

【サーバーサイド×フレームワーク】 Vista時代の最新サーバーサイド開発事情 - ASP.NET AJAX & Visual Studio 2008概説 -

Windows Vista (.NET Framework 3.x) になり、サーバサイドの開発も大きく変わろうとしています。本セッションでは、「ユーザエクスペリエンス」をキーワードに、.NET Frameworkにおける最新フレームワークや開発ツール(ASP.NET AJAXやVisual Studio 2008など)を紹介、サーバサイド開発動向を概観します。



山田祥寛 (YAMADA, Yoshihiro)

yoshihiro@wings.msn.to

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

ユーザエクスペリエンスの重要性

■ 「機能要件を満たしていれば良い」は時代遅れ

- さまざまなサービスがWebアプリケーションとして提供されるのが当たり前
- 開発ツールやフレームワークの高機能化(例えば、VS2005 + ASP.NET)によって、十分な機能を持ったアプリケーションの開発が容易になった
- アプリケーションとしての「機能」だけでは差別化を図ることが困難
テレビ、掃除機、車など成熟した製品も単なる基本機能だけでは売れない

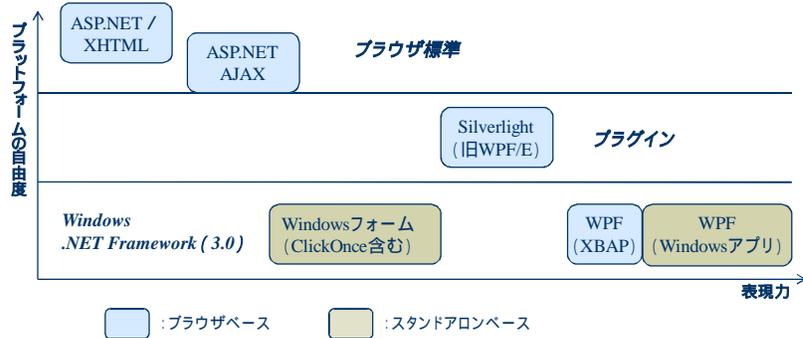
■ そこでユーザエクスペリエンス(=ユーザ体験)

- ユーザにより良い体験をもたらすこと
→ アプリケーションとしての使いやすさ、直観的なユーザインターフェイス、満足度
- ただ単に見栄えを派手にすることではない
→ ビジネス戦略からユーザ要件、アプリケーション構造、製品評価などアプリケーション開発のすべてのフェーズにおいて「ユーザ中心設計」(User Centered Design)を取り入れること
→ その結果としての「ユーザエクスペリエンス」
- 現時点で、ユーザ体験を定量的に分析する手法が確立されているとは言い難いが...
→ 直観的なユーザインターフェイスを構築するための基本技術は揃いつつある

マイクロソフトのプレゼンテーション技術(1) - プレゼンテーション関連技術の相関 -

■ 要件に応じて選択可能なプレゼンテーション技術

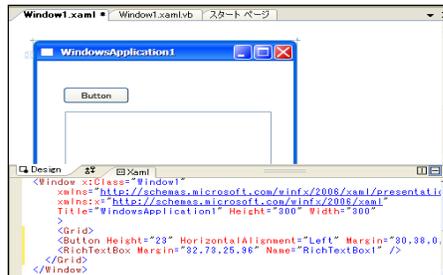
- 表現力とプラットフォームからの自由度に応じて選択
- それぞれの特性(得手/不得手)を理解しておく必要がある



マイクロソフトのプレゼンテーション技術(2) - Windows Presentation Foundation (WPF) -

■ リッチなユーザインターフェイスを共通したモデルで実現可能

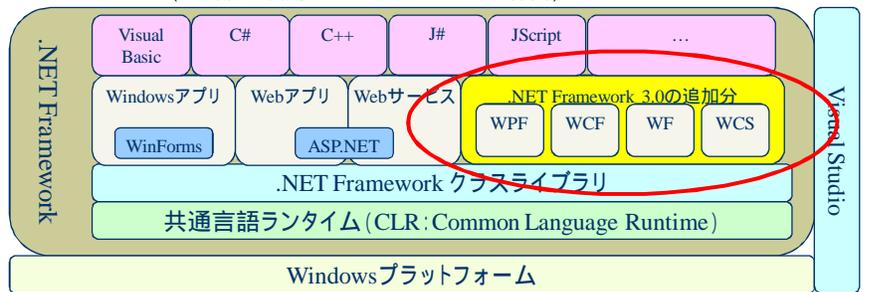
- .NET Framework 3.0で提供されるプレゼンテーション基盤 (Vista, XP, 2003 Server)
- 2D / 3Dグラフィック、静止画像 / 動画、ドキュメント等のコンテンツをリッチに表現
→ Direct3D、GPU (Graphic Processing Unit) によるレンダリング
- UI記述言語として、マークアップ言語XAML (eXtensible Application Markup Language)
 - 開発者向け Visual Studio 2005 Extension, 2008 β2、
 - デザイン向けツール Expression Blend を提供
 - Windows / Webアプリ双方で
プログラマ / デザイナとの分業
- Windows/Webアプリに関わらず、共通したプログラミングモデル
 - セルフホスト型アプリ (フル機能を利用可能)
 - Webブラウザホスト型アプリ (セキュリティサンドボックスによる一部機能の制限)
XBAP = XAMLブラウザアプリ



参考) .NET Framework 3.0

■ .NET Framework 3.0 = .NET Framework 2.0 + Wxx基盤

- Windows XP SP2 / Windows Server 2003 SP1 / (Windows Vista) 対応
- .NET Framework 3.0再配布パッケージ(ランゲージパックも同梱)
- Visual Studio 2005 extensions for .NET Framework 3.0 (WF対応)
- Visual Studio 2005 extensions for .NET Framework 3.0 (WCF & WPF), November 2006 CTP(正式版は次期Visual Studio 2008まで保留)



