



**Future  
Technology Days**



Microsoft®  
**Visual Studio®**

# “4”時代の次世代 Web 開発

マイクロソフト株式会社  
デベロッパー & プラットフォーム統括本部

Microsoft

# Agenda ~ Web 開発編 ~

1. Visual Studio Web 開発 オーバー ビュー
2. HTML/CSS/JavaScript クライアントサイド開発
3. Web フォーム
4. ASP.NET MVC
5. AJAX / jQuery / CDN
6. Web アプリケーションの配置
7. その他の新機能
8. まとめ

# Agenda ~ Silverlight 4 編 ~

1. Silverlight 4 概要
2. Silverlight 4 の新機能
3. Silverlight 4 の開発環境
4. まとめ
  - Appendix
  - 関連リソース



Microsoft®  
**Silverlight®**

# Visual Studio Web 開発 オーバー ビュー

*.NET Framework 4*  
*ASP.NET 4*

*Silverlight 4*

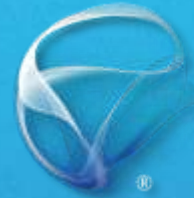
4

*Internet  
Information  
Services*

*Windows Azure  
Platform*



ASP.NET



Microsoft®  
**Silverlight**®



Microsoft®  
**Visual Studio**® 2010



**Windows Server**  
Internet Information Services



**Windows**® Azure™

# Web 開発 オーバービュー (1)

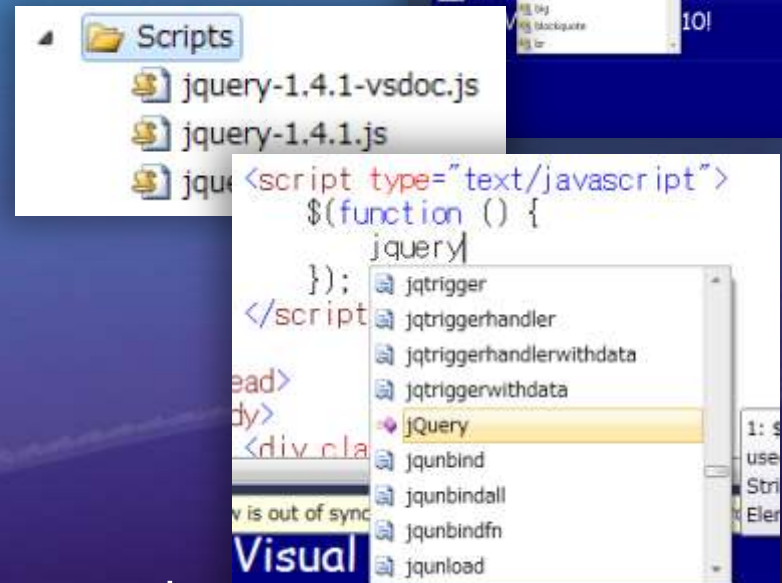
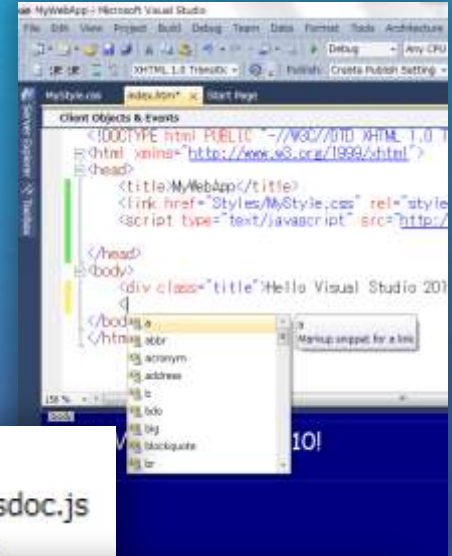
## Visual Studio 2010 で出来ること

### – Web ページ デザイン

- HTML, CSS などを活用したデザイン
- HTML, JavaScript, CSS の入力支援
- Microsoft Expression Web との連携

### – 処理ロジックの記述

- クライアントサイド
  - JavaScript, jQuery, ...
- サーバーサイド
  - C#, Visual Basic, ...
  - .NET Framework の利用
- データ アクセス
  - LINQ, ADO.NET Entity Framework, ...
- ASP.NET 認証機能の利用



# Web 開発 オーバービュー (2)

## Visual Studio 2010 で出来ること

### - デバッグ

- 開発用 Web サーバーを使ったデバッグ
- JavaScript のステップ実行や変数ウォッチ
- テスト駆動開発 (TDD) サポート



### - 配置と展開

- ワンクリック 発行
- Web 配置パッケージ
- Web セットアップ プロジェクト





# Web 開発 オーバービュー (3)

## Visual Studio 2010 で出来ること

### Web Forms

GUI ベースの  
Web ページ開発

### ASP.NET AJAX

ASP.NET に統合された  
AJAX フレームワーク



### ASP.NET MVC

MVC パターンによる  
Web 開発

### Dynamic Data

データ駆動型 Web アプリ  
ケーションの開発

さまざまな .NET テクノロジーを活用した  
Web 開発

# Web プロジェクトの種類

## Visual Studio 2010 Web アプリケーション

カテゴリ	プロジェクトの種類
Web フォーム	ASP.NET Web アプリケーション
	ASP.NET 空の Web アプリケーション <b>New!</b>
MVC	ASP.NET MVC 2 Web アプリケーション <b>New!</b>
	ASP.NET 空の MVC 2 Web アプリケーション <b>New!</b>
Dynamic Data	動的データ LINQ to SQL Web アプリケーション
	動的データ エンティティ Web アプリケーション
コントロール	ASP.NET サーバー コントロール
	ASP.NET AJAX サーバー コントロール
	ASP.NET AJAX サーバー コントロール エクステンダ

# ASP.NET と Visual Studio

## 各種フレームワークの関係



A diagram showing the software stack for ASP.NET 3.5 in Visual Studio 2008. It consists of several layers: a green button for 'AJAX Control Toolkit', an orange button for 'ASP.NET MVC 1.0 / 2', a purple rounded rectangle for 'ASP.NET 3.5' containing an orange button for 'ASP.NET AJAX' and a green button for 'Microsoft AJAX Library', and an orange button for 'Dynamic Data'. At the bottom is a blue rounded rectangle for '.NET Framework 3.5 (.NET Framework 3.0 / 2.0)'. To the right is the Visual Studio 2008 logo.

AJAX Control Toolkit

ASP.NET MVC 1.0 / 2

ASP.NET 3.5

ASP.NET AJAX

Microsoft AJAX Library

Dynamic Data

.NET Framework 3.5  
(.NET Framework 3.0 / 2.0)

Microsoft  
Visual Studio® 2008



A diagram showing the software stack for ASP.NET 4 in Visual Studio 2010. It consists of several layers: a green button for 'AJAX Control Toolkit', a green cloud for 'jQuery', a purple rounded rectangle for 'ASP.NET 4' containing an orange button for 'ASP.NET MVC 2', an orange button for 'ASP.NET AJAX', and a green button for 'Microsoft AJAX Library', and an orange button for 'Dynamic Data'. At the bottom is a blue rounded rectangle for '.NET Framework 4 (.NET Framework 3.5 / 3.0 / 2.0)'. To the right is the Visual Studio 2010 logo.

