

必ずできる!

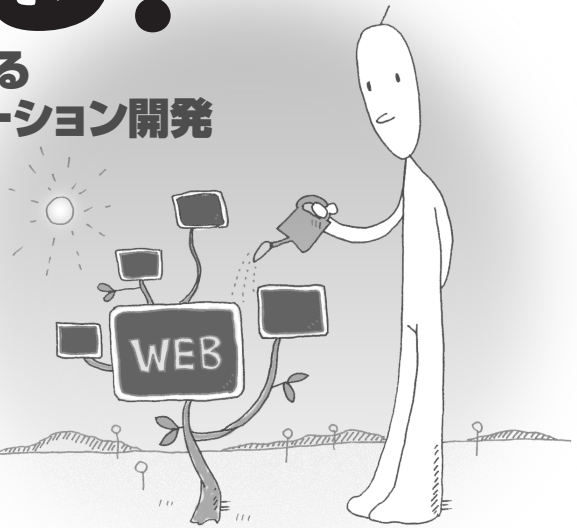
基礎固め ゼロからはじめる Webアプリケーション開発

ASP.NET

第10回

構成ファイルで アプリケーション管理を 効率化しよう

山田 祥寛 YAMADA, Yoshibiro
<http://www.wings.msn.to/>



Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
MSDE

Level



Samples



前回までスケジュール管理アプリケーションを例に、ASP.NET+Visual Studio .NETによるアプリケーションの構築方法を見てきました。ASP.NETの基本的な機能を利用した開発の流れがみなさんの中でもだんだんとイメージできてきたのではないのでしょうか。

さて、今回からは、趣をガラリと変えて、構成ファイルやGlobal.asaxなど、ASP.NETアプリケーションを裏から支える補助的な機能について見てゆきましょう。具体的に目に見える動きや機能を提供するサーバーコントロールなどに比べると地味に思える機能ですが、アプリケーションの保守性や堅牢性、パフォーマンスなどを意識した場合、これらの知識は必須ともいえます。



個々の設定項目を説明する前に、ま

ずはASP.NETにおける構成ファイルの関係を簡単に見ておきましょう。

ASP.NETの構成ファイルは、マシン共通の「マシン構成ファイル (machine.config)」とアプリケーションごとの設定を記述するための「アプリケーション構成ファイル (Web.config)」とに大別されます。

マシン構成ファイル (machine.config)

マシン共通の構成情報で、「<System Root>%Microsoft.NET%Framework %<バージョン番号>%CONFIG」フォルダに“machine.config”という名前前で配置されています。machine.configには、.NET Frameworkの挙動全般に関する設定のほか、すべてのASP.NETアプリケーションから採用されるデフォルトの設定が記述されています。構成ファイルのルートとも言うべきファイルで、通常、アプリケーション開発者はこのファイルを編集するべきではありません。

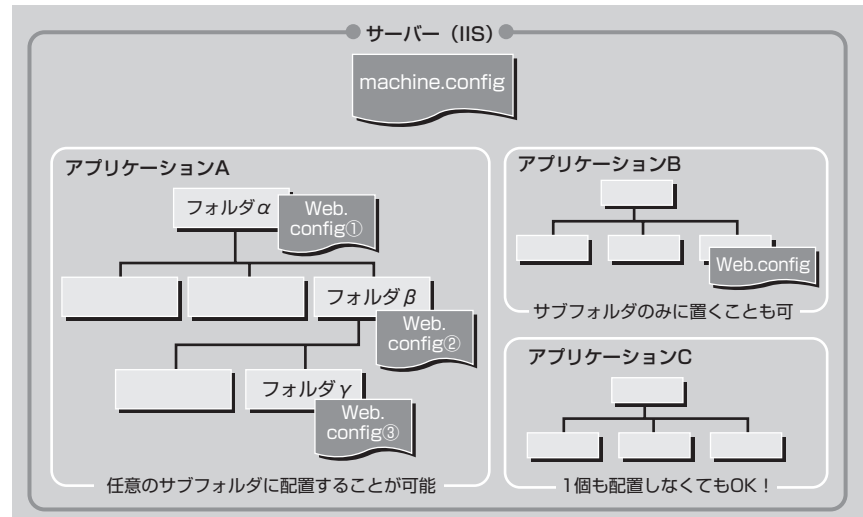
アプリケーション構成ファイル (Web.config)

アプリケーション個々の構成情報で、仮想ディレクトリ配下の任意のフォルダに“Web.config”という名前で配置します。machine.configが.NET Frameworkをインストールした段階で用意されているのに対して、Web.configはアプリケーション開発者が自ら配置する必要があります。また、Web.configは仮想ディレクトリ配下の各サブフォルダ単位に複数個、配置できます。

ASP.NETでは、マシン構成ファイルを頂点に、構成ファイルが多層構造を採ることで、複雑な構成を最低限の記述で効率的に記述できるのが特徴です。たとえば、図1のように、構成ファイルが配置されていたと仮定しましょう。

ここで注目してほしいのは、個々のWeb.configで「すべての設定要素を記述する必要はない」ということです。個々のWeb.configで記述されていない項目は、より上層のフォルダでもっとも近接しているWeb.configの内容を継承します。つまり、図1のmachine.configとWeb.configにおいて、仮に表1のように設定している場合には、各フォルダの設定は表2のように適用されます。該当する(サブ)フォルダにおいて、Web.configが省略された場合には、すべての設定項目を上位のWeb.config

図1：マシン構成ファイルとアプリケーション構成ファイル



(machine.config) から引き継ぎます。

なお、設定項目によっては、

- machine.configでしか設定できないもの (MachineOnly)
- machine.configまたは仮想ディレクトリ直下の設定ファイルでしか設定できないもの (MachineToApplication)
- どこでも設定可能なもの (Everywhere)

に分類されるので、注意してください。

さて、構成ファイルの基本的な考え方を理解したところで、構成ファイルの個々の設定について詳述してゆくにしましょう。もっとも、構成ファイルの仕様は膨大なので、ここではすべての項目を紹介することはできません。

本稿では、その中でも特に重要と思われる「セッション管理」「エラーページの設定」「HTTPハンドラクラス」について取り上げることにします。

多様なセッション 管理の手法

ページ間にまたがる情報を保持したいという場合、ASP.NETでは便利なセッションの仕組みを提供しています。セッション情報の利用方法については前号でもご紹介しましたが、ここではもう少し突っ込んで、セッション情報を保持する仕組みについて、簡単に見ておくことにしましょう。

図2はASP.NETがセッションを保持する標準的な方法を示しています。図2

表1：構成ファイルと設定の継承 (「-」は未設定)

| 設定ファイル | 項目A | 項目B | 項目C | 項目D |
|----------------|-----|-----|-----|-----|
| machine.config | A0 | B0 | C0 | D0 |
| Web.config① | A1 | B1 | - | D1 |
| Web.config② | A2 | - | - | - |
| Web.config③ | A3 | B3 | - | - |

表2：各フォルダに適用される設定値

| フォルダ | 項目A | 項目B | 項目C | 項目D |
|------|-----|-----|-----|-----|
| α | A1 | B1 | C0 | D1 |
| β | A2 | B1 | C0 | D1 |
| γ | A3 | B3 | C0 | D1 |