参考) Expression Studio

■ 「餅は餅屋」 - デザイナー向けに特化した専用ツール -

- .NET Frameworkデファクトスタンダードの開発環境Visual Studio 2005 / 2008
 - しかし、もともとがデザインを目的とした環境ではない
 - 高機能であり、デザイナーが利用するにはハードルも高い
- そこで、デザイナー向けに特化した専用のツールがExpressionシリーズ

製品名	概要
Expression Web	XHTML / CSSに対応したWebページデザインツール
Expression Blend	WPFを利用したリッチクライアント作成のためのツール
Expression Design	Expression Web / Blendで利用可能なベクタ画像はじめ、各種画像形式に対応したグラフィックデザインツール
Expression Media	各種ファイル形式に対応したコンテンツ管理ツール
Expression Studio	上記4製品のスイート製品

- Visual Studioとの連携も容易
 - Expression Studioで作成したプロジェクトをそのままVisual Studioで編集が可能

マイクロソフトのプレゼンテーション技術(3)

- Silverlight (開発コード"WPF/Everywhere")-

■ ブラウザプラグインで動作するFlash対抗技術「Silverlight」

- クロスプラットフォーム (Windows XP SP2 / Vista / Mac OS X) 対応
→ 対応ブラウザ Internet Explorer, Firefox, Safari 開発コード"WPF/E"
- WPFと共通のXAML言語による開発が可能
ただし、WPFとは異なるアーキテクチャで動作する技術なので要注意
- 開発言語は.NET対応言語から選択が可能 (Visual Basic / C# / IronRubyなど)
ただし、1.0はJavaScriptのみ
- PHPやJavaEE、ASP.NET AJAXなど既存技術との連携も容易
- 先日9月4日にいよいよ正式版1.0がリリース
- サポートをLinuxに拡大するべく、ノベルとの協力も発表
→ Linux環境での.NET環境である「Mono Project」をベースとした「Moonlight」も
- Expression Media Encoder 1.0もリリース (ビデオのインポートやエンコード、公開を支援)
- 関連ベンダとの協力関係を図る「Silverlight Partner Initiative」プログラムを立ち上げ

マイクロソフトのプレゼンテーション技術(4)

- ASP.NET AJAX (開発コード"Atlas" Framework)-

■ ASP.NET 2.0のAJAX拡張「ASP.NET AJAX」

- ASP.NET 2.0のAJAX対応拡張モジュールとして2007年1月に1.0がリリース
- プラットフォームを選ばない高い自由度
- ASP.NET + JavaScriptの標準的な知識で開発が可能

■ これらプレゼンテーション技術は排他的ではない

- プラットフォームの自由度、表現力などから、適材適所の使い分けが必要
- 複数技術の組み合わせも可能
Ex. Windowsフォーム + WPF (既存のWindowsアプリにWPFの表現力を)
ASP.NET AJAX + Silverlight (Silverlightをコーディングレスで呼び出し)
厳密にはASP.NET FuturesのXAML/Mediaコントロールで対応
- いずれにも.NETベースであることから組み合わせや使い分けも容易であるのが特長

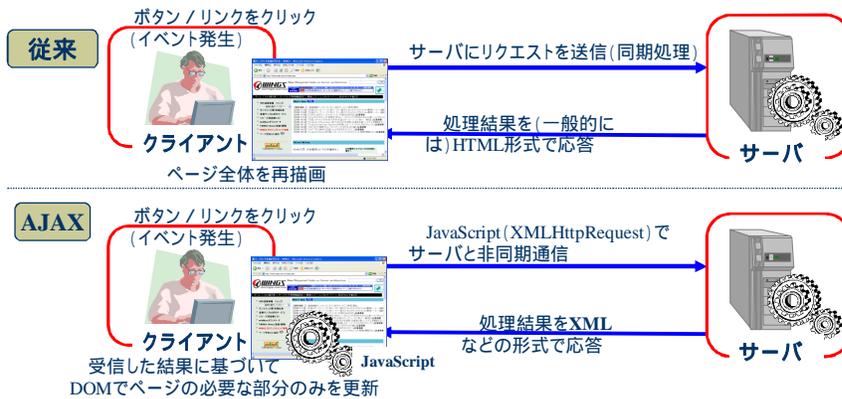
→ 本セッションではASP.NET AJAXを中心に解説

WPF 参考) E-7 「WPFでやってくるコントロール新時代」

AJAX (Asynchronous Javascript And Xml) とは？

■ AJAX (Asynchronous Javascript And Xml) とは...

- ▶ JavaScriptでサーバとの非同期 (Asynchronous) な通信を行い、その処理結果に基づいて、ページ上の必要なコンテンツだけを動的に更新する手法



AJAX技術の導入例と効果

■ AJAX技術の導入例

サイト名	URL	概要
Google Maps	http://maps.google.co.jp/	マウスによって表示位置を移動したり、拡大 / 縮小率を変更できる地図情報サービス
Windows Live	http://www.live.com/	RSS/Ajaxをフル活用したパーソナライズ可能なポータルページ (ニュースや天気、検索機能などを提供)
Amazon.com Diamond Search	http://www.amazon.com/gp/search/finder/?productGroupID=loose%5fdiamonds	ダイヤモンドの商品検索システム。形状やカラット数、価格などの条件に応じて商品を動的に表示

■ AJAX技術の導入効果

- ▶ ページ内の必要な箇所のみをリフレッシュ (チラツキの改善)
 - サーバ処理の間もクライアント側では操作を継続できる (作業を中断されない)
 - クライアント / サーバ間のトラフィック量を抑制
- ▶ 新たな機能の可能性 (メールの自動保存のような定期的なサーバとの通信処理など)
- ▶ 標準的なJavaScript (DOM) やXMLの知識のみで開発が可能



