



IE8 Beta2 紹介セミナー

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
Next Web推進部 エバンジェリスト
渡辺弘之

Agenda

- Internet Explorer 8 (IE8) とは
 - IE8 の特徴
- IE8 互換性とその対策
 - 相互運用性
 - 互換性・・・3つのレンダリングモード
 - IE8 レンダリングモード対応方法
- IE8 新機能について
 - アクセラレータ
 - WebSlice
 - その他の新機能

Internet Explorer 8 (IE8) とは

Internet Explorer 8 の特徴



より速く

+



より便利に

+



より安心に



Internet Explorer 8 の特徴



より速く

+



より便利に

+



より安心に



高速化のための様々な改善

起動処理の
高速化

HTML 解析
の高速化

使用メモリ
量の軽減

スクリプト
処理の高速化

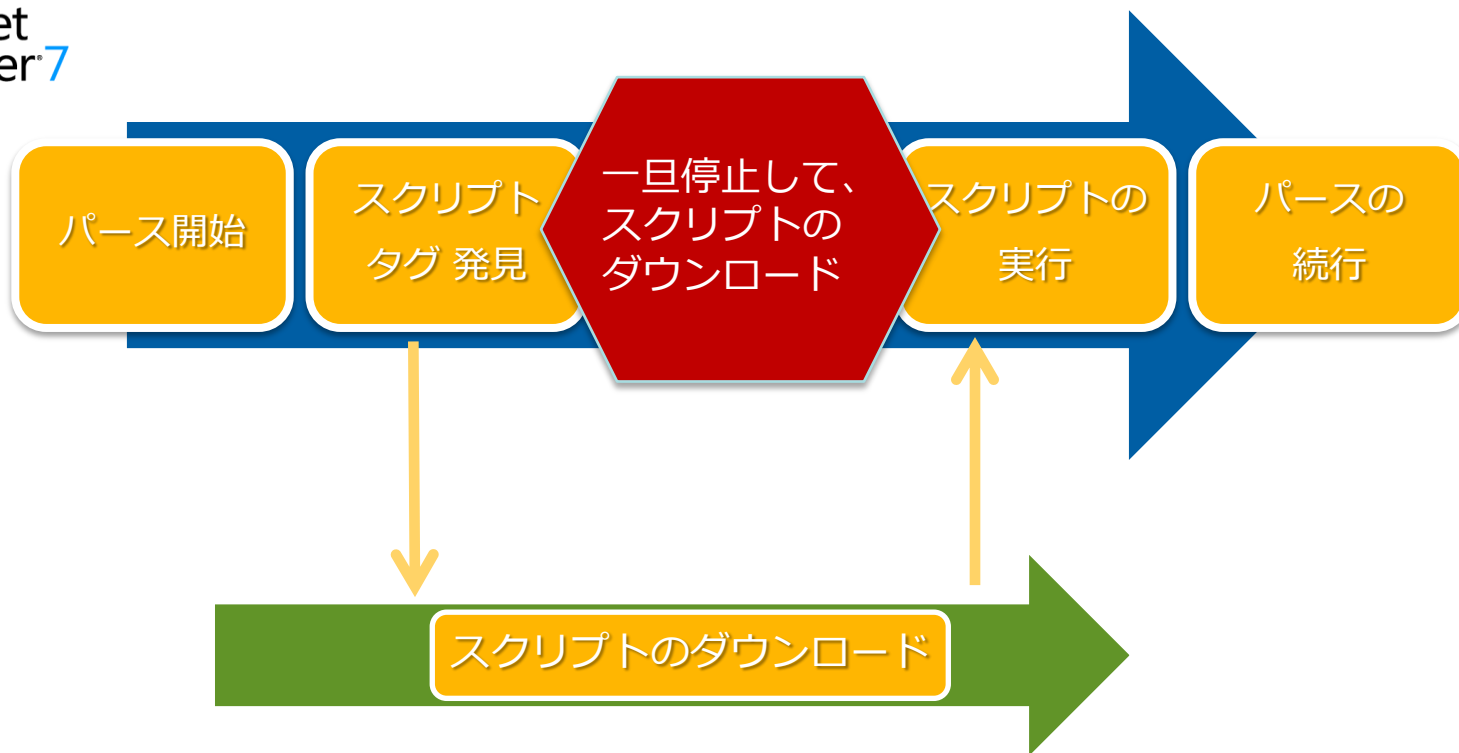
ダウンロー
ドの高速化



HTML 描画速度の向上



HTML