AJAX Control Toolkit

jQuery

ASP.NET 4

ASP.NET MVC 2

ASP.NET AJAX

Microsoft AJAX Library

Dynamic Data

.NET Framework 4  
(.NET Framework 3.5 / 3.0 / 2.0)

Microsoft  
Visual Studio® 2010

※ VS 2008 / .NET Framework 3.5 は SP1

# HTML / CSS / JavaScript クライアントサイド開発

# Web 標準 と Visual Studio オーバービュー

- **HTML 標準**への対応
  - HTML 5 対応を視野に...
- **CSS 標準**への対応
  - CSS 3 対応を視野に...
- **JavaScript**
  - jQuery の正式サポート
- **Web アクセシビリティ 指針**への対応



# Web 標準 と Visual Studio

## HTML / CSS 支援機能

### – HTML 検証

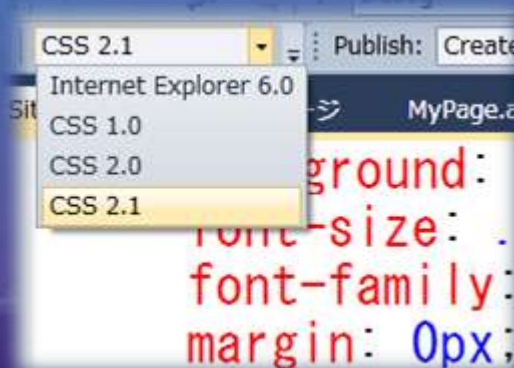
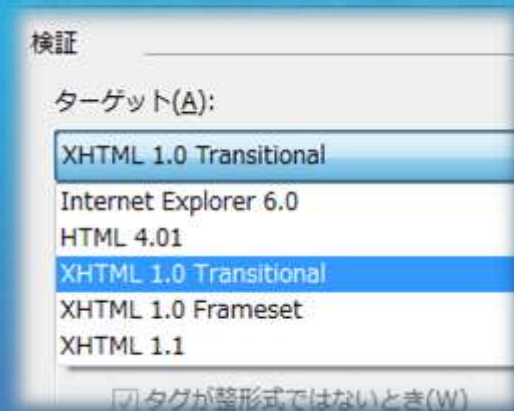
- HTML 4.01 / XHTML 1.0 等のスキーマに沿った検証

### – CSS 検証

- CSS 1.0 / 2.0 / 2.1 に対する検証

### – アクセシビリティ検証

- WCAG\* / Access Board 508\* に沿ったアクセシビリティ検証



※ WCAG (Web Content Accessibility Guidelines) : W3C 勧告  
※ Access Board 508 : 米国連邦政府ハビリテーション法 508 条

# JavaScript と Visual Studio

## デバッグ対応

ブレーク  
ポイント

スクリプト  
ドキュメント

ツールヒント  
による変数の  
ウォッチ

イミディエイト  
ウィンドウ

ウォッチ  
ウィンドウ

ブレーク  
ポイント一覧

The screenshot displays the Visual Studio IDE with a JavaScript file open. The code shows a `domReady` function. A yellow highlight is on the `$('#btn1').click` line, and a red dot indicates a breakpoint. The `Watch` window shows the `document` object and its properties. The `Breakpoints` window shows the active breakpoint. The `Immediate Window` shows the current state of the `document` object. The `Solution Explorer` shows the project structure.

名前	値	型
document	{object}	DispHTMLDoc
[メソッド]		
[イベント]		
activeElement	{object}	DispHTMLBody
alinkColor	"#0000ff"	String
all	{カウント = 11}	DispHTMLElem
anchors	{カウント = 0}	DispHTMLElem
applets	{カウント = 0}	DispHTMLElem
attributes	null	Object
bgColor	"#ffffff"	String
body	{object}	DispHTMLBody
charset	"utf-8"	String
childNodes	{カウント = 2}	DispHTMLNode
compatible	"CSS1Compat"	String
constructor	{function}	Function

名前	ラベル	条件
jQueryCDN.htm, 11行 13文字		(条件なし)

```
document
{object}
  activeElement: {object}
  alinkColor: "#0000ff"
  all: {カウント = 11}
  anchors: {カウント = 0}
  applets: {カウント = 0}
  attributes: null
  bgColor: "#ffffff"
  body: {object}
  charset: "utf-8"
  childNodes: {カウント = 2}
```

# インテリセンス と スニペット

## Visual Studio 2010 入力支援機能

### - インテリセンス

- オートコンプリート・オートコレクト

### - コード スニペット

- HTML と JavaScript でも使用可能に！
- コード ブロックのひな形を簡単に挿入
- カスタム スニペットを作成して拡張可能

The screenshot illustrates the Visual Studio 2010 code snippet feature. On the left, a list of markup snippets is shown, with 'actionlink' selected. A blue arrow points from this list to a larger view of the selected snippet. The snippet is a code block for an ASP.NET page, containing a table with headers 'UserName', 'Message', and 'ImageSource'. The snippet is highlighted with a red box, and the text '<%= Html.ActionLink("linktext", "actionname") %>' is highlighted in yellow.

```
<h2><%= Html.Encode(ViewData["Message"]) %></h2>
<p><%= Request.UserAgent %></p>
<act|
  a
  abbr
  accessdatasource
  acronym
  actionlink
  address
  asp:AccessDataSource
  asp:AdRotator
  asp:AppearanceEditorPart
  ImageSource
  Markup snippet for an ASP.NET
  <table>
    <tr>
      <th>
        UserName
      </th>
      <th>
        Message
      </th>
      <th>
        ImageSource
      </th>
    </tr>
  </table>
  <%= Html.ActionLink("linktext", "actionname") %>
```



# Web フォーム

# Web フォーム と Web 標準

## HTML / CSS 互換性の強化



- レンダリングの互換性
  - XHTML 1.0 Strict として HTML 要素を生成
    - `<xhtmlConformance mode="Strict" />` の扱いと同じ

### Web.config

```
<system.web>  
  <pages controlRenderingCompatibilityVersion="4.0"/>  
</system.web>
```

### 例：ラベルコントロール

```
<asp:Label id="Label1" runat="server"  
  Text="ラベル" Enabled="false">
```

### 従来 (3.5) のレンダリング

```
<span id="Label1" disabled="disabled">ラベル</span>
```

### 4.0 のレンダリング

```
<span id="Label1" class="aspNetDisabled">ラベル</span>
```

# Web フォームと HTML

## クリーン HTML の生成 (1)

- ClientIDMode プロパティ
  - クライアント ID の生成方法を厳密に指定
  - コントロール単位、ページ単位で指定可能

```
<asp:Label id="Label1" runat="server" ...  
  ClientIDMode="AutoID|Predictable|Static|Inherit">
```

AutoID	<span id="ctl00_MainContent_Label1"></span>
Predictable	<span id="MainContent_Label1"></span>
Static	<span id="Label1"></span>

- AJAX や CSS によるクライアント処理を簡単に

# Web フォームと HTML

## クリーン HTML の生成 (2)

### – ViewStateMode プロパティ

- ViewState の継承関係を考慮した詳細なコントロール
- コントロール単位、ページ単位で指定可能

```
<asp:Label id="Label1" runat="server" ...  
    ViewStateMode="Enabled|Disabled|Inherit">
```

<例>

aspx ページ (ViewStateMode="Disabled")

コントロール A

ViewState

ViewStateMode="Enabled"

コントロール B

~~ViewState~~

ViewStateMode="Disabled"

コントロール C

~~ViewState~~

ViewStateMode="Inherit"

- 必要最小限の ViewState でパフォーマンス UP!

# Web フォームと HTML

## クリーン HTML の生成 (3)

- **RenderOuterTable** プロパティ
  - 外部 Table 要素やインライン スタイル属性を使用しない HTML レンダリング
  - *FormView, Login, PasswordRecovery, ChangePassword*

```
<asp:Login id="LoginUser" runat="server"  
    RenderOuterTable="true|false">
```

```
<table ... style="border-collapse:collapse;">  
    <div> Content </div>  
</table>
```

外部 Table の  
有無を設定可能に

- CSS を使用したスタイル付けが簡単に

# ASP.NET MVC

# ASP.NET MVC

## 新しい Web アプリ開発フレームワーク



### – ASP.NET MVC 1.0

- 2009 年 4 月 RTM リリース
- Visual Studio 2008 の追加コンポーネント

### – ASP.NET MVC 2

- 2010 年 3 月 RTM リリース (for VS 2008 SP1)
- Visual Studio 2010 (ASP.NET 4) に標準搭載

- **HTML ベース**の UI 開発が容易に
- UI と処理ロジックを**疎結合**に
- **単体テスト** (TDD) を容易に
- **URL ルーティング**が可能に

# MVC とは？

## Web アプリにおける MVC パターン

1. Controller が  
リクエストを受信

HTTP Request

Controller

2. Controller が  
Model を初期化

3. ビジネス ロジック  
を実行

入力と制御

4. Controller が View  
を呼び出す

Model

プレゼンテーション

5. View が Model  
を参照

データ表現と  
ビジネス ロジック

HTTP Response

View

6. View がレスポンス  
を送信



# Model / View / Controller

## 実装コード例

### Model と Controller

```
public class HomeController : Controller {
```

Model の準備

```
    MyModelService myModel = new MyModelService();
```

```
    public ActionResult Edit(int id) {  
        MyData md = myModel.GetData(id);
```