AJAX技術導入における課題

■ AJAX技術導入に際しての問題点

- クロスブラウザ問題 (ブラウザ間でのJavaScriptの挙動の違い)

```

if(window.XMLHttpRequest){ // Internet Explorer以外のブラウザ
  req = new XMLHttpRequest();
}else if(window.ActiveXObject) { // Internet Explorer
  try{
    req = new ActiveXObject("Msxml2.XMLHTTP"); // 6.0
  }catch(e){
    try{
      req = new ActiveXObject("Microsoft.XMLHTTP"); // 5.x以前
    }catch(e){
      return null;
    }
  }
}

```

- JavaScript固有の構文がサーバサイド開発者には馴染みにくい
- XML DOMによるデータ処理はコードが冗長になりがち

→ そこでAJAX対応フレームワーク

ASP.NET AJAX

<http://asp.net/ajax/>

■ ASP.NET 2.0上で動作する高性能なAJAX対応フレームワーク

- サーバ/クライアントを包括する総合的なAJAX対応フレームワーク
- Visual Studioとの緊密な連携 (フォームデザインからデバッグまで対応)
- サーバセントリック/クライアントセントリックな開発が選択可能
 - 従来のサーバコントロール型開発 (定型的な要件をコーディングレスで実現)
 - JavaScript + DOMを駆使した開発 (やや高度。しかし、より柔軟な要件にも対応)
- Control Toolkit: ASP.NET開発チームとコミュニティとの共同開発

ASP.NET AJAX Control Toolkit

ASP.NET AJAXベースのコントロール集&コントロール開発のための基盤を提供

ASP.NET AJAX

Microsoft AJAX Library

JavaScript上で動作する
AJAX対応ライブラリ

クライアントサイドフレームワーク

ASP.NET 2.0 AJAX Extensions

ASP.NET 2.0のAJAX拡張

ASP.NET 2.0

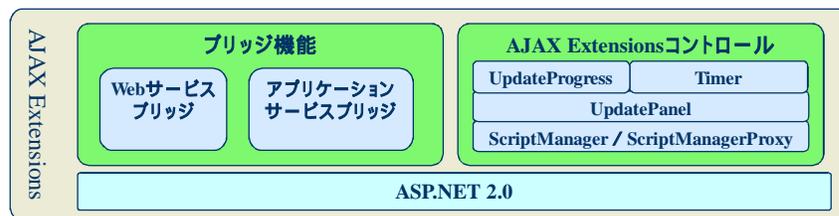
サーバサイドフレームワーク

ASP.NET 2.0 AJAX Extensions

- 概要と全体の構成 -

■ サーバセントリックな開発を支援するASP.NETベースのフレームワーク

- AJAX Extensionsコントロール
 - 従来型のサーバコントロールの配置とプロパティ設定を中心とした開発
 - 基本的にJavaScriptの知識は不要
- ブリッジ機能
 - クライアントサイドスクリプトからASP.NETの機能にアクセスするための橋渡し役
こちらはJavaScriptによるコーディングが必要
 - Webサービスブリッジ : XML Webサービス ('.asmx'ファイル)
 - アプリケーションサービスブリッジ: メンバシップフレームワーク (認証 / プロファイル)



ASP.NET 2.0 AJAX Extensions

- AJAX Extensionsコントロール(1) -

■ コーディングレスで定型的なAJAX機能を実装したい

- AJAX Extensions で提供されるサーバコントロール

コントロール名	概要
ScriptManager	AJAX機能の実行に必要なJavaScriptのコードを生成 / 管理 (例外処理やJavaScriptの国際化対応、外部JavaScript管理なども) ASP.NET AJAX機能全般を利用する際に必須となるコントロール
ScriptManagerProxy	マスタページ対応のScriptManagerコントロール
UpdatePanel	部分的に更新可能な領域を定義
UpdateProgress	非同期通信時に進捗状況を表示
Timer	一定間隔でポストバックを発生するタイマ機能を提供

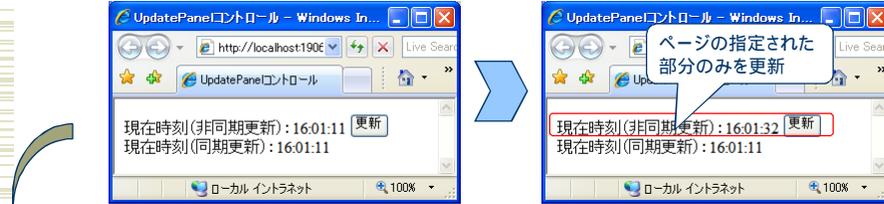
- 参考資料: 触ってみようASP.NET 2.0 AJAX Extensions (CodeZine)
<http://codezine.jp/a/article/aid/799.aspx>

ASP.NET 2.0 AJAX Extensions

- 参考) UpdatePanelコントロールの例 -

■ ページの部分的な更新を実現するUpdatePanelコントロール

➢ 非同期通信の結果を定義された領域だけに反映



実際のレイアウト&コードはこれだけ！(部分更新したいコンテンツをUpdatePanelで括る)

```

ScriptManager - manager
UpdatePanel - upanel
現在時刻(非同期更新): 16:01:11 [更新]
更新:
[Literal "ltrAsync" ]
現在時刻(同期更新): [Literal "ltrSync" ]

Protected Sub Page_Load(ByVal sender As Object, ~)
    ltrSync.Text = DateTime.Now.ToLongTimeString()
    ltrAsync.Text = DateTime.Now.ToLongTimeString()
End Sub
  
```

ASP.NET 2.0 AJAX Extensions

- AJAX Extensionsコントロール(2) -

■ AJAX Extensionsコントロールを利用する場合の注意点

- AJAX Extensionsコントロールは確かに便利(簡単)
 - でも、手軽さゆえに犠牲になっているところや気にすべきところも。
- 1. UpdatePanelコントロールはすべてのリクエストデータを送信
 - 部分更新に必要な情報のみを送信しているわけではない
 - 特に巨大なビューステート(ViewState)には要注意
- 2. UpdatePanelコントロールでは同期ポストバックと同様の処理が行われる
 - 不要な処理はScriptManager.IsAsyncPostBackプロパティで識別
 - 変更系イベント(テキストボックスの変更など)も非同期ポストバックで処理される可能性がある
- 3. Timerコントロールのポストバック間隔には要注意
 - Timer.Intervalプロパティ(通信間隔)には十分な間隔を指定
 - Intervalプロパティをエンドユーザの入力値に委ねるべきではない