パーサー

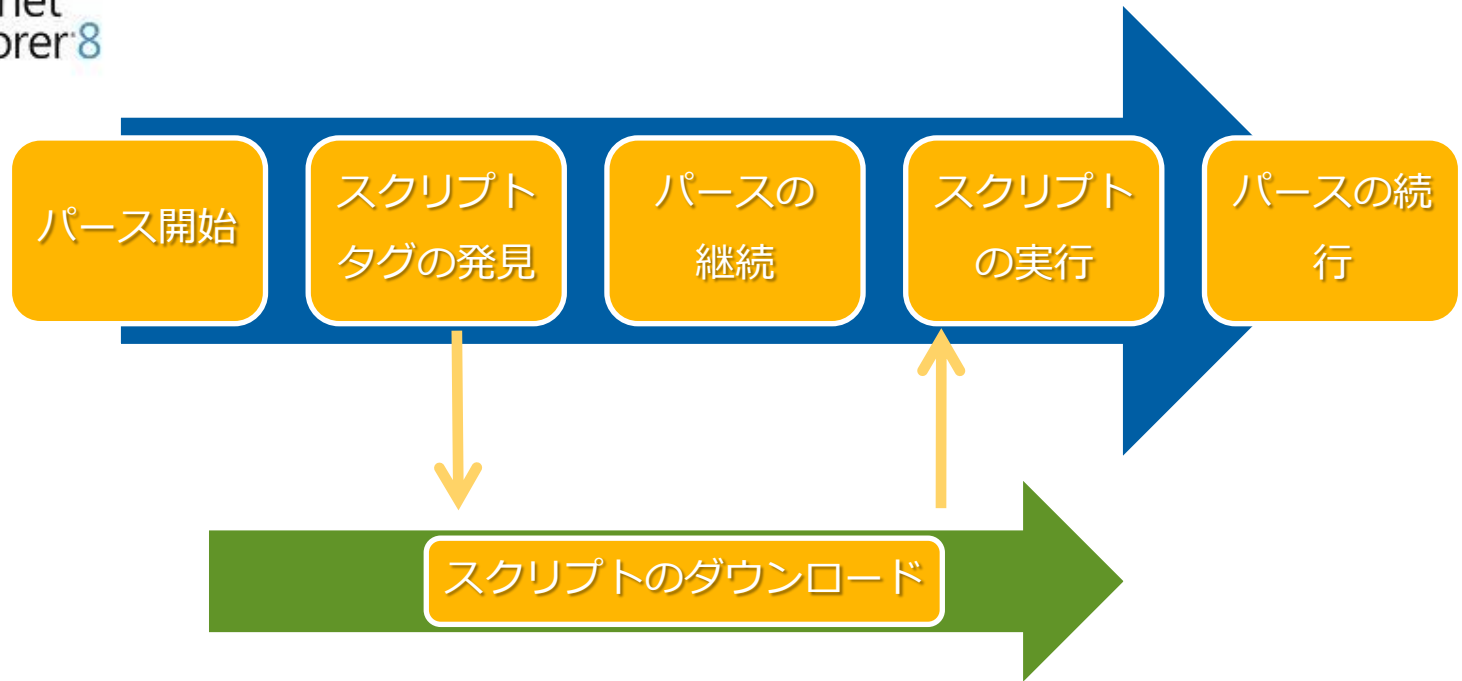


スクリプトのダウンロード終了まで待機

HTML 描画速度の向上

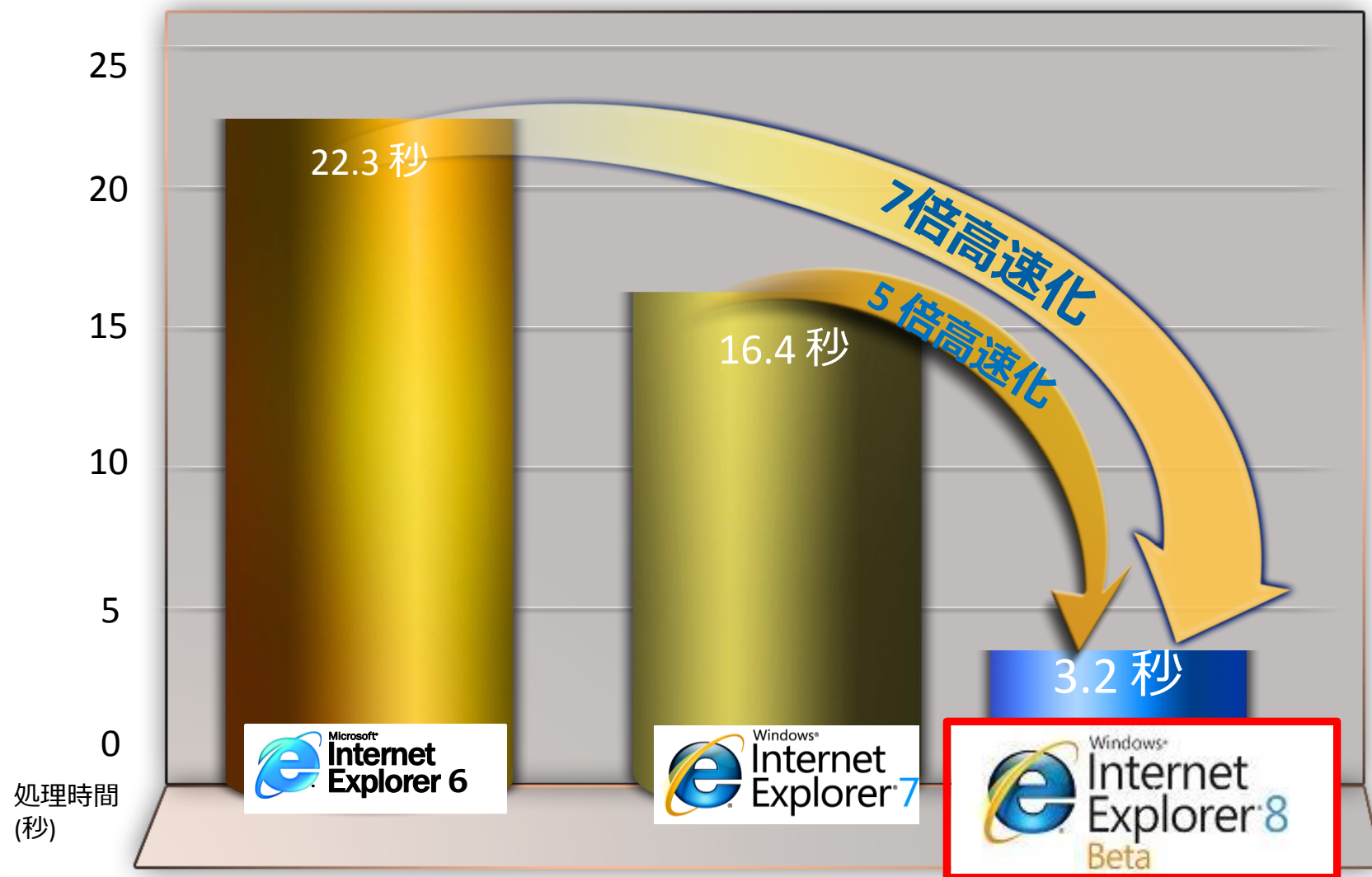


HTML
パーサー



スクリプトエンジンの実行中も、パースを続行

スクリプト処理速度の向上



※Sun Spider JavaScript BenchMark による測定
検証環境：CPU-Pentium 4 Processor 2.8Ghz, Memory- 2GB,
複数回の検証による平均値