HTTP リクエスト  
によるメソッドの  
実行  
(例: Home/Edit/5)

```
        return View(md);
```

ロジックの実行と  
データ取得

```
    }  
}
```

View の呼び出し

### View

```
<html>
```

```
    <p><%= Html.TextBox("data", Model.data) %></p>
```

HTML タグ生成と  
データの埋め込み

```
</html>
```

# Controller と View の命名ルール

## コンセプト「設定より規約」

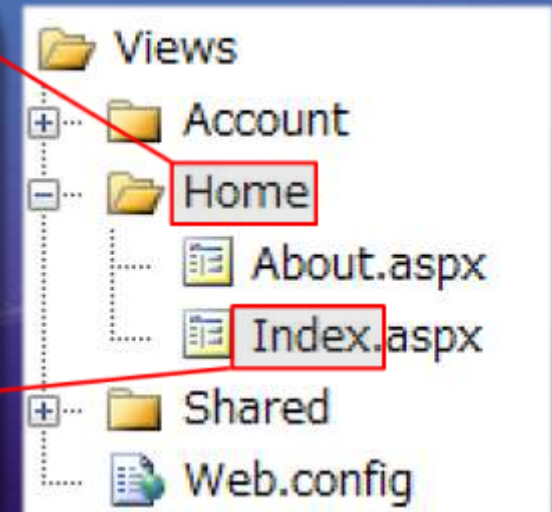
- CoC (Convention over Configuration)
  - “規約は設定に勝る”
- 命名ルールによってモジュール間の結合を疎に
  - コードや設定ファイルなどで関連付けない

### Controller クラス

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        ViewData["Message"] = "ASP.NET MVC!";
        return View();
    }
    ...
}
```

Controller 名

アクション名



# HTML ヘルパー メソッド

## HTML 生成を助ける関数群

メソッド名	概要
<b>Encode</b>	HTML エンコードをおこなう
<b>ActionLink</b>	アクション名やコントローラ名から a タグを生成
<b>BeginForm</b>	form タグを生成
<b>TextBox</b>	input タグによるテキスト入力フィールドを生成
<b>Password</b>	input タグによるパスワード入力フィールドを生成
<b>ListBox</b>	select タグによるリストボックスを生成
<b>CheckBox</b>	input タグによるチェックボックスを生成

### 使用例

※上記は代表的な HTML ヘルパーメソッド

```
<%= Html.TextBox("message", Model.Message) %>
```

```
<%= Html.ActionLink("Go to Home", "Index") %>
```

# ASP.NET MVC 2

## 代表的な新機能 (1)

- 強く型付けされた HTML ヘルパー と DataAnnotations バリデーション
  - Model 中心の HTML 生成と入力検証ルール

### View

```
<%= Html.LabelFor(model => model.data) %>  
<%= Html.TextBoxFor(model => model.data) %>  
<%= Html.ValidationMessageFor(model => model.data) %>
```

### Model

```
public class MyData {  
    [DisplayName("データ")]  
    [Required(ErrorMessage="データを入力してください")]  
    public string data { get; set; }  
}
```

# ASP.NET MVC 2

## 代表的な新機能 (2)

### – Areas

- 単一 MVC プロジェクト内でアプリケーションを分割
- *AreaRegistration.RegisterAllAreas()*

### – Templated Helpers

- テンプレートによる Auto-Scaffold UI ヘルパーメソッドのカスタマイズ
- *Html.DisplayForModel()*, *Html.DisplayFor()*, *Html.Display()*
- *Html.EditorForModel()*, *Html.EditorFor()*, *Html.Editor()*

### – RenderAction

- アクションメソッド呼び出しによる HTML の部分レンダリング
- *Html.RenderAction(<Action>, <Controller>)*

# ASP.NET MVC 2

## 代表的な新機能 (3)

### – AsyncController

- 非同期アクションメソッド呼び出しのサポート
- *void SomeMethodAsync()*
- *ActionResult SomeMethodCompleted()*

### – アクションメソッドのデフォルト値

- *ActionResult Index([DefaultValue(1)] int id)*
- *ActionResult Index(int id = 1)*

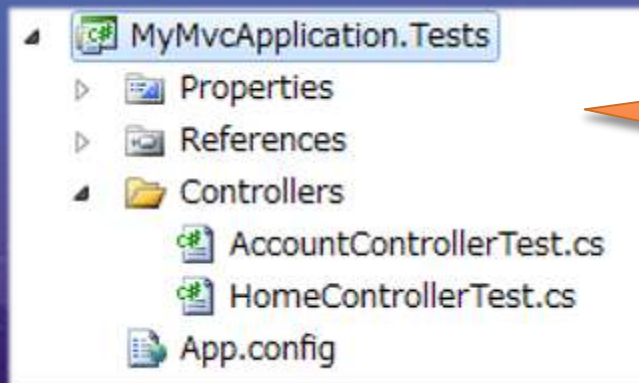
### – 新しい HTTP Method 属性

- *HttpPost, HttpGet, HttpPut, HttpDelete*

# ASP.NET MVC のテスト駆動開発

## Controller の単体テスト

- Controller メソッドの**単体テスト**が容易に
  - テスト時には IIS などの Web サーバー は不要
  - Visual Studio 2008/2010 Professional 以上
  - NUnit, MUnit, XUnit なども利用可能
- ASP.NET MVC のプロジェクトと同時にテストプロジェクトも作成できる
  - Action メソッドのテストコードの雛型が自動生成



生成された  
テストプロジェクト

# Web フォーム vs. MVC

## 特徴と使い分け

- ASP.NET MVC は Web フォーム を置き換えるものではない
- 適材適所で使い分けることが重要!
  - プロジェクトの要求、エンジニアのスキル、開発手法、など ...

Webフォーム	ASP.NET MVC
豊富なコントロールを使用可能	UI 開発は <b>HTML ベース</b>
イベント駆動型プログラミング	アクションメソッドによるリクエスト処理
ViewState やポストバック処理を多用する	ViewState やポストバック処理を使用しない
ロジックの検証に Web サーバーが必要	ロジックの検証に Web サーバーが不要 (テスト駆動開発をサポート)



# AJAX / jQuery / CDN

# ASP.NET と AJAX

## Visual Studio 2010 における実装



### - サーバー サイド AJAX には ...

( ASP.NET サーバー コントロール ベースの実装では ... )

#### - **ASP.NET AJAX**

- AJAX Extensions (サーバーコントロール群)
- Microsoft AJAX Library (JavaScript ライブラリ)

#### - **ASP.NET AJAX Control Toolkit**

- 高機能 AJAX サーバーコントロールとエクステンダ

### - クライアント サイド AJAX には ...

( HTML や JavaScript ベース、MVC の実装では ... )

#### - **jQuery**

- ASP.NET プロジェクトで標準採用

#### - **jQuery UI**

- jQuery ベースのコントロールライブラリ

# jQuery のサポート

## ASP.NET と jQuery



- 高速・軽量な JavaScript ライブラリ
  - DOM 操作 / CSS 操作 / Ajax 処理を容易に
  - オープンソース

- jQuery 公式サイト <http://jquery.com/>

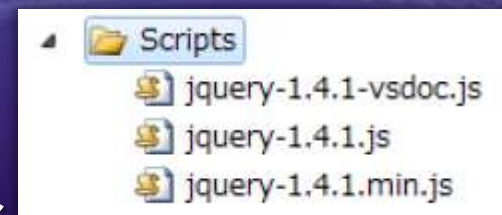
- 最新バージョンは 1.4.2

