントラクトに依存させない
- ビュー自体には依存関係がない
- アドインビューはAddInBase属性を付ける

アダプター

- ビューとコントラクトの
  変換をおこなうクラス



- View To Contract
  - コンストラクタにViewが渡される
  - Contractを実装
- Contract To View
  - コンストラクタにContractが渡される
  - Viewを実装
- HostAddpter/AddInAdapter属性

実装（Contract）

```
Imports System.AddIn.Contract
Imports System.AddIn.Pipeline

<AddInContract()> _
Public Interface ICalcAddInContract
    Inherits IContract

    Function Calc(ByVal x As Integer, ByVal y As Integer)
As Integer

End Interface
```

- シリアル化可能な型（基本的なものに限る）とコントラクト、列挙値のみ受け渡し可能

実装（View）

```
Imports System.AddIn.Pipeline

<AddInBase()> _
Public Interface ICalcAddIn
    Function Calc(ByVal x As Integer, ByVal y As Integer)
As Integer

End Interface
```

- Add-in側はAddInBase属性
- 基本的にはHost/Add-inとも同じ

実装（Add-in Side Adapter）

```
Imports System.AddIn.Pipeline

<AddInAdapter()> _
Public Class ViewToContractAddInSideAdapter
    Inherits ContractBase
    Implements ICalcAddInContract

    Private _view As Wankuma.ICalcAddIn ' AddIn View

    Public Sub New(ByVal view As Wankuma.ICalcAddIn)
        MyBase.New()
        _view = view
    End Sub

    Public Function Calc(ByVal x As Integer, ByVal y As Integer) As
Integer Implements ICalcAddInContract.Calc
        Return _view.Calc(x, y)
    End Function
End Class
```

実装（Host Side Adapter）

```
Imports System.AddIn.Pipeline

<HostAdapterAttribute()> _
Public Class ContractToViewHostSideAdapter
    Implements Wankuma.ICalcAddIn ' HostView

    Private _contract As ICalcAddInContract
    Private _handle As ContractHandle

    Public Sub New(ByVal contract As ICalcAddInContract)
        MyBase.New()
        _contract = contract
        _handle = New ContractHandle(contract)
    End Sub

    Public Function Caluclate(ByVal x As Integer, ByVal y As Integer) As
Integer Implements ICalcAddIn.Calc
        Return _contract.Calc(x, y)
    End Function
End Class
```

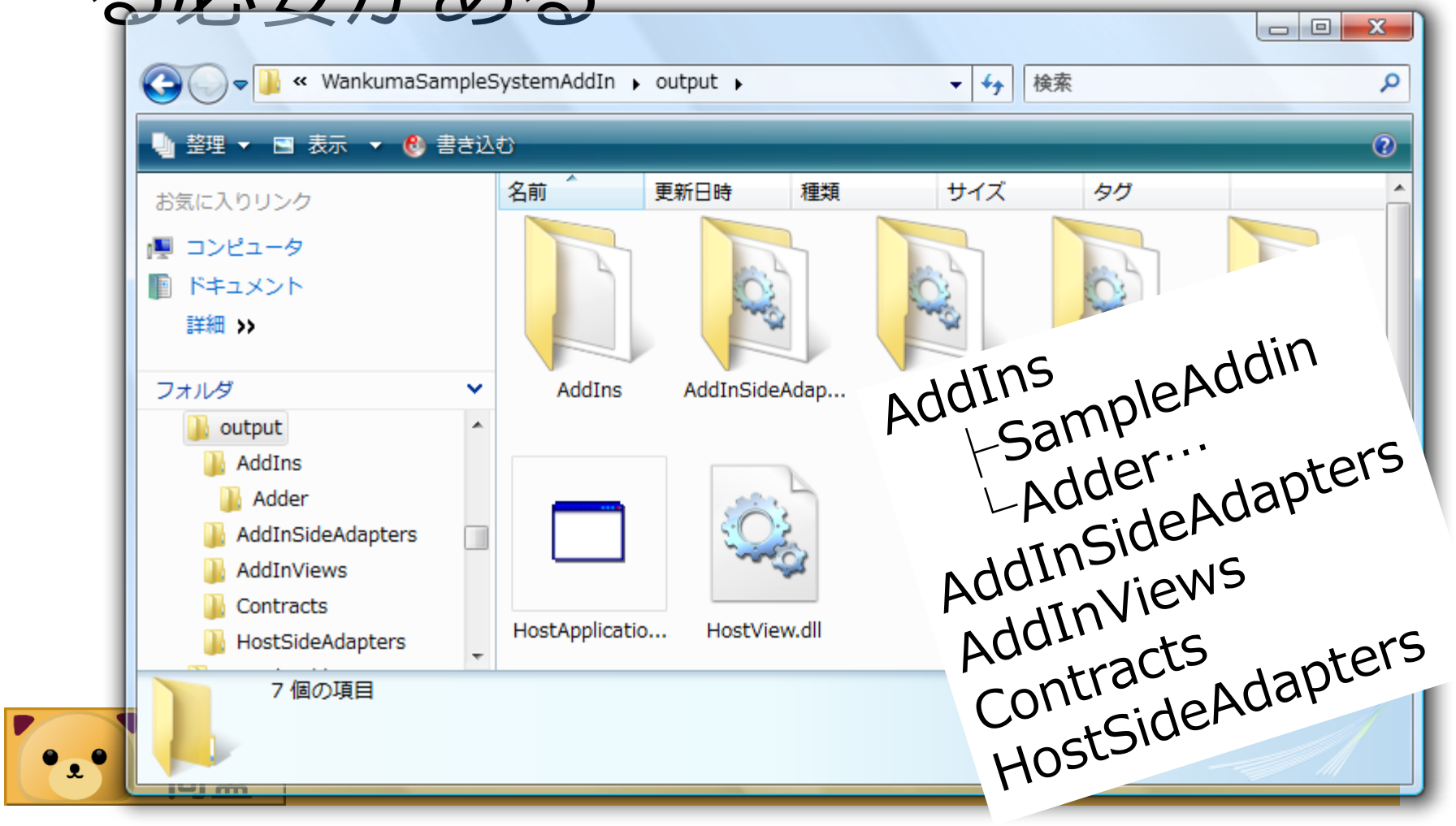
実装（Add-in）

```
Imports System.AddIn

<AddIn("Adder AddIn", Version:="1.0.0.0")> _
Public Class AdderAddIn
    Implements Wankuma.ICalcAddIn ' Add-in View

    Public Function Calc(ByVal x As Integer, _
                         ByVal y As Integer) As Integer _
        Implements Wankuma.ICalcAddIn.Calc

        Return x + y
    End Function
End Class
```