Internet Explorer 8 の特徴



より速く

+



より便利に

+



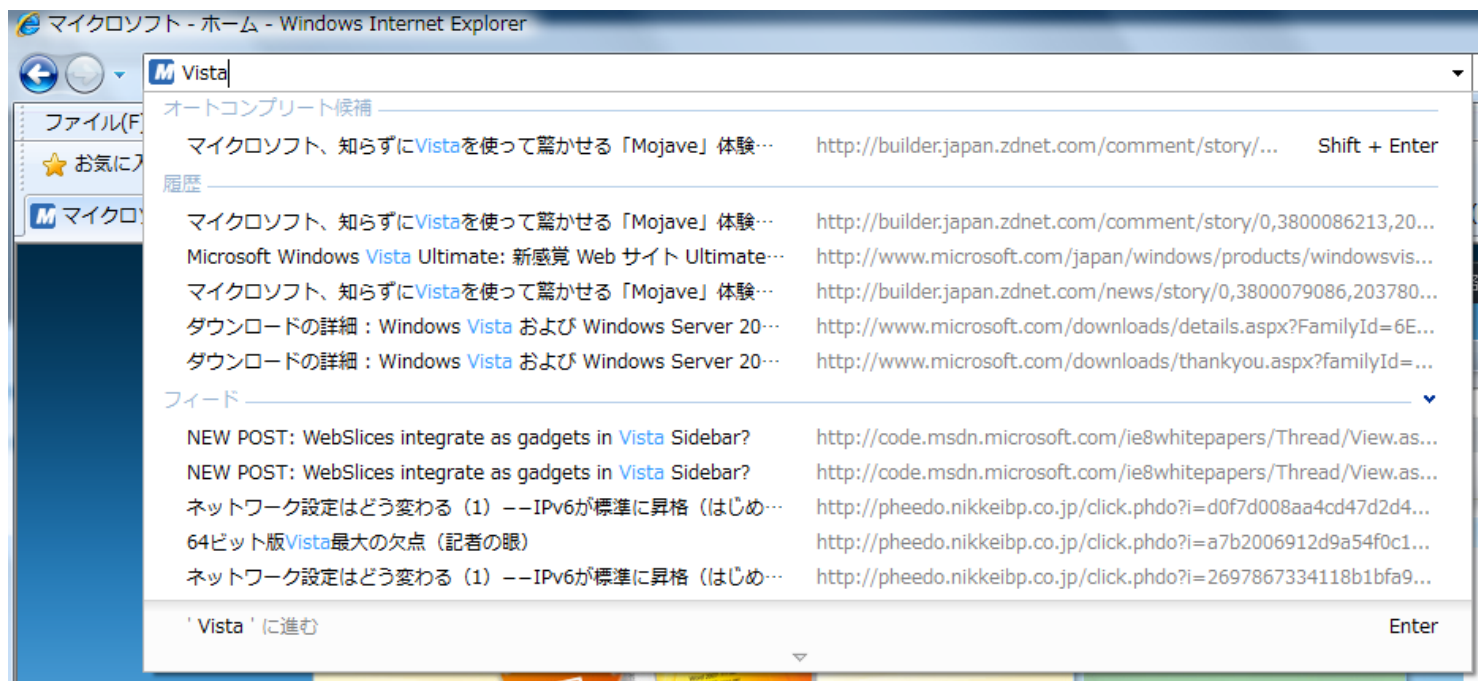
より安心に



ブラウザの操作を簡単にする新機能

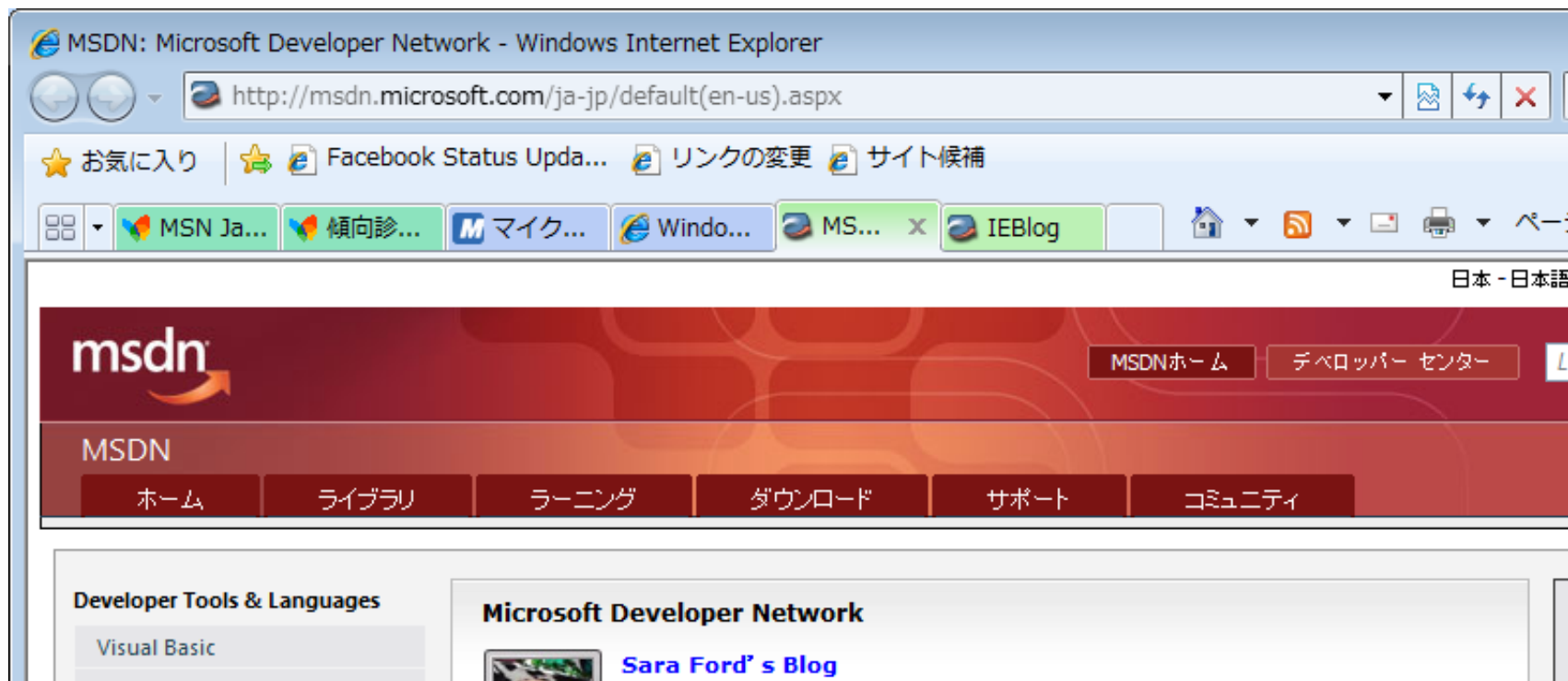
お気に入り・履歴と融合した「アドレスバー」

「アドレスバー」にホスト名やキーワードを入力するだけで、「オートコンプリート」・「履歴」・「お気に入り」から検索を行ない、該当するサイトを一覧で表示します。