```
$(function() {  
    $('#button1').click(function() {  
        $('#div1').fadeIn('slow');  
    });  
});
```

- ASP.NET 4 と jQuery

- Web フォーム や MVC プロジェクト  
で標準採用

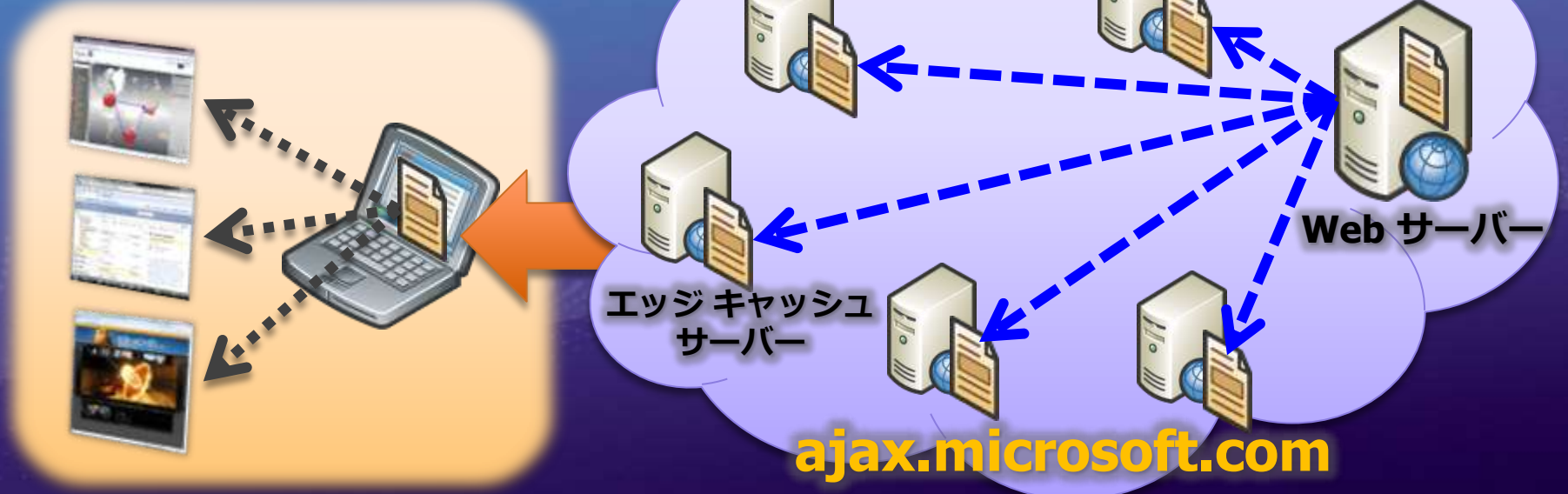
- jQuery 1.4.1 が標準搭載
- インテリセンスとコードスニペット  
にも対応 (詳細は Appendix を参照)



# Content Delivery Network

## CDN の活用

- Microsoft Ajax **C**ontent **D**elivery **N**etwork
  - [ajax.microsoft.com](http://ajax.microsoft.com) ドメインで主要 JS ファイルをホスト
  - Microsoft AJAX Library や jQuery の利用を容易に
  - 最小限のネットワーク ホップ数で効率的にロード
  - ブラウザーキャッシュに格納されたスクリプトをアプリケーション間で再利用
  - 無償サービス



# Microsoft Ajax CDN

## ホストされる JavaScript ライブラリ

- jQuery 1.4.2
- jQuery 1.4.1
- jQuery 1.4
- jQuery 1.3.2
- jQuery Validate 1.7
- jQuery Validate 1.6
- jQuery Validate 1.5.5
- AJAX Control Toolkit 40412
- ASP.NET AJAX 4
- ASP.NET AJAX 3.5
- ASP.NET MVC 1.0

※ URL 等、詳細はこちら ※

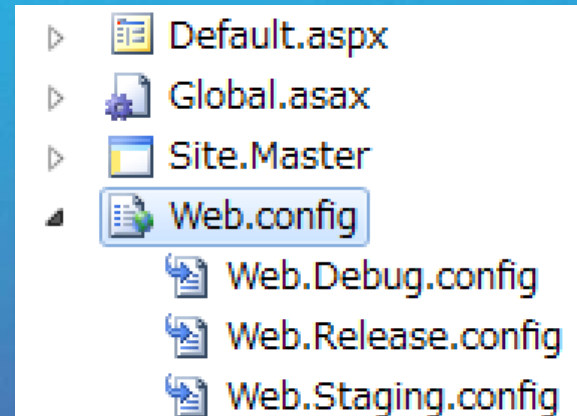
[http://www.asp.net/  
ajaxLibrary/CDN.ashx](http://www.asp.net/ajaxLibrary/CDN.ashx)

# Web アプリケーションの配置

# Web.config の変換

## Visual Studio 2010 配置新機能 (1)

- Debug, Release などのビルド設定毎に変換ルールを定義可能



### Web.xxxx.config

```
<configuration xmlns:xdt=
  "http://schemas.microsoft.com/XML-Document-Transform">

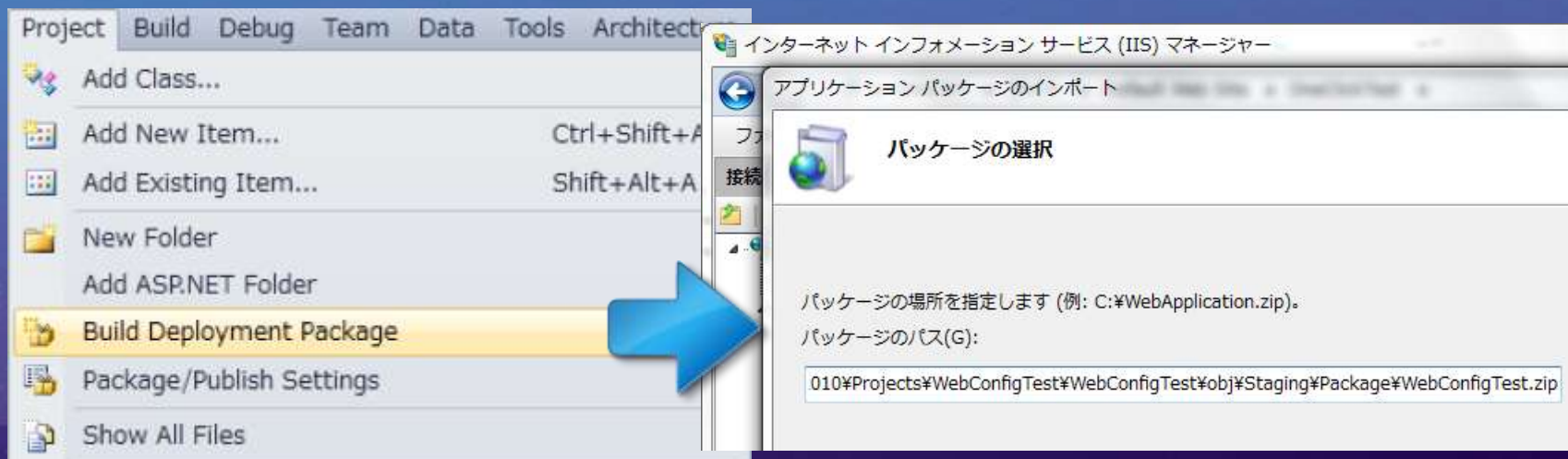
  <system.web>
    <compilation
      xdt:Transform="RemoveAttributes(debug)" />
    </system.web>

