

魅せます!

進化を続ける ASP.NET 2.0 の新機能

より使いやすく、より作りやすく、
よりパワフルな技術

山田 祥寛

YAMADA, Yoshihiro

<http://www.wings.msn.to/>

特集

3

Level

1 2 3 4 5

Technology Tools

Visual Basic

Visual C#

Visual C++

SQL Server

Oracle

Access 2002

ASP.NET

Other:

Visual Studio 2005 ベータ2

SQL Server 2005 ベータ2

IIS 6.0

ASP.NET 2.0

Samples

はじめに

1.xでも十分に魅力的なアーキテクチャを提供してきたASP.NETですが、2005年後半に登場予定の新バージョン2.0では、さらに飛躍的な進化を遂げます。

本稿では、正式版が待ちきれない、という皆さんのために、先日2005年4月にリリースされたVisual Studio 2005 β2 日本語版をベースに、ASP.NET 2.0の主要な新機能を取り上げて紹介することにします。

マスターページでデザインを共通に

企業ロゴを含むヘッダ部分、コピーライトを含むフッタ部分、メニュー部分など、サイトには多くの「共通部分」が存在します。ASP.NET 2.0で導入された「マスターページ」を利用することで、このような共通部分を容易に管理できるようになります。

▶ マスターページ的设计

マスターページとは、サイトに共通するベースの部分だけを定義した「断片的なWebフォーム」だと思えばよいでしょう。見かけも仕組みもほとんどWebフォームと同様で、イベント処理も記述できます。

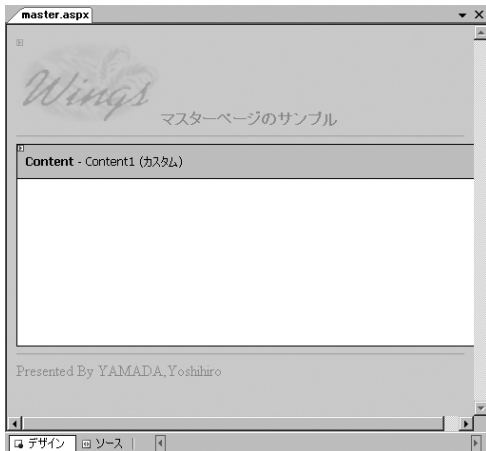
マスターページがWebフォームと異なるのは、以下の3点だけです。

- ・拡張子は「.master」
- ・@Pageの代わりに@Masterディレクティブを使用
- ・ひとつ以上のContentPlaceHolderコントロールが必須

@Masterディレクティブは@Pageディレクティブのサブセット的な位置づけであり、@Pageディレクティブとほとんど共通の属性を利用可能です。ただし、ContentType属性やBuffer属性、ResponseEncoding属性など、ページ全体に関わる属性は使用できません。

ContentPlaceHolderコントロールは、その名のとおりに、個別のコンテンツを配置するためのプレースホルダ（置き

図1: マスターページを適用したWeb フォーム



場所)です。マスターページでは、あらかじめひとつ以上のContentPlaceHolderコントロールを配置しておくことで、個別のWebフォームから異なるコンテンツをはめこむことができるというわけです。

▶ 個別のコンテンツを作成する

マスターページを利用するには、「新しい項目の追加」ダイアログで [マスターページを選択する] にチェックをつけておく必要があります。次の「マスターページを選択してください」ダイアログで適用するマスターページを選択することで、マスターページへの関連付けが行なわれ、図1のような画面が表示されるはずですが。

マスターページで定義された領域はグレー表示され、編集することはできません。個別のWebフォームから編集できるのは、あくまで「Content」で囲まれたブロックのみです。Contentブロックの中には、通常のWebフォーム同様、任意のサーバーコントロールやテキストなどを配置することができます。

▶ マスターページの高度な活用

Page.Masterプロパティを利用することで、自身が利用しているマスターページの内容を、個別のWebフォームから参照することも可能です。たとえば、以下はマスターページ内のImageコントロールを操作するコードです。

```
Dim img As Image = _
    CType(Page.Master.FindControl("Image1"), Image)
img.ImageUrl = _
    "http://www.wings.msn.to/image/wings.jpg"
```

リスト1: マスターページsite.masterで定義されたPageTitleプロパティ

```
Public Property PageTitle() As String
    Get
        Return Page.Header.Title
    End Get
    Set(ByVal value As String)
        Page.Header.Title = value
    End Set
End Property
```

マスターページにあらかじめプロパティを設定しておくことも可能です (リスト1)。マスターページのプロパティにアクセスする場合には、個別のWebフォーム上に@MasterTypeディレクティブを記述し、マスターページの型を厳密に指定しておく必要があります。

```
<%@ MasterType VirtualPath="~/site.master" %>
```

これによって、以下のように、マスターページのプロパティに対してより直接的にアクセスできるようになります。

```
Master.PageTitle = "サンプルページ"
```



ユーザビリティなどという言葉を持ち出すまでもなく、ユーザーに対してわかりやすいナビゲーションを示すことは重要です。さまざまなサイトを見ていると、ツリー式のメニューやポップアップメニュー、パンくずリスト^[注1]を見かけることがあるかもしれません。

ASP.NET 2.0で導入された「ナビゲーションコントロール」を利用することで、このようなナビゲーションインターフェイスを容易に実装できるようになります。

▶ サイトマップ定義ファイルを作成

ナビゲーションコントロールを利用するには、まずサイトマップ定義ファイル (web.sitemap) を仮想ディレクトリの直下に用意しておく必要があります (リスト2)。

サイトマップ定義ファイルは、ASP.NET標準の設定ファイルであるmachine.configやweb.configと同様、XML形式のファイルで、ルート要素は<siteMap>要素です。

<siteMap>要素の配下に記述できるのは、基本的に<siteMapNode>要素だけです。<siteMapNode>要素ひとつでひとつのコンテンツ情報を表現することができます。もしも配下にサブメニューを追加したいと思ったら、<siteMapNode>要

注1) Topページから閲覧中のページまでのリンクを連鎖リスト型に記述したもの。童話「ヘンゼルとグレーテル」で、迷子にならないようにパンくずを落としながら歩いたという故事に由来。