

新 OS で ASP.NET 対応の Web サーバー構築と課題開発

林 文彬

The Construction of Web Server Which Corresponding to ASP.NET on New OS and Development of an Applied System

Bunhin Hayashi

1. はじめに

今年の4月9日にMicrosoft社によるWindowsXPサポートの終了に従って、Windowsの作業環境でパソコンを利用している方々はWindows7等のバージョンに更新せざるを得ない状況になっている。

当校(近畿職業能力開発大学校)でもすでに去年から導入したパソコンの環境がWindowsXPをWindows7に切り替えた。このOSの更新により、一部のWindowsバージョンに依存する作業システムが新しい環境で動かなくなった。

筆者も長年Microsoft社が開発したWeb developerという作業システムを使って、ASP.NET(C#)によるWebアプリケーションシステム構築に関する在職者訓練(能開セミナー)を担当してきた。今回のWindows環境の変化で、今まで蓄積してきたWebアプリケーションの構築技術は新しい環境で使えなくなるので、作業環境の再構築が行わなければならない。しかし、このような新しい環境構築技術を紹介する市販図書や技術文献等が見つからないため、去年から独自でASP.NETに対応するWebサーバー構築およびSQL Server2008とのデータ接続技術を取り組んできた。

本報告はこの新しいOSでASP.NET対応のWebサーバー構築と教材に使われる課題開発の成果を紹介する。

2. セミナーコース環境概要¹⁾

図1は当セミナーコース「Web-DBシステムを利用した生産支援システムの構築(ASP.NET編)」を実施

するシステムの動作環境である。ここで、図1のDBサーバーはSQL Server 2008、WebサーバーはWindows7のIIS、アプリケーションシステムを構築する環境はWeb Developer 2010、プログラムはASP.NET(C#)を使う。

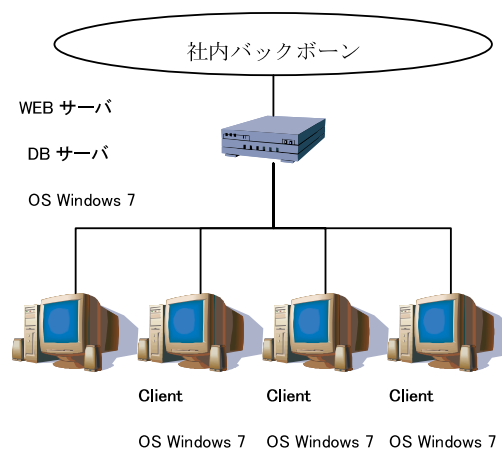


図1 システムの構成

3. セミナーコース環境の構築

今回はOSをWindows XPからWindows 7に切り替えたため、新しいOSでの実施環境構築は手探りでいった。以下はその経験により得られた環境構築法を紹介する。

3.1 ASP.NET対応のWebサーバー構築

Windows 7のIISでASP.NET対応のWebサーバー構築は図2のように「アプリケーション開発機能」を追加してインストールする必要がある。

さらに、ASP.NET 4.0をWebサーバーに登録しないとASP.NET対応のWebサーバーが実行できなくなり、エラーが発生する。それで、図3のファイルを実行することで、上記問題の解決ができた。

*生産電子情報システム技術科

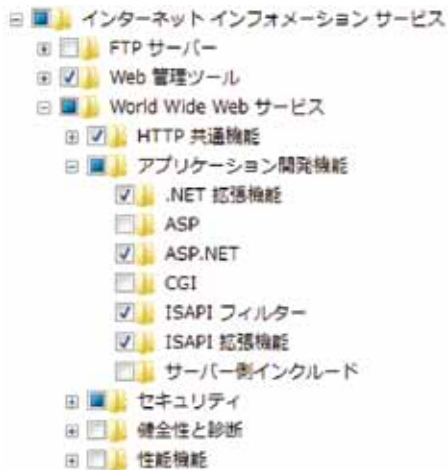


図2 IISによるWebサーバー設定

```
>C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis.exe -i
```

