

デフォルトのコントロールでは
満足できないあなたに贈る
ユーザーコントロール作成の極意

特集

3

ユーザーコントロールで “ASP.NETらしい” レイアウト部品を作ろう

山田 祥寛

YAMADA, Yoshihiro
<http://www.wings.msn.to/>

Level

1 2 3 4 5

Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET

Other:

TreeView コントロール

Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、
<http://www.shoeisha.com/mag/windev/> からダウンロード可能です。



ASP.NETにおけるレイアウト部品化の技術は、大きくカスタムコントロールと（本稿で紹介する）ユーザーコントロールとに分類できます。

カスタムコントロールとは、ASP.NET上で利用可能なサーバーコントロールをイチから構築するための仕組みです。デフォルトで用意されているサーバーコントロール、またはそれらの組み合わせによっては実現できない（しにくい）機能の開発に利用します。既存機能にとらわれない柔軟性という意味では優れた仕組みであり、特に海外では多くの実用性あるカスタムコントロー

ルが無償／有償を問わず提供されています（表1）。しかし、カスタムコントロールでは、動作に必要な機能を最初から自分で実装しなければなりませんので、ちょっとした機能を実装するためにも大仰なコードが必要になります。

ASP.NETにはすでに多くの優れたサーバーコントロールが用意されているわけですから、これらの機能を組み合わせる、またはカスタマイズすることでまかなえるならば、カスタムコントロールに手を出す必要はありません。

ここで登場するのが、ユーザーコントロールです。

ユーザーコントロールとは、言うなれば「断片的なWebフォーム」です。Control派生クラスをイチから定義しな

表1：主要なカスタムコントロールの公開サイト

サイト名	URL
411 ASP.NET	http://www.411asp.net/home/webapps/
ASP.NET Resource Directory	http://www.123aspx.com/Directory.aspx?dir=205
Control Gallery	http://www.asp.net/Default.aspx?tabindex=2&tabid=31
HotScripts.com	http://www.hotscripts.com/ASP.NET/Scripts_and_Controls/
SharpToolbox	http://sharptoolbox.com/

リスト1: TreeViewコントロールを利用したサイトメニューの例 (MyMenu.aspx)

```

<%@ Page ContentType="text/html" Language="C#" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SqlClient" %>
<!--カスタムコントロールTreeViewを有効化する-->
<%@ Register TagPrefix = "ie"
    Namespace = "Microsoft.Web.UI.WebControls"
    Assembly = "Microsoft.Web.UI.WebControls" %>
<script runat="Server">
private void Page_Load(Object sender, EventArgs e){
// AppendNodeメソッドを呼び出し
// TreeViewコントロールにツリーノードを追加
this.AppendNode(0, menu.Nodes);
}

private Boolean AppendNode(int parent, TreeNodeCollection nodes){
// 配下に子ノードが存在するかどうかを判定するフラグ変数
Boolean flag = false;
using(
SqlConnection db = new SqlConnection("Data Source=(local);"
+ "User ID=sa;Password=sa;Persist Security Info=True;"
+ "Initial Catalog=dotnet")){
// sitemapテーブルから parent / category フィールドを
// キーに検索
// パラメータ parent で指定されたメニューIDを
// 親として持つメニューを抽出
SqlCommand cmd = new SqlCommand(
"SELECT id,title,url FROM sitemap "
+ "WHERE parent=@parent AND category=@category",db);
cmd.Parameters.Add("@parent", parent);
cmd.Parameters.Add("@category", Request.QueryString["category"]);
db.Open();
SqlDataReader reader=cmd.ExecuteReader();
// 抽出したメニュー情報を現在のノード配下に追加
// AppendNodeメソッドを再帰的に呼び出し、
// さらに子ノードがないかを確認
while(reader.Read()){

```

```

TreeNode node = new TreeNode();
node.Text = reader.GetString(1);
node.NavigateUrl = reader.GetString(2);
// AppendNodeメソッドがtrue/falseいずれであるかによって
// ノード型 (ノード型は<ie:TreeNodeType>要素で定義) を決定
// true (子ノードを持つ) の場合: Folder型
// false (子ノードがない) の場合: File型
if(AppendNode(reader.GetInt32(0), node.Nodes)){
node.Type = "Folder";
}else{
node.Type = "File";
}
nodes.Add(node);
flag = true;
}
db.Close();
}
// AppendNodeメソッドは、
// 子ノードが存在するかどうかをtrue/falseで返す
return flag;
}
</script>

<ie:TreeView id="menu" runat="Server"
SystemImagesPath="/webctrl_client/1_0/treeimages/">
<!--それぞれのノード名とアイコンイメージを定義。
ImageUrl属性で閉じたイメージのアイコンを、
ExpandedImageUrl属性で開いたイメージのアイコンを、
それぞれ指定する-->
<ie:TreeNodeType Type="Folder"
ImageUrl="/webctrl_client/1_0/images/folder.gif"
ExpandedImageUrl="/webctrl_client/1_0/images/folderopen.gif" />
<ie:TreeNodeType Type="File"
ImageUrl="/webctrl_client/1_0/images/html.gif" />
</ie:TreeView>

```

ければならないカスタムコントロールとは異なり、Webフォーム（「.aspx」ファイル）を作成するのと同じ要領で作成できる手軽さが特徴です。

もっとも、その手軽さゆえにどうにでも記述できてしまう“適当さ”は、長所であり短所であると言えるかもしれません。本当に、皆さんはASP.NETらしいユーザーコントロールを記述しているのでしょうか？

特に、古いASP 3.0で開発を行っていた方は、ユーザーコントロールを単なるインクルードファイルの代替として使用してしまう傾向があるようにも思われます。本稿でもう一度、“ASP.NETらしい”ユーザーコントロールの記法について見直してみましょう。



レガシーASPでの部品化技術

理想的なユーザーコントロールの書き方についてお話しする前に、まず古いASP 3.0（以降レガシーASP）でのレイアウト部品化について紹介しておく必要があります。レガシーASPでページレイアウトを部品化するには、大きく2つのアプローチがありました。

- SSI（サーバー側インクルード）
- Server.Executeメソッド

そして、これらのアプローチは、ASP.NETになっても健在です。