AJAX(非同期)通信は、通常でも同期通信に比べて目に見えにくい。
 更に、AJAX Extensionsがますますその通信を隠ぺいしてしまう。
 → 通信の発生頻度とデータ量には十分に留意すべき

ASP.NET 2.0 AJAX Extensions

- ブリッジ機能 -

- クライアントサイドからサーバ側のサービスにシームレスにアクセス
 - ブリッジ機能の役割はプロキシクラス(代理クラス:JavaScript)を自動生成すること

The diagram illustrates the interaction between a Client and a Server. On the Client side, a user requests a page. The server (ASP.NET AJAX) generates a proxy class (JavaScript) from the service class. The client downloads this proxy class and uses it to access the service class on the server. The server returns the results in JSON format.

- JavaScriptのコーディングは必要であるが、非同期通信(XMLHttpRequestオブジェクト)を意識したコーディングは一切不要
- Webサービスブリッジ機能は汎用的なサービスへのアクセス機能を、アプリケーションサービスブリッジ機能は、ASP.NETの認証/プロファイル機能アクセスに特化した機能を、それぞれ提供(しくみはいずれも同様)

ASP.NET 2.0 AJAX Extensions

- 参考) ブリッジ機能による例 -

- ブリッジ機能を利用したAJAX版”Hello, World”
 - テキストボックスに入力した名前に対して、挨拶「こんにちは、 さん！」を応答
 - プロキシクラスを利用することで、サービスクラスに対してあたかもローカルのライブラリであるかのようにアクセスすることができる。

サービスクラス (ServiceBridge.asmx)

```
<WebService(Namespace:="http://tempuri.org/")> _
<WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)> _
<ScriptService()> _ → AJAX対応サービスであることを宣言
Public Class ServiceBridge
    Inherits System.Web.Services.WebService
    <WebMethod()> _
    Public Function GetHello(ByVal name As String) As String
        Return String.Format("こんにちは、{0}さん！", name)
    End Function
End Class
```

コールバック関数で通信完了時の処理を記述

クライアントからのアクセス (ServiceBridge.aspx)

```
function Button_Click(){
    ServiceBridge.GetHello(
        $get('txtName').value,
        function(result) {
            $set('result').innerHTML = result;
        });
    return false;
}
```

Microsoft AJAX Library

- 概要と全体の構成 -

■ クライアントセントリックな開発を支援するJavaScriptベースのフレームワーク

- ASP.NET環境だけでなく、PHPやJava EEなどの非.NET環境でも動作可能
- ブラウザ互換層: クロスブラウザ問題を吸収するためのレイヤ
(Internet Explorerをはじめ、Firefox、Safariなど主要なブラウザに対応)
ただし、使用している機能(やその組み合わせ)によっては未対応の場合もあり
- スクリプトコア: JavaScript上で名前空間やインターフェイス、継承/実装のような.NET Frameworkライクなオブジェクト指向構文を利用するための機能を提供
- 基本クラスライブラリ: 基本オブジェクト拡張からブリッジ機能への対応クラス、その他AJAX機能構築のための支援機能を提供

AJAX Library

基本クラスライブラリ

AJAX機能開発のための支援機能を提供

スクリプトコア

JavaScript上で.NETライクなオブジェクト指向構文を利用可能に

ブラウザ互換層

ブラウザ間の実装差を吸収し、ブラウザニュートラルなコーディングを可能に

Microsoft AJAX Library

- 基本クラスライブラリ -

■ JavaScriptの基本拡張からリッチなAJAX機能支援機能まで

名前空間名	機能
(Global)	JavaScript基本オブジェクト(ArrayやString、Errorなど)の拡張、スクリプトコアの実体であるTypeオブジェクト、ショートカット関数を提供 最低限、この名前空間の内容は理解しておきたいところ!
Sys	アプリケーションライフサイクルやクライアントコンポーネントの管理、デバッグ機能、カルチャ(国際化機能)などアプリケーション全体に関わる汎用的な機能を提供
Sys.Net	非同期通信に関わる基本機能を提供(ブリッジ機能へのアクセス手段であるプロキシクラスもここで提供)
Sys.Serialization	構造化データのJSON形式へのエンコード/デコード機能を提供
Sys.Services	アプリケーションサービスブリッジ機能へのアクセス手段を提供
Sys.WebForms	UpdatePanelコントロールによる非同期通信の前後で発生するイベントと、その管理手段を提供(PageRequestManagerクラスに注目!)
Sys.UI	クライアントコントロールのための基本クラスや、個別要素へのアクセス手段、また、その中で発生するイベントの管理を分担

Microsoft AJAX Library

- 参考) Sys.WebForms.PageRequestManagerオブジェクト -

■ PageRequestManagerオブジェクトによるUpdatePanelコントロールの細かな制御

- 部分更新に関わる処理はUpdatePanelコントロールに委ねつつ、最低限の処理をJavaScript側で記述する

Sys.Application.add_load(→ ページロード時の処理を登録

```
function(){
    var mng = Sys.WebForms.PageRequestManager.getInstance();
    mng.add_beginRequest(
        function(sender, args){
            args.get_postBackElement().disabled=true;
        });
};
```

非同期通信開始時にページ要素を使用不可

- PageRequestManagerオブジェクトではさまざまなイベントを公開

イベント名	発生タイミング
initializeRequest	リクエストの初期化時
beginRequest	リクエストの開始時 (通信中にページ内の状態を変更するなど)
endRequest	リクエストの終了時 (通信中に発生したエラーのハンドリングなど)
pageLoading	ページ反映中 (通信中に変更した状態をもとに戻すなど)
pageLoaded	ページへの反映を完了 (サーバ側から得た付随データに基づく処理)