図3 ASP.NET 4.0の登録ファイル

3.2 SQL Server 2008 への接続設定

ASP.NET による SQL Server 2008 への接続には、Windows XP の場合が SQL Server 2008 で「サーバー名¥ASPNET」というユーザーを使うことに対し、Windows 7 の場合が「IIS APPPOOL¥DefaultAppPool」を使う。このため、SQL Server 2008 で「IIS APPPOOL¥DefaultAppPool S」ユーザーを新規作成する必要がある。また、そのユーザーにデータベースのアクセス権限を与える設定も行わなければならない。

3.3 Webサーバー起動できない対策

ASP.NET 対応の Web サーバーが起動できない場合は、下記のような原因が考えられる。

- 1) Internet Explorer のプロキシサーバーが外されていない。
- 2) 「コントロールパネル」での「既定の Web サイト」が「開始」を設定していない。
- 3) 「Apache」の Web サーバーと競合している。

その他、Web Developer 2010 による Web ページを開かない場合は、「コントロールパネル」既定の Web サイトのプロパティを開いて、「統合 Windows」にチェックを入れれば、解決ができる。

4. 課題の開発²⁾

3章の記述した方法で、Web-DB システムの構築環境ができるようになった。ここで、教材で使う応用例には、ダイアリの課題を構築した。この課題で使われているデータベースのテーブル(Schedule)は表1に示す。

表1 Schedule

列名	データ型	NULL 許容
Sid	int	notnull
Sdate	datetime	null
Stime	Char(5)	null
Subject	Varchar(100)	null

4.1 Web ページデザイン

ダイアリのページデザインは図4に示す。ページに「Calendar」、「ラベル」、「テキストボックス」、「ボタン」等のコントロールが配置された。

また、2つのデータを接続する「SqlDataSource」コントロール(sds, sds2)も配置された。そのうち、「sds2」で接続しているデータはカレンダーに用件の表示として使われ、「sds」は用件データをデータベースに書き込み・削除するために使われる。「sds」と「sds2」の SQL 文構成は表2に示す。



図4 Diaryのページデザイン

表2 「sds」と「sds2」のSQL文構成

	クエリ	SQL文構成
sds	SelectQuery	SELECT * FROM [Schedule]
	InsetQuery	INSERT INTO Schedule(sdate, stime, subject) VALUES (@sdate, @stime, @subject)
	DeleteQuery	DELETE FROM Schedule WHERE (sdate = @sdate)
sds2	SelectQuery	SELECT [stime], [subject] FROM [Schedule]

4.2 メソッドの構築

本課題ではDiaryの機能を実現するための日付選択、用件表示およびデータベースへのデータ更新・削除等のメソッドを作成した。表3は構築したメソッド一覧を示す。

表3 構築したメソッド一覧

メソッド名	目的
Cal_DayRender (...)	データベースのデータを読み出してカレンダーの日付のところに用件のデータを表示するメソッド
Cal_SelectionChanged (.....)	カレンダーの日付が選択されると、その日付をカレンダー下のラベルに表示するメソッド
sds_Inserting (...)	テキストボックスにある時間や用件等のデータをデータベースに挿入するメソッド
sds_Inserted (.....)	データをデータベースに挿入した後の処理を行うメソッド
sds_Deleting (.....)	選択された日付の用件を削除するメソッド
sds_Deleted (.....)	データを削除した後の処理を行うメソッド
InsertButton_Click (.....)	「用件追加」ボタンをクリックするときに起動するメソッド
DeleteButton_Click (.....)	「用件削除」ボタンをクリックするときに起動するメソッド

例えば、図4の「用件追加」というボタンをクリックすると、その関連メソッドの実行流れは図5に示す。

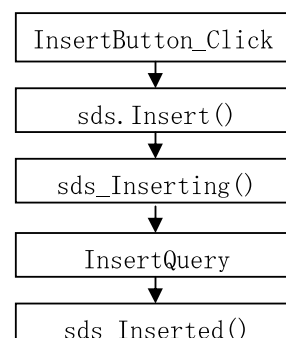


図5 データ挿入時にメソッドの流れ

図5の流れはまず、「用件追加」ボタンのクリックにより、InsertButton_Clickメソッドが起動されるので、メソッド内のsds.Insertメソッドも実行される。sds.InsertはOnInsertingメソッドを呼び出し、Insertingイベントが発生する。このイベントの実行でデータベースに挿入したいデータを表2のInsertQueryのパラメーター値に設定する。次にInsertQueryの実行でパラメーター値がデータベースに挿入される。