  </configuration>
```

# Web 配置パッケージ

## Visual Studio 2010 配置新機能 (2)

- アプリケーションの圧縮ファイル作成 (.zip)
  - Web コンテンツ、DB やスキーマ、IIS 設定等 ...
- Web 配置ツールによる IIS への配置
  - IIS マネージャー による手動配置 (GUI)
  - MSDeploy.exe を使った自動配置 (コマンドライン)

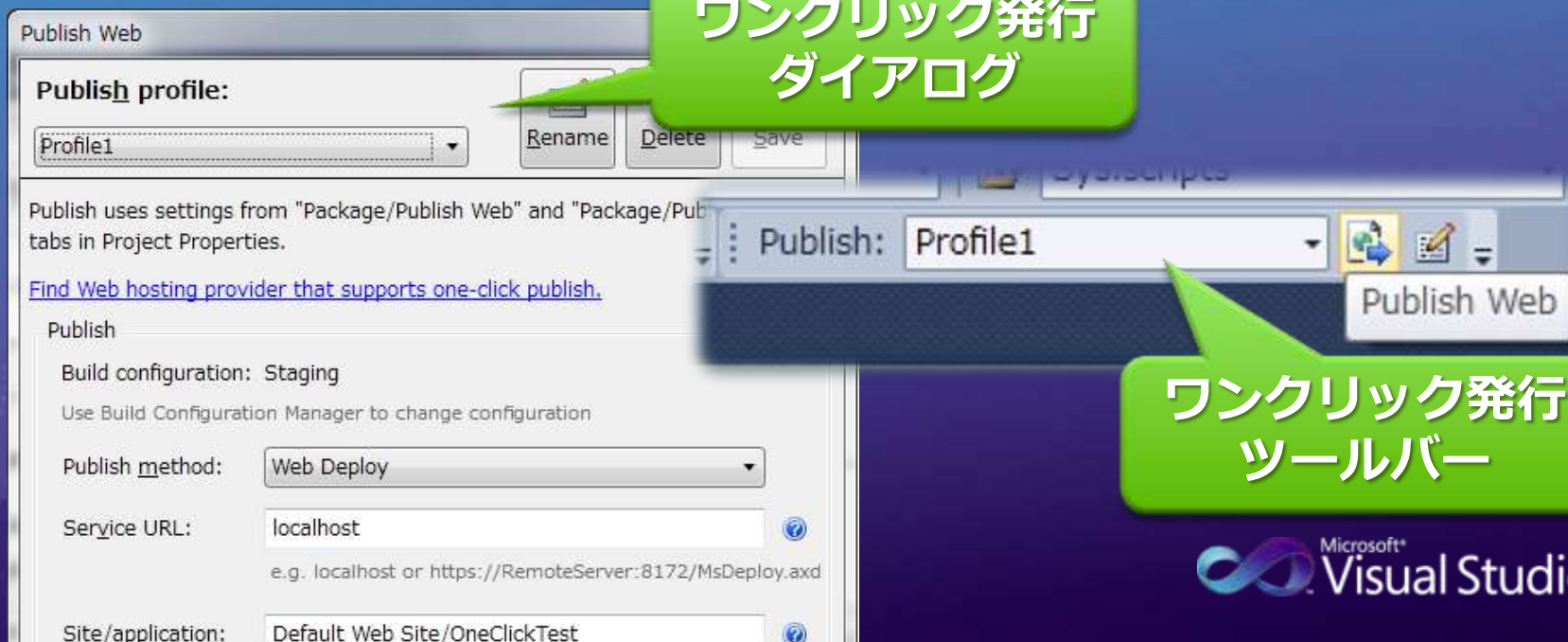




# ワンクリック発行

## Visual Studio 2010 配置新機能 (3)

- 一回のクリックで IIS へ配置
  - ローカル や リモート環境の Web サーバーへ
- 複数のプロファイルを作成、切り替えが可能
  - テスト用、ステージング用など ...



# その他の新機能

# URL ルーティング

## クリーン URL の生成

Web ページの配置パスに依存した URL

/Products/List.aspx?category=car

/Products/Details.aspx?category=car&id=5

URL ルーティング

/Products/Car/List/

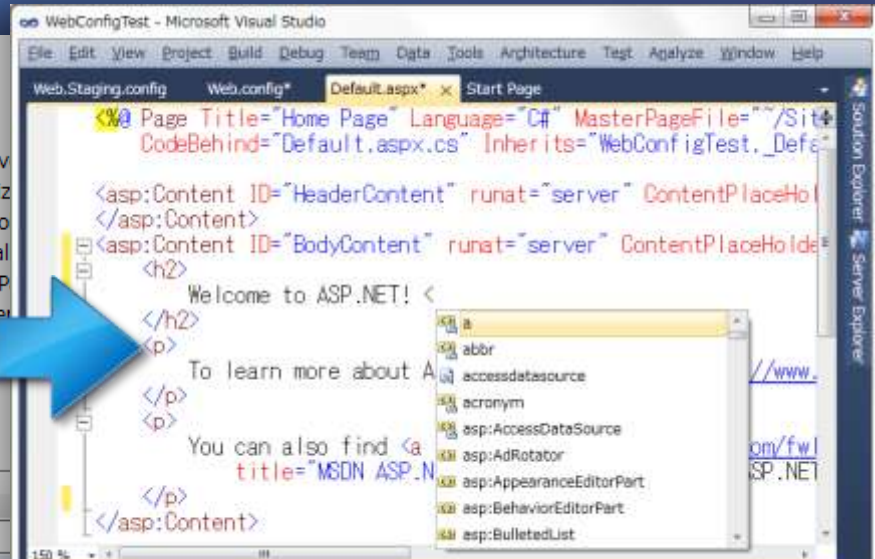
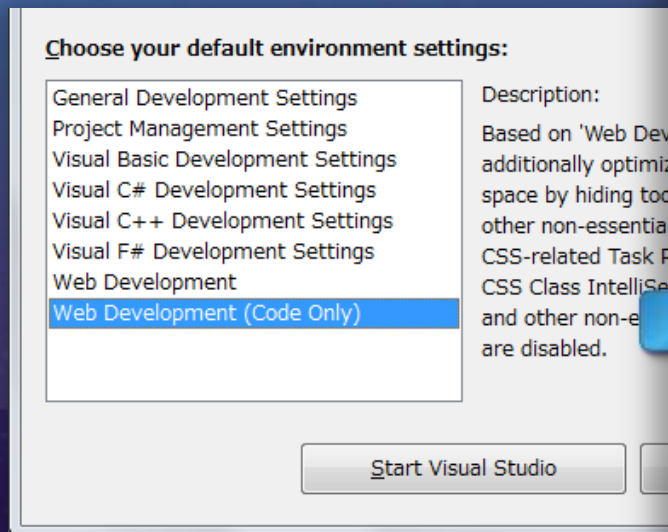
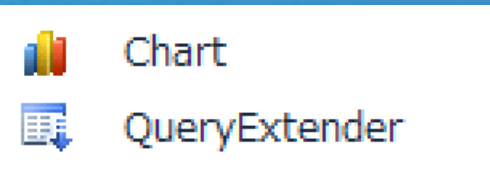
/Products/Car/Details/5

Web フォーム  
MVC  
Dynamic Data  
で利用可能に

- ✓ Web ページの URL を柔軟に構築可能
  - ✓ RESTful な Web アプリケーションの構築
  - ✓ SEO (Search Engine Optimization) 対策

# Visual Studio 2010 新機能 まだまだたくさん!

- 新しい HTML Encoding 構文
  - `<%: SomeStringData %>`
- 新しい サーバー コントロール
  - *Chart, QueryExtender*
- Web 開発に最適化された環境設定
  - *Web Development (Code Only)*



# まとめ ～ Web 開発編 ～

# まとめ

～ 一歩進んだ Web 開発のために ～

標準技術を生かした Web 開発

Web フォーム  
ASP.NET MVC  
ASP.NET AJAX  
Dynamic Data

  
Microsoft®  
Visual Studio® 2010

HTML  
CSS  
JavaScript  
jQuery

ASP.NET の豊富な機能を生かした Web 開発

# Silverlight 概要

# Silverlight

## 歴史



- (2006 年) コード ネーム **WPF/E**
  - WPF のサブセット版として開発が開始…
- 2007 年 9 月 **Silverlight 1**
  - 正式リリースだったが、開発言語は JavaScript に制限されメディア機能が中心。日本語も非対応。
- 2008 年 10 月 **Silverlight 2**
  - メジャーバージョンアップで、VB/C# での開発が可能に。また、コントロールの拡充など RIA プラットフォームとしての地位を確立。
- 2009 年 7 月 **Silverlight 3**
  - 様々な機能向上・メディア拡張によりアプリケーションとしての幅が広がる。ブラウザ外稼働も。





# Silverlight 4

- よりリッチなエクスペリエンスの実現
- ブラウザを超えた新たな形態
- 開発者のニーズに応えられる環境

Silverlight 4

# Silverlight 4 の新機能

# メディア

## 入力から配信・再生まで

- メディアデータの入力サポート
  - ウェブカムからの映像入力