ブラウザの操作を簡単にする新機能

たくさんのタブを一目で把握する「タブグループ」
タブの親子関係を色で区別してグループ表示します。



ブラウザの操作を簡単にする新機能

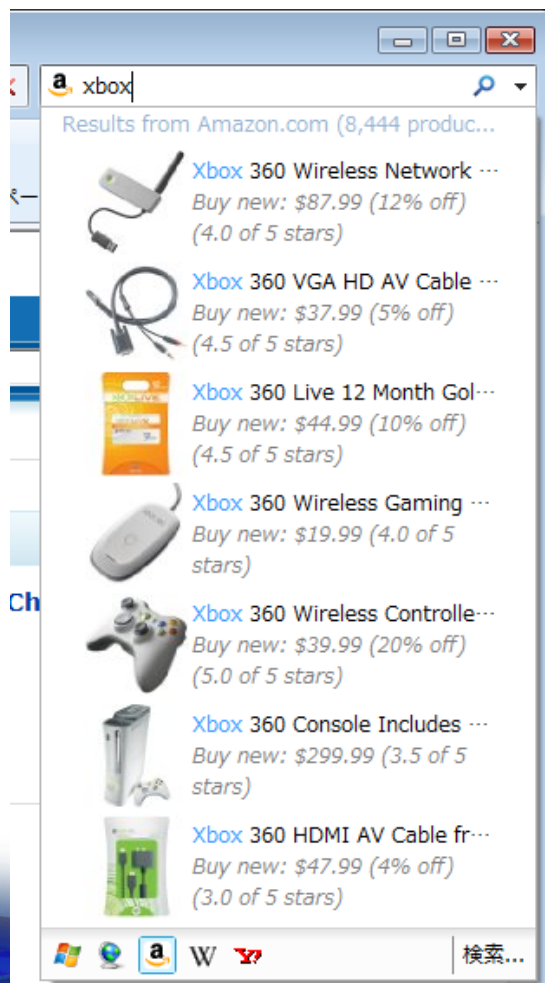
次の操作がすぐに見つかる「新しいタブ」



ブラウジングをより便利にする新機能

検索候補の表示

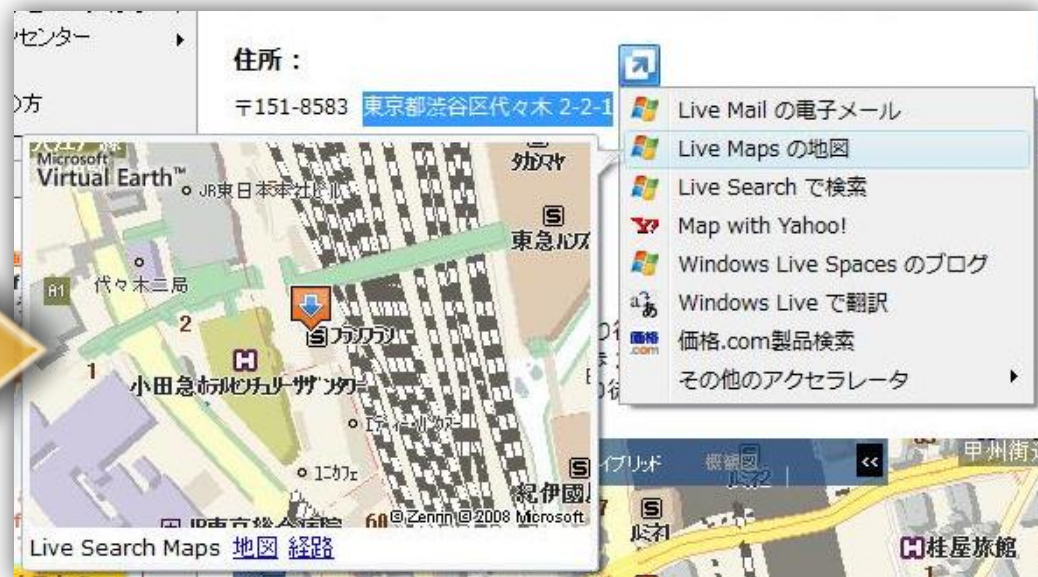
検索ボックスに入力された文字列から、検索候補文字列や検索結果候補を表示