Microsoft AJAX Library

- スクリプトコア(Typeオブジェクト) -

■ JavaScript上に.NET Frameworkライクなオブジェクト指向構文を導入

- JavaScriptはプロトタイプベースのオブジェクト指向構文を提供
 - Visual BasicやC#, Javaなどに慣れた開発者には馴染みにくい
 - 名前空間やインターフェイス、継承/実装などの構文には未対応
- スクリプトコア(Typeオブジェクト)によるオブジェクト指向構文の補強

```
Type.registerNamespace("MyAjax"); → 名前空間の定義

MyAjax.SuperMember = function(firstName, lastName, email) {
    MyAjax.SuperMember.initializeBase(this, [firstName, lastName]);
    this._email = email; ↑ 基底クラスのコンストラクタを呼び出し
}

MyAjax.SuperMember.prototype = {
    ...中略...
} ↓ SuperMemberクラスの登録 ↓ Memberオブジェクトの継承
MyAjax.SuperMember.registerClass(MyAjax.SuperMember, MyAjax.Member);
```

- 参考: Ajax時代のJavaScriptプログラミング再入門
<http://www.atmarkit.co.jp/fdotnet/ajaxjs/index/index.html>

ASP.NET AJAX Control Toolkit

<http://www.asp.net/ajax/control-toolkit/live/>

- ASP.NET AJAXベースのコントロール集 + コントロール開発基盤
 - AJAX対応コントロールを構築するための開発基盤
 - ASP.NET AJAXを基盤に構築されたサーバコントロール集
 - 2007年9月現在で35のAJAX対応サーバコントロールを提供
 - .NET Framework 2.0 / 3.5対応のパッケージを提供

Control Toolkit

AJAX対応コントロール

ASP.NET AJAXとコントロール開発基盤によって作成されたコントロール群 (2007年9月時点で35種類)

入力支援系コントロール

検証支援系コントロール

リスト&パネル系コントロール

視覚効果系コントロール

その他コントロール

コントロール開発基盤

AJAX対応コントロール開発の基本となる基本機能

ExtenderControlBaseクラス
(サーバサイド)

BehaviorBaseオブジェクト
(JavaScript)

その他ユーティリティ
クラス / オブジェクト

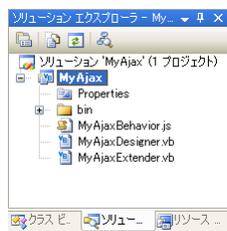
ASP.NET AJAX Control
Project (Visual Studio)

ASP.NET AJAX Control Toolkit

- コントロール開発基盤 -

- AJAX対応コントロール開発のための基盤機能を提供
 - Visual Studioのプロジェクトテンプレート「ASP.NET AJAX Control Project」を提供
 - コントロール開発に必要なファイルを自動生成
 - AJAX対応コントロール (Extenderコントロール) の開発に必要なファイル

ファイル名	概要
XxxxxExtender.vb	エクステンダクラス (サーバサイドに公開するプロパティやイベント、コントロール内で利用するリソース等を宣言するクラス)
XxxxxBehavior.js	Behaviorオブジェクト (クライアントサイドの挙動を規定するオブジェクト。AJAX対応コントロールでは中核となるオブジェクト)
XxxxxDesigner.vb	デザイナークラス (フォームデザイナー上の挙動や表示を決定するためのクラス。多くの場合はデフォルトのまま可能)



- Extenderコントロール: 標準 (非AJAX対応) コントロールに対して、特定の機能を追加 (拡張) するためのコントロール。それ単体では動作しない

ASP.NET AJAX Control Toolkit

- 参考) Control Toolkitによるコントロール開発(1) -

■ エクステンダクラスはサーバ側の挙動とコントロール全体のリソースを管理

```

<Designer(GetType(MyAjaxDesigner))> _ ← デザイナクラスの宣言
<ClientScriptResource("MyAjax.MyAjaxBehavior", "MyAjax.MyAjaxBehavior.js")> _
<TargetControlType(GetType(Button))> _ ← 対象コントロールの宣言 ↑ リソースの宣言
Public Class MyHelloExtender
    Inherits ExtenderControlBase ← Extenderコントロールの基本機能を提供

    <ExtenderControlProperty()> _ ← クライアント側に公開するプロパティであることを宣言
    <DefaultValue("")> _
    Public Property DialogText() As String
    Get
        Return GetPropertyValue("DialogText", "")
    End Get
    Set(ByVal value As String)
        SetPropertyValue("DialogText", value)
    End Set
End Property
End Class

```

ほぼ通常のプロパティ宣言と同様
(GetPropertyValue / SetPropertyValueは
ExtenderControlBaseクラスで提供されたもの)

ASP.NET AJAX Control Toolkit

- 参考) Control Toolkitによるコントロール開発(2) -

■ Behaviorオブジェクトはクライアント側の挙動を定義(コントロール開発の中核)

```

Type.registerNamespace('MyAjax');

MyAjax.MyHelloBehavior = function(element) {
    MyAjax.MyHelloBehavior.initializeBase(this, [element]);
    this._DialogTextValue = null; ← プロパティ値を格納するためのプライベート変数を宣言
}
MyAjax.MyHelloBehavior.prototype = {
    initialize : function() {
        MyAjax.MyHelloBehavior.callBaseMethod(this, 'initialize');
        SaddleHandler(this.get_element(),
            "click",
            Function.createDelegate(this, this._onClick)
        );
    },
    ...中略...
    get_DialogText : function() {
        return this._DialogTextValue;
    },
    set_DialogText : function(value) {
        ...中略...
    }
}
MyAjax.MyHelloBehavior.registerClass('MyAjax.MyHelloBehavior', AjaxControlToolkit.BehaviorBase);

```

初期化メソッド内で、ターゲット要素をクリックしたときの挙動を関連付け(_onClickイベントハンドラについては割愛)