操作が完了した後、OnInsertedメソッドが呼び出されてInsertedイベントが発生する。このイベントではエラー等の後処理を行う。

5. 完成課題の実行

本課題の実行結果は図6に示す。まず、カレンダーの7月30日を選択し、その日付が自動的にカレンダー下のラベルに表示された。さらに、時間と用件を入力し、「用件追加」ボタンをクリックすると、カレンダーの7月30日に用件が表示された。

次に、SQL Server 2008のデータベースを開き、「2014-07-30」のデータを登録していることが確認できた(図7)。

また、用件を削除する場合も図8の示す通りに実行する。図8の例ではまず、7月21日を選択する。これによって、カレンダーの下方に「2014-07-21」という日付が表示された。次に、「用件削除」をクリックすると、カレンダーの7月21日の用件が消された(図9)。

最後に、SQL Server 2008 のデータベースを開き、
テーブル「Schedule」に「2014-07-21」のデータが

消えたことが確認できた(図 10 を参照)。



図 6 Diary の実行結果

結果	メッセージ	sid	sdate	stime	subject
1		1	2014-01-10 00:00:00.000	10:00	大阪情報社と打ち合わせ
2		5	2014-01-06 00:00:00.000	10:00	会議
3		6	2014-01-16 00:00:00.000	10:00	会議
4		7	2014-01-23 00:00:00.000	10:00	会議
5		10	2014-06-26 00:00:00.000	10:00	高校訪問
6		11	2014-07-21 00:00:00.000	10:00	科内会議
7		14	2014-07-30 00:00:00.000	09:00	企業訪問

図 7 「Schedule」テーブルの内容



図 8 用件削除

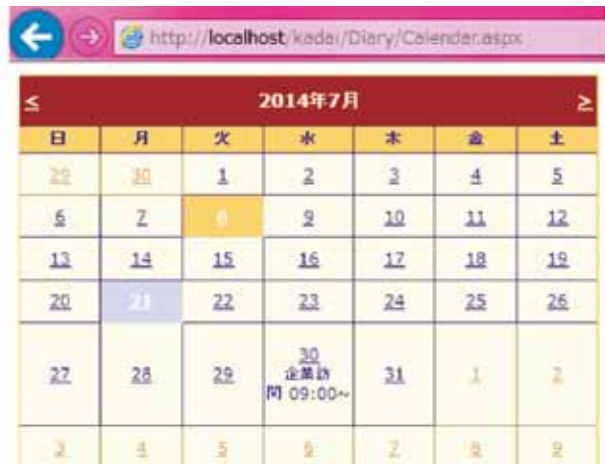


図 9 データ削除後

結果	メッセージ	sid	sdate	stime	subject
1		1	2014-01-10 00:00:00.000	10:00	大阪情報社と打ち合わせ
2		5	2014-01-06 00:00:00.000	10:00	会議
3		6	2014-01-16 00:00:00.000	10:00	会議
4		7	2014-01-23 00:00:00.000	10:00	会議
5		10	2014-06-26 00:00:00.000	10:00	高校訪問
6		14	2014-07-30 00:00:00.000	09:00	企業訪問

図 10 削除後のテーブルデータ

6. 結論

本報告は新しい OS での Web-DB システム環境構築を紹介した。ここで、その結果を下記のようにまとめた。

- 1) 市販図書や技術文献等を頼らず、新しい OS に対応する Web-DB システム実行環境の構築が成功し、教育訓練に活用することができた。
- 2) ダイアリのような応用課題構築により教材開発の充実が図られた。
- 3) 本教材は 2013 年度のセミナーに導入され、実施後のアンケート調査から、コース総合評価スコアは 91.7%であり、教材内容に高い満足度が示された。

参考文献：

- 1) Bunhin Hayashi, et al. : 「Development of Web-Oriented Parts Order System by MRP Method」, The Annual Conference on Computers and Industrial Engineering of 2005, Chinese Institute of Industrial Engineers, CIIE-027, PP. 1-6
- 2) 山田：「独習 ASP.NET2.0」、SHOEISHA