

特集

2

Windows Ajax!

でも



ASP.NET 2.0でAjax/Atlas

吉松 史彰
YOSHIMATSU, Fumiaki

softはGoogleに指摘されるまでXML HttpRequestオブジェクトの存在を忘れていたわけではないということだ。ASP.NET 2.0には、“軽量PostBack”と呼ばれる機能が含まれている (Microsoftが.NET Framework 2.0を公開したPDC 2003は2003年10月末に開催された)。当時はまだAjaxという言葉がなかったので、この機能はAjaxとは呼ばれていなかったが、いわゆるAjax的なWebアプリケーションの実装を手助けする機能がASP.NET 2.0には元から含まれていたということだ。

軽量PostBack機能

ASP.NET 2.0の軽量PostBack機能は、主としてデータコントロールで利用されている。

具体的には、ASP.NET 2.0の「DetailsView」「GridView」「TreeView」の各コントロールにこの機能が実装されている。

DetailsViewコントロールでは、「EnablePagingCallbacks」プロパティを「True」に設定すると、ページングの処理で画面全体が再描画される通常のPostBackの代わりに、DetailsViewだけを再描画する軽量PostBackが行なわれる。

GridViewコントロールでは、「EnableSortingAndPagingCallbacks」プロパティを「True」に設定すると、ページングと並べ替えで軽量PostBackが行なわれる。

TreeViewコントロールでは、「EnableClientScript」プロパティを「True」に設定すると、ツリーの展開と収納の処理で軽量PostBackが行なわれる。

AjaxとASP.NET 2.0

GoogleがGoogle Mapsを公開したのが2005年2月、同じ月のJesse James GarrettのブログからAjaxという言葉が生み出され、それからはもうあっという間にWeb開発の目玉機能に成り上がった。

もっとも、Ajax的機能を利用したWebアプリケーションの例としてよく取り上げられるGoogleのGmailやA9.comが公開されたのは2004年4月だし、それ以前からこの機能を効果的に利用したWebアプリケーションは存在していた。また、Ajaxというコトバが流行し始めたころ、Microsoftの中の人々がしきりに強調したのは、「Outlook Web Accessは、はるか昔からAjaxだった」ということだった。

何を言いたいのかという、Micro

Level



Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET
- Other:
 - ↓
 - Visual Studio 2005
 - ASP.NET 2.0
 - JavaScript
 - Atlas

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。

リスト1: DataGridの置き換え (Default.aspx)

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Ajax Sample 2</title>
<link rel="stylesheet" href="StyleSheet1.css">
<script src="http://maps.google.com/maps?file=api&v=1
&key=ABQIAAAAuFlI6kvQVigI6ATIIsLmBAxQFucwGYCvriCAXGV
vhhnciDdEedBR3e0leGNiIJG7-rMpY2V6dkwzKew"
type="text/javascript">
</script>
</head>
<body>

<script type="text/javascript">
<!--
var lon, lat;
//-->
</script>
<form id="form1" runat="server">
<div id="photoFrame">
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString=
"%$ ConnectionStrings:Ajax1ConnectionString %">
```

```
SelectCommand="SELECT * FROM [GeotaggedPhotos]">
</asp:SqlDataSource>
<asp:GridView ID="GridView1" runat="server"
DataSourceID="SqlDataSource1"
ShowHeader="False" PageSize="1" AllowPaging="True"
AutoGenerateColumns="False" BorderStyle="None"
OnRowDataBound="GridView1_RowDataBound">
<Columns>
<asp:ImageField DataImageUrlField="url" />
</Columns>
<PagerSettings Mode="NextPrevious"
PreviousPageText="#171;" NextPageText="#187;" />
<PagerStyle HorizontalAlign="Center" />
</asp:GridView>
</div>
</form>
<div id="mapFrame"></div>
<script type="text/javascript">
<!--
if (GBrowserIsCompatible()) {
var map = new GMap(document.getElementById("mapFrame"));
map.centerAndZoom(new GPoint(lon, lat), 4);
}
//-->
</script>
</body>
</html>
```

GridViewコントロールの軽量PostBackを解剖

前号では、最初ASP.NET 1.1のDataGridコントロールを用いて、通常のPostBackを利用した通常のWebアプリケーションを実装し、それからDataGridをHTMLのDIV要素などに置き換えた上でAjax的実装へ切り替えた。

ASP.NET 2.0のGridViewコントロールは、1.1のDataGridコントロールを置き換える存在なので、GridViewがAjax的軽量PostBackをサポートしているのなら、さしたる苦もなく前回のWebアプリケーションをAjax的実装に変えられるように思える。

そこで、GridViewを利用してAjax的実装にトライしてみることにする。

EnableSortingAndPagingCallbacksの動作

GridViewコントロールは、デフォルトではDataGridと大差ない動作をする。DataSourceコントロールとの連結によって従来必要だったデータを読み取ってDataBindするコードを記述しなくても動作するようになったなどの違いはあるが、GridViewコントロール自体の動作にはあまり違いはない。

2005年12月号で作成したWebForm1.aspxから、DataGrid

リスト2: リスト1のコードファイル (Default.aspx.cs)

```
(略)
public partial class Default : System.Web.UI.Page {
protected void GridView1_RowDataBound(object sender,
GridViewRowEventArgs e) {
if ((!IsCallback && (e.Row.RowType == DataControlRowType.DataRow)) {
DataRowView row = (DataRowView)e.Row.DataItem;
ClientScript.RegisterClientScriptBlock(this.GetType(), "latlon",
string.Format("lat={0};lon={1};", row["lat"], row["lon"]),
true);
}
}
}
```

コントロールをSqlDataSourceとGridViewの2つのコントロールで置き換えたものがリスト1である。DataSourceコントロールやImageFieldコントロールなどの新しい機能はあるものの、ほとんど違いはない。リスト2はコードビハインドであるが、データを読み取る処理をSqlDataSourceコントロールに任せているぶん、コードは簡潔になっている。これらのコードによって実行時に生成されるページのHTMLも、DataGridを利用した場合とGridViewを利用した場合でほとんど変わらない。

GridViewでEnableSortingAndPagingCallbacksをTrueに