ほぼ通常のプロパティ宣言と同様
サーバ側で宣言されたプロパティ名と対応するアクセサメソッドを設定

↓ BehaviorBaseオブジェクトの基本機能を提供

ASP.NET AJAX Control Toolkit - AJAX対応コントロール(1) -

■ ASP.NET AJAXベースで構築されたコントロール集

- ASP.NET AJAXを基盤に構築されたサーバコントロール集
→ 2007年9月現在で約35のサーバコントロールを提供
- 利用ドキュメント(英語)も充実(サイト上で動作デモ環境も提供)
- ただし、厳密にはAJAX機能を持たず、単なるJavaScriptライブラリをサーバコントロール化したにすぎないものも多いので注意。
Ex.) TabsコントロールはJavaScript的にページの切り替えを行っているにすぎない(→ すべてのタブの内容が一気にロードされる)
- 「A set of *sample* ASP.NET AJAX controls and source」(ReadMe.txtより)
サンプルとはいえ、十分に利用できるコントロールばかりであるが、正式なサポートはない点、複合的なコントロールの利用では競合もあろう点には要注意(要は、通常のオープンソースソフトウェアと同等)
- 日本語資料: 「ASP.NET AJAX Control Toolkitを使ってみよう」
<http://www.web-deli.com/tutorial.aspx?id=ASP-TTR-00058>
Control Toolkitの使い方から主要なプロパティ情報までを紹介するTIPS集

ASP.NET AJAX Control Toolkit - AJAX対応コントロール(2) -

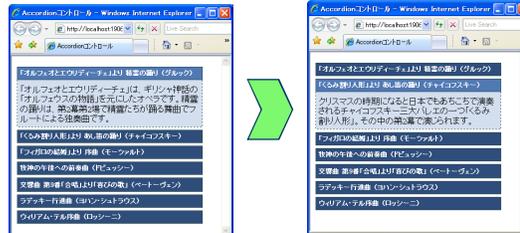
■ Extenderコントロールを中心に、単独で動作する高機能なコントロールも

分類	概要
入力支援	入力を容易にするためのフォーム部品、または、既存のフォーム部品に入力効率化のための機能を付加するためのコントロール群 AutoComplete / CascadingDropDown / PopupControl / Calendar / ListSearch / Rating / NumericUpDown / Slider / MutuallyExclusiveCheckBox / TextBoxWatermark / ToggleButton / DropDown
検証支援	それ自身が検証作業を行うものではないが(一部例外あり)、入力値のフィルタ/マスキングを行ったり、検証メッセージのバルーン表示、ボット対策を行うなど、不正な値が投入されるのを未然に防ぐためのコントロール群 FilteredTextBox / MaskedEdit / PasswordStrength / ConfirmButton / ValidatorCallout / NoBot
リスト系	伸縮可能なマルチペインやドラッグ&ドロップで順番の変更が可能なりスト、スライドショー機能などリッチなりスト機能を提供するコントロール群 Accordion / PagingBulletedList / ReorderList / SlideShow
パネル系	モーダルウィンドウやタブパネル、ドラッグパネルなど、パネル系の機能を提供するコントロール群 CollapsiblePanel / ModalPopup / Tabs / HoverMenu / DragPanel / AlwaysVisibleControl / ResizableControl
視覚効果	アニメーションやブロック領域の形状の制御など、視覚的な効果を提供するコントロール群 Animation / UpdatePanelAnimation / DropShadow / RoundedCorners
その他	上記に分類されなかったコントロール。 サービスメソッドから取得したコンテンツをページに反映するDynamicPopulate

ASP.NET AJAX Control Toolkit

- 参考) Accordionコントロールによる伸縮可能なマルチペイン -

■ 伸縮可能なマルチペイン(データベース連携)をコーディングレスで実現



```

<ajaxToolkit:ToolkitScriptManager ID="manager" runat="server" />
<ajaxToolkit:Accordion ID="acc" runat="server" ... DataSourceID="sds">
  <HeaderTemplate><%# Eval("nam") %>
  (<%# Eval("comp") %>)</HeaderTemplate>
  <ContentTemplate><%# Eval("desc") %></ContentTemplate>
</ajaxToolkit:Accordion>
<asp:SqlDataSource ID="sds" runat="server"
  ConnectionString="<%$ ConnectionStrings:MyDB %>"
  SelectCommand="SELECT nam, comp, desc FROM data" />
  
```

→ データソースとの関連付け

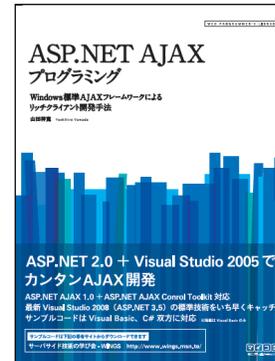
ヘッダ部 / コンテンツ部のテンプレートを定義

→ データベースへのアクセス

参考資料

■ AJAX技術学習のための書籍 / 記事

- ▶ 「ASP.NET AJAXプログラミング
 - Windows標準AJAXフレームワークによるリッチクライアント開発手法 -」
 - (毎日コミュニケーションズ: ISBN 978-4-8399-2438-6)
 - <http://www.wings.msn.to/index.php/-/A-03/978-4-8399-2438-6/>
- ▶ 連載: Ajax時代のJavaScriptプログラミング再入門
<http://www.atmarkit.co.jp/fdotnet/ajaxjs/index/>
- ▶ CodeZine: 触ってみようASP.NET 2.0 AJAX Extensions
<http://codezine.jp/a/article/aid/799.aspx>
- ▶ WebDeli - Spicy Tools, Delicious Sites-
<http://www.web-deli.com/>



次期Visual Studio 2008 + .NET Framework 3.5

■ 2006～2007年まで五月雨に登場した新技術を統合

- Visual Studio 2005 (.NET Framework 2.0)の登場が2006年2月
- Visual Studio 2005の登場以来、多くの新技術 / 製品がリリースされた