  - マイクからの音声入力
  - データの録画・録音・記録
- より強化されたストリーミング
  - マルチキャストストリーミング
  - Windows Media サービス
  - 配信映像の保護
  - オフライン DRM



# 業務アプリケーション向け機能 アプリケーション要件への挑戦

- 業務アプリケーションで必要機能
  - 印刷機能
  - リッチ テキスト ボックス
  - クリップボードの利用
  - ドラッグ & ドロップ対応
- 最適なマウスオペレーション
  - 右クリックメニュー
  - マウスホイールへの対応



®

# より高いエクスペリエンスを さらに高度なアプリケーションへ

- 更に進化した多彩な表現力
  - アラビア語、ヘブライ語などにも対応し  
右から左への表記にも対応
  - Silverlight 内で HTML レンダリング
  - さらに拡張された 100 を超える  
コントロール群
  - Fluid スタイル
  - Implicit (暗黙の) スタイル

# ブラウザを超えて #1

## ブラウザ外稼働を更に拡張

- Sandbox モード
- 制限された環境下での拡張
  - Windows API 呼び出し
  - ポップアップメッセージ
  - HTML サポート
  - ドロップターゲット



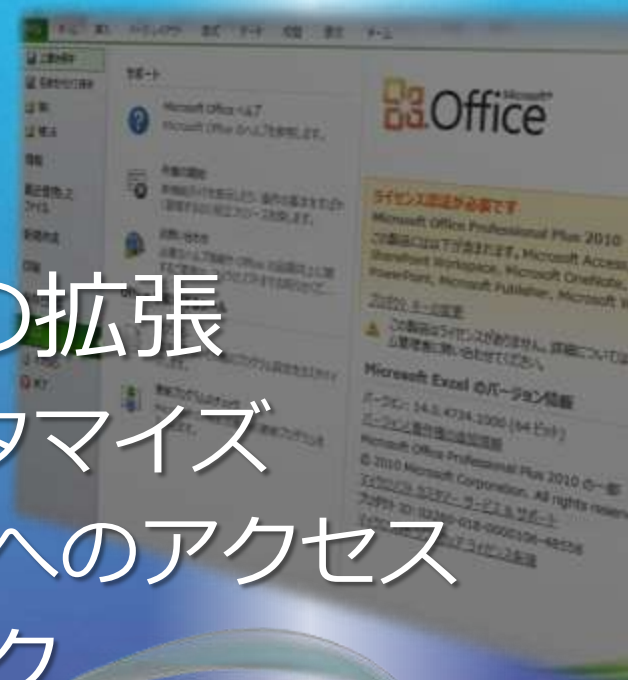
# ブラウザを超えて #2

## ブラウザ外稼働を更に拡張

### - Trusted モード

### - よりセキュアな環境下での拡張

- ウィンドウ デザインのカスタマイズ
- ローカル ファイル システムへのアクセス
- クロス ドメイン ネットワーク
- フル スクリーン モードでのキーボード対応
- ハードウェア デバイスへのアクセス
- COM オートメーションのサポート



# パフォーマンスも向上

Silverlight 4 になるだけで速くなる

- パフォーマンスアップも十分
  - 処理速度は **2 倍** になることも
  - 起動が **30%** 高速化
  - パフォーマンス プロファイリング



# Silverlight 4 の開発環境

# Silverlight 4

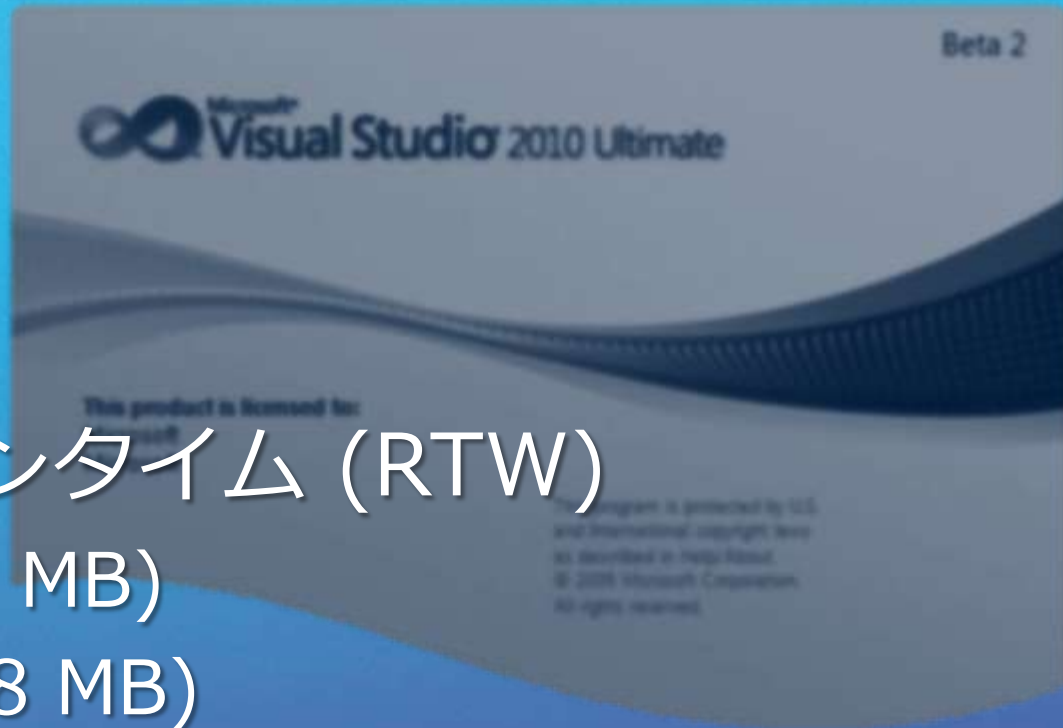
## 開発環境

### ■ 動作環境

- Silverlight 4 ランタイム (RTW)
  - Windows (8.14 MB)
  - Macintosh (17.8 MB)

### ■ 開発環境

- **Visual Studio 2010**
  - **Silverlight 4 Tools** for Visual Studio 2010 (RC2)
- Expression Blend 4 RC



# まとめ ～ Silverlight 4 編 ～

# まとめ



～ 一歩進んだ UX 開発のために ～

## – Silverlight 4 の開発環境は VS 2010 !

- 完全に編集可能なデザイン画面
- ドラッグ & ドロップによるデータバインド
- データソースに合わせてバインド相手となる最適なコントロールを自動的に選択
- XAML, C#, VB エディタの Intellisense サポート

## – Silverlight 4 新機能は開発者の為に

- よりリッチな UI が UX への流れを作る

## – Ready for Business !

- 業務アプリケーションの UX も RAD で実現



# Appendix

# jQuery と インテリセンス

## Visual Studio 2010 入力支援機能

### - jQuery インテリセンス対応

#### - jquery-1.x.x-vsdoc.js ファイルを提供

- インテリセンス向けのコメント情報が記述される
- HTML では通常の jquery-1.x.x.js への参照を追加
- IDE がプロジェクト内の -vsdoc ファイルを認識