- Name（必須）, Description, Publisher, Versionの記述が可能

配置

- 決められたディレクトリ構成にする必要がある



アドインを使う（ホストの作成）

- アドイン探索
  - アドインとパイプラインセグメントの登録
  - アドインの検出
- アドインのアクティブ化

アドイン探索

```
' (Imports System.AddIn.Hosting)

Dim addInRoot = Environment.CurrentDirectory

' パイプラインセグメントキャッシュの更新
Dim warnings = AddInStore.Update(addInRoot)
For Each warning In warnings
    Console.WriteLine(warning)
Next

' アドインの検出
Dim tokens = AddInStore.FindAddIns(GetType(ICalcAddIn),
addInRoot)
```

- AddInStore Class: パイプラインとアドイン情報を格納・探索

アドインのアクティブ化

```
' ひとつめのアドイン選択
Dim token = tokens.First ' AddInToken Class

' アドインのアクティブ化
Dim calc = token.Activate(Of
ICalcAddIn)(AddInSecurityLevel.Internet)

' アドイン実行
Console.WriteLine(token.Name)
Console.WriteLine("Calc({0}, {1}) = {2}", 160, 2,
calc.Calc(160, 2))
```

- AddInToken Class:
  アクティブにできる
  アドインを表す

特長とか特徴とか

- 上位/下位互換性の確保
  - Adapterでバージョン間の差異を埋めることによってView, Contractのバージョンが変化してもHost/Add-inの変更なしで運用が可能
- 分離レベルと外部プロセス
  - HostとAdd-inが同じAppDomain
  - 複数のAdd-inが同じAppDomain
  - 各Add-inが各AppDomain
  - 外部プロセスのひとつの/別個のAppDomain

応用とか発展とか

- WPFのアドイン
  - アドイン自体がUI/UIを返す
  - INativeHandleContract型
- Pipeline Builder
  - CTP March-2008
  - ContractからView, Adapterの自動生成

参考Webサイト

- アドインおよび拡張機能
  - http://msdn.microsoft.com/ja-jp/library/bb384241.aspx
- アプリケーションの機能拡張
  - http://msdn.microsoft.com/ja-jp/library/bb909809.aspx
- CLR 徹底解剖 .NETアプリケーションの拡張性
  - http://msdn.microsoft.com/ja-jp/magazine/cc163476.aspx
  - http://msdn.microsoft.com/ja-jp/magazine/cc163460.aspx
  - http://msdn.microsoft.com/ja-jp/magazine/cc700355.aspx
- CLR Add-In Team Blog
  - http://blogs.msdn.com/clraddins/default.aspx
  - http://blogs.msdn.com/clraddins/pages/info.aspx
- System.AddIn Tools and Samples
  - http://codeplex.com/clraddins