時期	製品 / 技術名	開発環境
2006.11	.NET Framework 3.0 (WPF / WF / WCF)	Visual Studio 2005 Extensions ただし、WPF / WCFについてはCTP
2006.11	2007 Microsoft Office System	Visual Studio 2005 Tools for 2007 Office System
2007.01	ASP.NET AJAX 1.0 & Control Toolkit	ASP.NET AJAX 1.0

- 開発にはVisual Studio 2005へのアドオンや追加コンポーネントのインストールが必要であった(一部は正式な開発環境が不在)
- Visual Studio 2008はこれら最新プラットフォーム / 技術を活用するためのIDE

次期Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能(JavaScript関連(1)) -

■ JavaScript開発機能の大幅な強化

- AJAX技術の中核であるJavaScriptはWebアプリケーション開発に欠かせない
- Visual Studio 2005でも部分的にJavaScriptのインテリセンスやデバッグ機能に対応
→ But、インテリセンスは外部ライブラリを認識しないなど、対応は不十分
- インテリセンス機能の強化
 - JavaScript標準のキーワードやメソッドだけでなく、外部ライブラリも認識
 - AJAX Extensionsのプロキシクラス(引数の型含む)も認識
 - 途中でJavaScriptのデータ型が変更になった場合も認識して、適切なメンバを表示

```

23 <script type="Text/javascript">
24   Sys.Application.
25
26 </script>
27 </head>
28 <body>
29   <form id="form1" runat="server">
30     <div class="form1">
31
32     </div>
33     <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
34     <cc1:SliderExtender ID="SliderExtender1" runat="server"
35       TargetControlID="TextBox1" runat="server">
36     </cc1:SliderExtender>
  
```

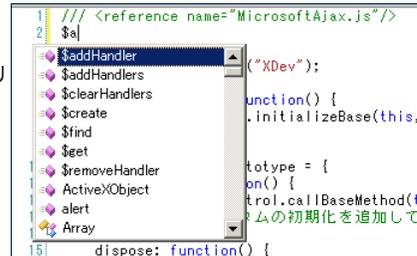
add_disposing
add_init
add_load
add_propertyChanged
add_unload
addComponent
beginCreateComponents
beginUpdate
constructor
dispose

次期 Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能(JavaScript関連(2)) -

■ 至れり尽くせりのJavaScript対応

- 「.js」ファイルでも<reference>要素を追加することで、外部ライブラリのインテリセンスを有効にできる
- パラメータツールチップも強化
→ ドキュメンテーションコメントをツールチップに反映も可能



```
function helloWorld(str){
  ///<summary>指定された文字列から挨拶を生成</summary>
  ///<param name="str">挨拶に含める名前</param>
  ///<return>void</return>
  window.alert("こんにちは、" + str + "さん!");
}
```

```
</asp:ScriptManager>
<script type="text/javascript">
helloWorld(
helloWorld(str)
str:
  挨拶に含める名前
</body>
<form id="form1" runat="server">
```

次期 Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能(JavaScript関連(3)) -

■ JavaScriptデバッグ機能対応も大幅に強化

- ブ레이크ポイントや変数監視のためのウォッチ機能など、デバッグ機能に対応

```
1 /// <reference name="MicrosoftAjax.js" />
2
3 function helloWorld(str){
4   ///<summary>指定された文字列から挨拶を生成</summary>
5   ///<param name="str">挨拶に含める名前</param>
6   ///<return>void</return>
7   window.alert("こんにちは、" + str + "さん!");
8 }
9
```

ローカル

名前	値	型
str	"山田"	String

ソリューション エクスプローラ

- Microsoft Internet Explorer
 - Default.aspx
 - WebResource.axd?id=ID5ynX...&G3eYpM2hevsB6A2
 - ScriptResource.axd?id=0JA1GI0vcHD7nyxuBIEC
 - ScriptResource.axd?id=0JA1GI0vcHD7nyxuBIEC
 - ClientLibrary.js
 - ScriptResource.axd?id=9MDVo3xgmQP5zW0C0_eJ
 - ScriptResource.axd?id=9MDVo3xgmQP5zW0C0_eJ
 - ScriptResource.axd?id=9MDVo3xgmQP5zW0C0_eJ
 - ScriptResource.axd?id=9MDVo3xgmQP5zW0C0_eJ
 - ScriptResource.axd?id=9MDVo3xgmQP5zW0C0_eJ
 - ScriptResource.axd?id=9MDVo3xgmQP5zW0C0_eJ
- C:\...\.XDev\
 - App_Code
 - Service.vb
 - App_Data
 - Bin
 - ClientControl.js

- ブ레이크ポイントは、ScriptManagerコントロールによって動的に呼び出されたアセンブリ内のリソース (Webリソース) に対しても設定が可能
- JavaScriptのコンポーネント、ライブラリの骨組み (テンプレート) も標準で提供

次期Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能 (デザイン関連 (1)) -

■ CSSサポート強化でページデザインも効率化

➢ “スタイルの管理” ウィンドウによってCSSスタイルの管理が容易に

➢ ソースビューからは利用可能なCSSクラス名に対してインテリセンス機能が有効に

次期Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能 (デザイン関連 (2)) -

■ CSSサポート強化でページデザインも効率化

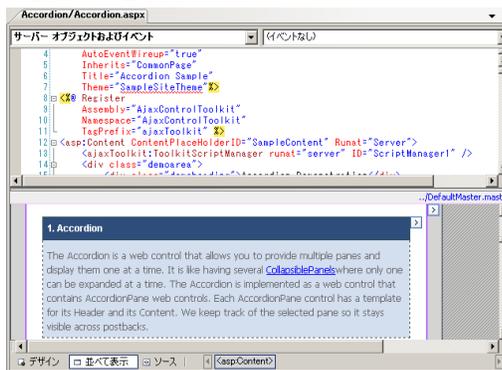
➢ CSSプロパティウィンドウで複雑なCSSの適用状態がより確認しやすく
→ 要素の選択によって現在の設定状況を一望できる