ブラウジングをより便利にする新機能

アクセラレータ

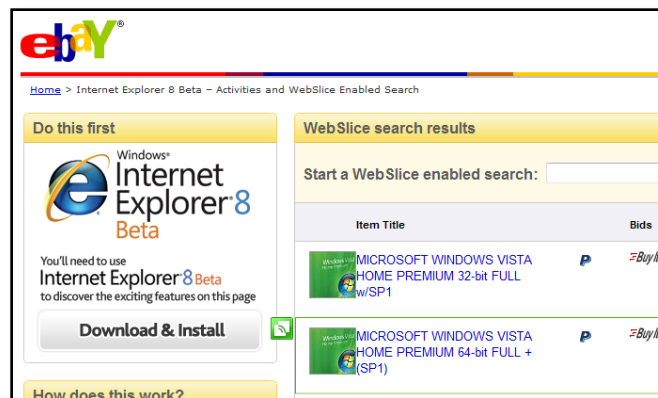
ユーザーを今見ているサイトから既存のサービスへ直接結び付けます



ブラウジングをより便利にする新機能

WebSlice

気になる情報を切り取って保存。いつでもすぐに最新情報を確認可能



Internet Explorer 8 の特徴



より速く

+



より便利に

+



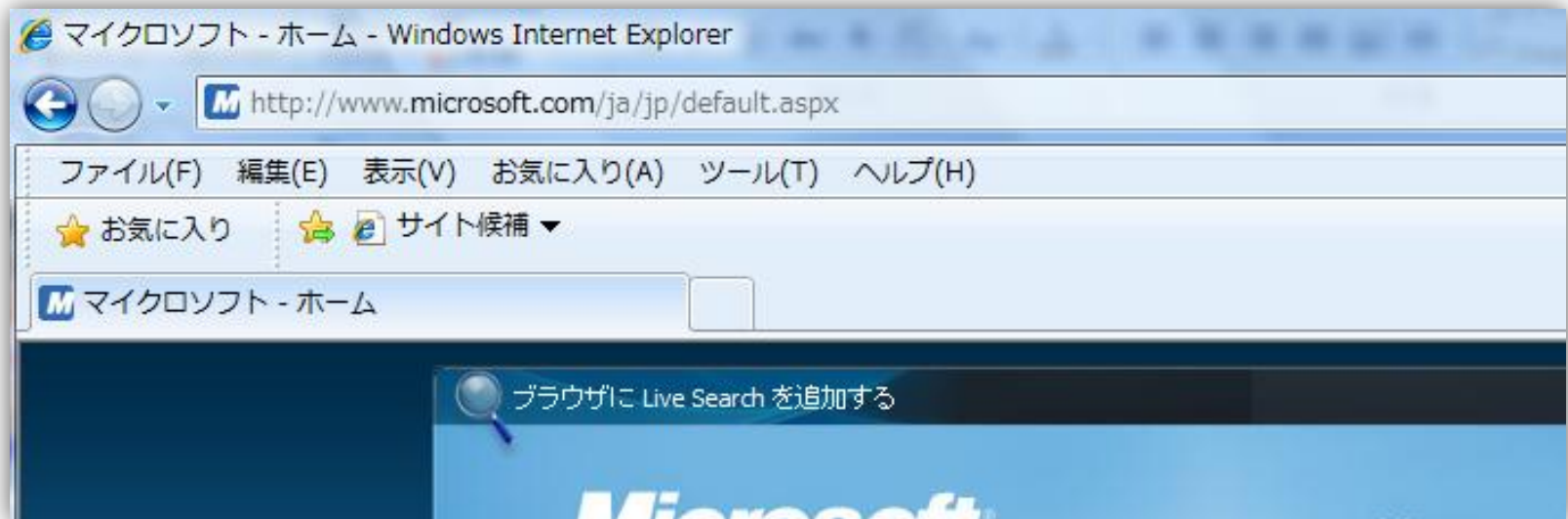
より安心に