```
<script src="jquery-1.4.1.js"  
        type="text/javascript"></script>
```

#### - 外部 JavaScript ファイルでは ...

```
/// <reference path="jquery-1.4.1.js" />  
function FadeOut() {  
    $("#Text1").fadeOut();  
    ...  
}
```

jQuery ファイル  
への参照の追加

# jQuery と スニペット

## Visual Studio 2010 入力支援機能



### - jQuery コードスニペット対応

- *jQuery Code Snippets for Visual Studio 2010*

<http://jquerysnippets.codeplex.com/>

対象	ショートカット	概要
HTML	<b>jq</b>	スクリプトタグと <code>\$(document).ready()</code> ショートカットを挿入
HTML	<b>jqscript</b>	jQuery ソースファイルへのスクリプトタグを挿入
JavaScript	<b>func</b>	JavaScript の匿名関数を挿入
JavaScript	<b>jqclick</b>	click イベントハンドラを挿入
JavaScript	<b>jqfadein</b>	fadeIn 関数を挿入

- 上記を含め 131 ものスニペットを提供

# Microsoft Ajax CDN

## 使用方法

### A) Script タグを記述して直接参照

jQuery

```
<script type="text/javascript" src=  
  "http://ajax.microsoft.com/ajax/jquery/jquery-1.4.2.js">  
</script>
```

### B) ASP.NET 4 の ScriptManager を利用

ScriptManager

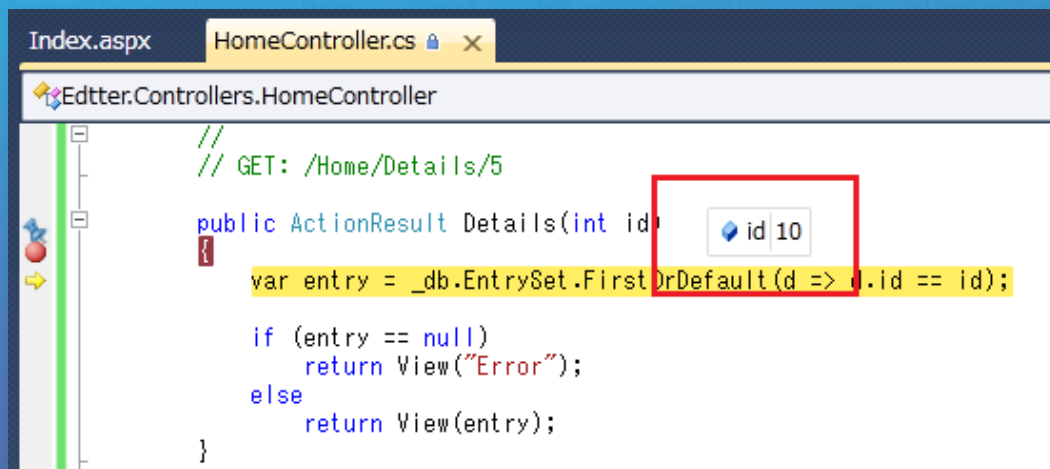
```
<asp:ScriptManager ... runat="server" EnableCdn="true" />
```



# デバッグ変数のピン止め (データヒント)

## Visual Studio 2010 のデバッグ新機能

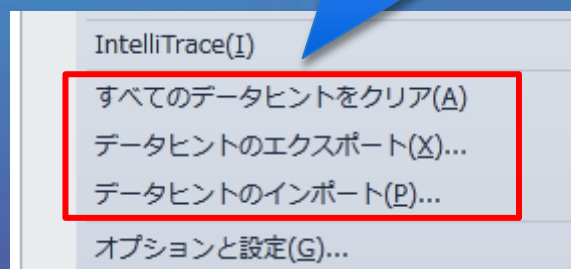
- デバッグ中の変数の値をコードエディタ上に常に表示



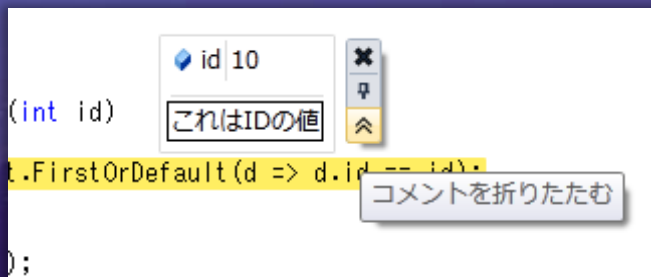
```
Index.aspx HomeController.cs x
Edtter.Controllers.HomeController
// GET: /Home/Details/5
public ActionResult Details(int id)
{
    var entry = _db.EntrySet.FirstOrDefault(d => d.id == id);

    if (entry == null)
        return View("Error");
    else
        return View(entry);
}
```

インポート/エクスポート  
も可能



- ピン止めした変数の値にコメントも付加可能



# Web カムからのキャプチャー

## Silverlight 4

```
if (CaptureDeviceConfiguration.AllowedDeviceAccess
    || CaptureDeviceConfiguration.RequestDeviceAccess())
{
    VideoCaptureDevice vcd =
        CaptureDeviceConfiguration.GetDefaultVideoCaptureDevice();
    if (null != vcd)
    {
        cs = new CaptureSource();
        cs.VideoCaptureDevice = vcd;
        cs.Start();

        VideoBrush videoBrush = new VideoBrush();
        videoBrush.Stretch = Stretch.Uniform;
        videoBrush.SetSource(cs);
        TO_FILL.Fill = videoBrush;
```



# 印刷

## Silverlight 4

```
PrintDocument pd = new PrintDocument();  
pd.DocumentName = "印刷文書の名前";  
pd.PrintPage += (s, args) =>  
{  
    args.PageVisual = LayoutRoot;  
    args.HasMorePages = false;  
};  
pd.EndPrint += (s, args) =>  
{  
    MessageBox.Show("印刷完了");  
}  
pd.Print();
```

ページ印刷ごとの  
イベント発生

印刷終了時の  
イベント発生



Microsoft®  
**Silverlight**

**What's New in  
Silverlight 4 Beta**

*Silverlight 4 Beta Technical Features*

# リッチテキスト

## Silverlight 4

- RichTextArea クラス
- 書式付テキスト, Hyperlink, 画像の混在表示
- Hyperlink 使用 : RichTextArea.IsReadOnly = True

```
<RichTextArea VerticalScrollBarVisibility="Auto">
```

```
<Paragraph>
```

```
    A RichTextArea with <Bold>initial  
content</Bold> in it.
```

```
</Paragraph>
```

```
</RichTextArea>
```

*Lorem ipsum*  dolor sit amet, consectetur adipiscing elit. Integer mattis ligula in neque semper bibendum. **Morbi imperdiet**, neque et consectetur sollicitudin, sem arcu egestas diam, sit amet consequat odio velit id tortor. **Quisque congue nisl sed nunc sagittis dictum. Vivamus condimentum urna lacus.** *Suspendisse* porttitor pretium quam ut accumsan. Curabitur vulputate, felis non fringilla vehicula, turpis neque vulputate lectus, *id faucibus turpis ante quis urna. Quisque sem tellus, sodales a interdum in, lacinia ut quam.*



FirstName	LastName	Address
A.	Zero	12 North Third Street, .
B.	One	34 West Fifth Street, A
C.	Two	56 East Seventh Street

# クリップボードの利用

## Silverlight 4

//オブジェクトからのコピー

```
Clipboard.SetText(rtb.Selection.Text);
```

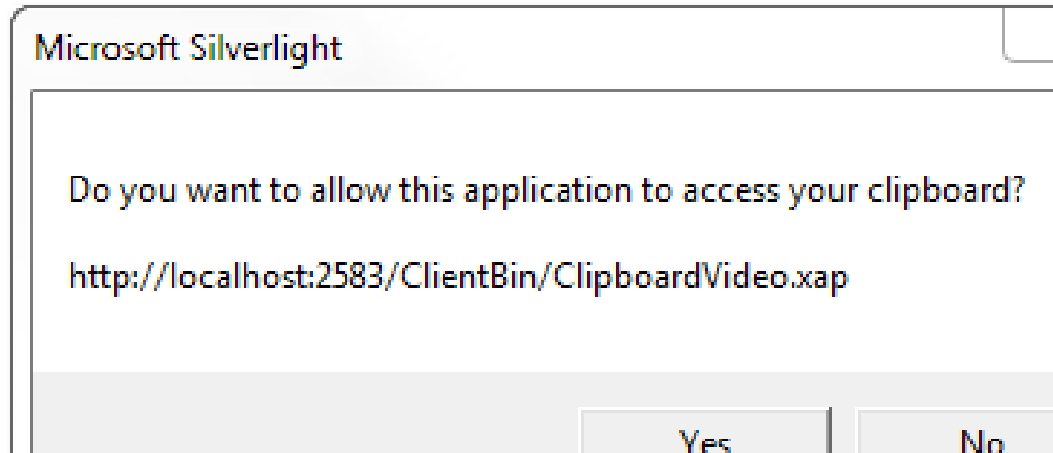
//オブジェクトへのペースト

```
rtb.Selection.Text = Clipboard.GetText();
```

//切り取りの実装

```
Clipboard.SetText(rtb.Selection.Text);
```

```
rtb.Selection.Text = "";
```



# マウスホイールの利用

## Silverlight 4

```
private void OnMouseWheel(  
    object sender, MouseEventArgs e)  
{  
    if (e.Delta > 0)  
        slider_X.Value += slider_X.LargeChange;  
    else  
        slider_X.Value -= slider_X.LargeChange;  
}
```



# 右クリックメニュー

## Silverlight 4

```
public MainPage()
{
    mediaElement.MouseRightButtonDown += new
        MouseButtonEventHandler(MouseRightButtonDownHandler);
}

private void MouseRightButtonDownHandler(object sender, ...
{
    e.Handled = true;
}

private void MouseRightButtonUpHandler(object sender, ...
{
    var menu = new MyCustomMenuControl(mediaElement);
    menu.Show(e.GetPosition(LayoutRoot));
}
```



# IDataErrorInfo でエラー検出

## Silverlight 4

```
public interface IDataErrorInfo
{
    string Error { get; }
    string this[string columnName] { get; }
}
```

```
public class Employee : INotifyPropertyChanged, IDataErrorInfo
{
    public string Error
    {
        get { return errors; }
    }
}
```

First Name:



Firstname's length has to be at least 5 characters!