次期Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能(デザイン関連(3)) -

■ フォームデザインの作業がより効率化

- 入れ子のマスタページがVisual Studio上でもデザイン可能に
 - 従来のASP.NET 2.0でも入れ子自体は対応(しかし、Visual Studio 2005は未対応)
- デザイン/コードビューに加え、「分割ビュー」を導入
 - デザインを確認しながら、生成されたコードの確認が可能(リアルタイム同期)
 - 予期せぬタグの混入がないかを確認できる
 - Web標準の浸透と、「美しい」コードの重要性
- マルチターゲットサポート(.NET Framework 2.0 / 3.0 / 3.5をサポート)
 - 既存のソフトウェアはそのままに、Visual Studio 2008の新機能を利用できる



次期Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能(ASP.NET AJAX) -

■ ASP.NET AJAXはどうなるの？

- ASP.NET AJAX中核のライブラリは、ASP.NET 3.5に統合
 - ASP.NET AJAX 1.0とASP.NET 3.5のAJAX機能はサイドバイサイドで導入可
- 統合に伴い、細かな改善も多数
 - アプリケーションブリッジ(認証/プロファイルアクセス)機能の強化
 - WebPartsコントロールへのUpdatePanelの対応
 - Webサービスブリッジ機能のWCF(Windows Communications Foundation)対応
- ASP.NET AJAX Control Toolkit は引き続き拡張コントロールとして提供
 - 現時点でASP.NET 2.0 / 3.5双方の対応バージョンをダウンロード可能
 - ASP.NET本体とは別に、今後も短スパンで機能追加の予定
- ASP.NET Futures: ASP.NETの将来バージョンで追加予定の開発中機能
 - ブリッジクラス(「.asbx」ファイル)によるWeb APIアクセス機能
 - クライアントサイドで動作する宣言型言語XML Scriptへの対応
 - Silverlight対応(XAML / Mediaコントロール)
 - 動的言語への対応(IronPython / ManagedJScriptなど)
 - Dynamic Dataコントロール

次期Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能(ASP.NETサーバコントロール(1)) -

■ 指定テンプレートに従って、リストを生成するListViewコントロール

- ListViewコントロールで利用可能な主なテンプレート

テンプレート	概要
LayoutTemplate	リスト全体(プレースホルダを指定可)
ItemTemplate	個別の項目
AlternatingItemTemplate	個別の項目(一行おき)
SelectedItemTemplate	選択された項目
EditItemTemplate	編集モードの項目
InsertItemTemplate	挿入モードの項目
EmptyDataTemplate	データソースの内容が空の場合の表示
ItemSeparatorTemplate	項目の区切り

- GridView、DataListコントロールにも似ているが、ListViewコントロール自体がマークアップを出力しない点がポイント(テンプレート定義されたタグのみを出力) 高機能なRepeaterコントロールのようなもの？

次期Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能(ASP.NETサーバコントロール(2)) -

■ ListViewコントロールによるリスト整形のコード例

```
<asp:ListView ID="ListView1" runat="server" DataSourceID="XmlDataSource1">
  <LayoutTemplate>
    <ul id="itemContainer" runat="server">
      </ul>
    </LayoutTemplate>
    <ItemTemplate>
      <li style="background-color: #E0FFFF;color: #333333;">aid:
        <asp:Label ID="aidLabel" runat="server" Text="<# Eval("aid") %>" />
        ...中略...
      </ItemTemplate>
    <EmptyDataTemplate>
      データは返されませんでした。
    </EmptyDataTemplate>
    <itemseparatortemplate>
      <br />
    </itemseparatortemplate>
  </asp:ListView>
```

リスト項目を格納するコンテナ
(全体の骨組み)を定義

次期Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能(ASP.NETサーバコントロール(3)) -

■ ページング機能を提供するDataPagerコントロール

- コントロールに対してページング機能を追加
- ターゲットとなるコントロールに対してページャの位置を自由に設定できるのが特長

```
<asp:DataPager ID="DataPager1" runat="server">
<Fields>
<asp:NextPreviousPagerField ButtonType="Button"
  FirstPageText="最初" LastPageText="最後"
  NextPageText="次へ" PreviousPageText="前へ"
  ShowFirstPageButton="True" ShowNextPageButton="False"
  ShowPreviousPageButton="False" />
<asp:NumericPagerField NextPageText="..." PreviousPageText="..." />
</Fields>
</asp:DataPager>
```

前後ページのレイアウトを定義0

↑ 数値ページのレイアウトを定義

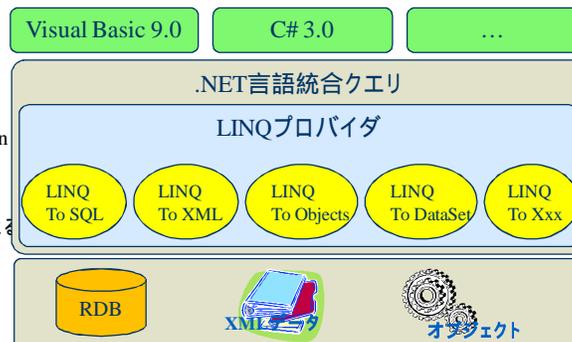


次期Visual Studio 2008 + .NET Framework 3.5

- Visual Studio 2008の新機能(ASP.NETサーバコントロール(4)) -

■ LINQデータをデータソースとして参照するLinqDataSourceコントロール

- LINQ(.NET Language Integrated Query :言語統合クエリ)とは....
リレーショナルデータベースやオブジェクト、XMLデータなどへの一元的なアクセス手段(プログラミングモデル)を提供
- SQL構文に酷似した馴染みやすい構文が特長
Ex.
var result=
from x in Comp.Person
select x;
- セッションC-2「.NET Framework によるサーバーサイドのアプリケーションプラットフォーム」



ご静聴ありがとうございました

■ 「サーバサイド技術の学び舎 - WINGS」

- <http://www.wings.msn.to/>
- サーバサイド技術に関する記事や書籍、関連サイトなど最新の情報を日々紹介
- RSSフィードも配信中



サーバサイド技術の学び舎 - WINGS