```
<TextBox x:Name="FirstNameTextBox"
    Text="{Binding Path= ... ValidatesOnDataErrors=True}" />
```



# 自動列幅調整 DataGrid

## Silverlight 4

```
<data:DataGrid AutoGenerateColumns="False"
  x:Name="dataGrid2" HeadersVisibility="All">
  <data:DataGrid.Columns>
    <data:DataGridTextColumn Header="Name"
      Binding="{Binding Name}" Width="*" />
    <data:DataGridTextColumn Header="Address"
      Binding="{Binding Address}" Width="2*" />
    <data:DataGridTextColumn Header="City"
      Binding="{Binding City}" Width="100" />
    :
  </data:DataGrid.Columns>
</data:DataGrid>
```

Name	Address	City
------	---------	------

# Web ブラウザコントロール

## Silverlight 4

- Web ブラウザコントロールを使って HTML ソースの表示が可能
- IFrame による別サイト読み込みも可能

```
string html = "<table border='3' cellpadding='3'><tr>" +  
    "<td><font size='36' color='red'>Red</font></td>" +  
    "<td><font size='36' color='blue'>Blue</font></td>" +  
    "</tr></table>";
```

**HtmlContent.NavigateToString(html)**



# ブラウザ外起動

## Silverlight 4

- Trusted モードに対応するには開発時に設定

32x32.png

48 x 48 Icon

48x48.png

128 x 128 Icon

128x128.png

Use GPU Acceleration

Show install menu

Require Elevated Permissions when running Out-of-Browser

### Out-of-Browser Settings

Window Title

OOB Application

Width 640

Height 480

Shortcut name

OOB Application

Application description

Out-Of-Browser Lab Sample Application on your desktop; at home, at work or on t

16 x 16 Icon

16x16.png

32 x 32 Icon

32x32.png

48 x 48 Icon

48x48.png

128 x 128 Icon

128x128.png

Use GPU Acceleration

Show install menu

Require Elevated Permissions when running Out-of-Browser

OK

# ポップアップメッセージ

## Silverlight 4

```
NotificationWindow notify  
    = new NotificationWindow();  
notify.Width = 330;  
notify.Height = 75;  
TextBlock tb = new TextBlock();  
tb.Text = "Sample notification";  
tb.FontSize = 24;  
notify.Content = tb;  
notify.Show(3000);
```



# Web Platform Installer (Web PI)

## コンポーネントやアプリの一括管理ツール

- Microsoft Web Platform のコンポーネントやオープンソースの Web アプリケーションのダウンロードやインストールなどを一括管理する無償ツール
  - IIS, .NET Framework, Visual Web Developer, SQL Server Express ...
  - WordPress, XOOPS Cube Legacy, SugarCRM, DotNetNuke ...
- Web PI 2.0 ダウンロード

<http://www.microsoft.com/web/downloads/platform.aspx>

The image shows a screenshot of the Microsoft Web Platform Installer 2.0 website on the left and its installation window on the right. The website is in Japanese and features a navigation menu with links like 'ホーム', 'プラットフォーム', 'Web アプリケーションギャラリー', 'ブログラムへの加入', 'ジャンプスタート', '最新情報', and 'ダウンロード'. The main content area includes a 'Microsoft Web Platform Installer 2.0' heading, a brief description, and a 'Download' button. The installation window, titled 'Web Platform Installer 2.0', displays a list of components and applications to be installed, including 'Web プラットフォーム' and 'Web アプリケーション'. The window also shows a list of '最新の Web プラットフォームの新しい拡張機能' (Latest Web Platform Extensions) such as 'Advanced Logging 1.0 (英語)', 'Application Request Routing 2.0 RC (英語)', 'Database Manager 1.0', 'Search Engine Optimization Toolkit Beta (英語)', 'URL Rewrite 1.1', and 'Windows Cache Extension 1.0 for PHP 5.2 - Beta (英語)'. At the bottom of the window, there are buttons for 'インストール' (Install) and '終了' (Close).

# WebsiteSpark

## Web 開発会社様 支援プログラム

Web 開発に必要なマイクロソフトの  
製品を 3 年間無償で利用できます!

### プログラム特典:

#### 開発ツール

Visual Studio 2010 Professional x 3

Expression Studio 3 x 1

Expression Web 3 x 2

#### サーバー ソフトウェア (検証、デモ用)

Windows Web Server 2008 R2 x 3

SQL Server 2008 Web Edition x 3

※これらのソフトウェアは MSDN サブスクリプションの  
ダウンロード サイトから提供させていただきます

#### 技術サポート

技術サポートを 2 インシデント無償提供

#### その他

参加企業の露出機会の提供

技術資料、トレーニングマテリアルなどの提供



### 参加要件:

- Web 開発ビジネスを主業務としていること (個人事業主を含む)
- 従業員数が 25 名以下であること
- プログラム加盟後、6 カ月以内に Windows プラットフォームを用いた新しいドメインの Web サイトを開発すること。

期間 : 最大 3 年間

費用 : 無料 (プログラム終了時にプログラム提供料として 100 米ドルお支払いいただきます)

[www.microsoft.com/web/websitespark](http://www.microsoft.com/web/websitespark)

# 関連リソース (1)

- ✓ MSDN ASP.NET デベロッパー センター  
<http://msdn.microsoft.com/ja-jp/asp.net/default.aspx>
- ✓ The Official Microsoft ASP.NET Site  
<http://www.asp.net/>
- ✓ CodePlex : ASP.NET  
<http://aspnet.codeplex.com/>
- ✓ Microsoft Web Platform  
<http://www.microsoft.com/web/>
- ✓ jQuery  
<http://jquery.com/>
- ✓ Microsoft Web 開発ガイドライン **New!**  
<http://msdn.microsoft.com/ja-jp/asp.net/ff602016.aspx>
- ✓ .NET Framework 4 技術資料一覧  
<http://msdn.microsoft.com/ja-jp/netframework/ee959223.aspx>

ASP.NET MVC 2  
サンプルアプリケーション  
ソースコードは  
[edtter.codeplex.com](http://edtter.codeplex.com)  
で公開中です！

# 関連リソース (2)

- ✓ Silverlight 製品サイト  
<http://www.microsoft.com/japan/silverlight/default.aspx>
- ✓ MSDN Silverlight デベロッパー センター  
<http://msdn.microsoft.com/ja-jp/silverlight/default.aspx>
- ✓ オフィシャル Silverlight コミュニティサイト  
<http://silverlight.net/>
- ✓ Silverlight 4 Training Kit  
<http://www.microsoft.com/downloads/details.aspx?FamilyID=24cea29e-042e-41c9-aa16-684a0ca5f5db&DisplayLang=en>



# 関連リソース (3)

## ✓ 講師のブログ

✓ 荒井 省三 : <http://blogs.msdn.com/shozoa/>

✓ 井上 章 : <http://blogs.msdn.com/chack/>

✓ 物江 修 : <http://blogs.msdn.com/osamum/>

✓ 井上 大輔 : <http://blogs.msdn.com/daisukei/>

## ✓ Future Technology Days

<http://www.microsoft.com/japan/powerpro/ftd/>

## ✓ Tech・Days 2010 テクニカル セッション ストリーミング

<http://www.microsoft.com/japan/events/techdays/2010/session/Download.aspx>

# *Microsoft*<sup>®</sup>



**Future  
Technology Days**

Microsoft<sup>®</sup>  
**Visual Studio**