インターネットを安全に利用するための新機能

ドメインハイライト

アドレスバーのドメイン名を見やすく表示。
なりすましを目的にした Web サイトを簡単に判別。

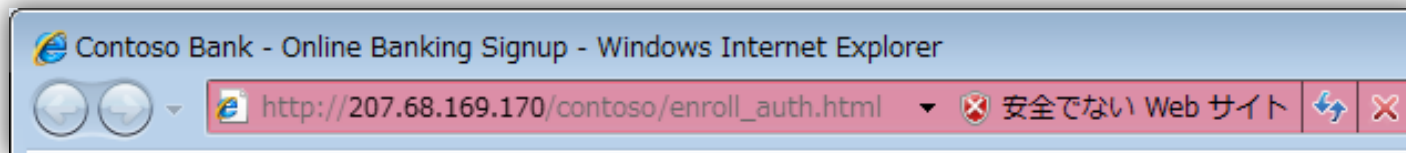


インターネットを安全に利用するための新機能

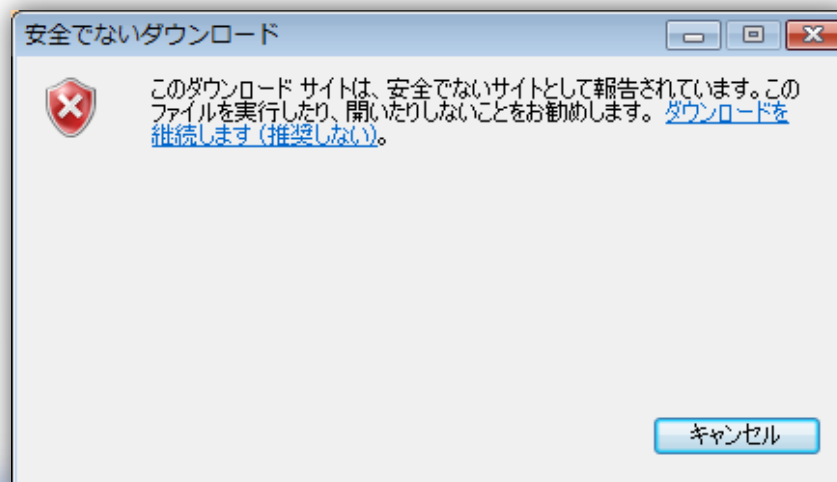
SmartScreenフィルター

フィッシング詐欺の疑いがあるアドレスへの警告に加え、マルウェアのダウンロードをブロックします。

フィッシングサイト警告時



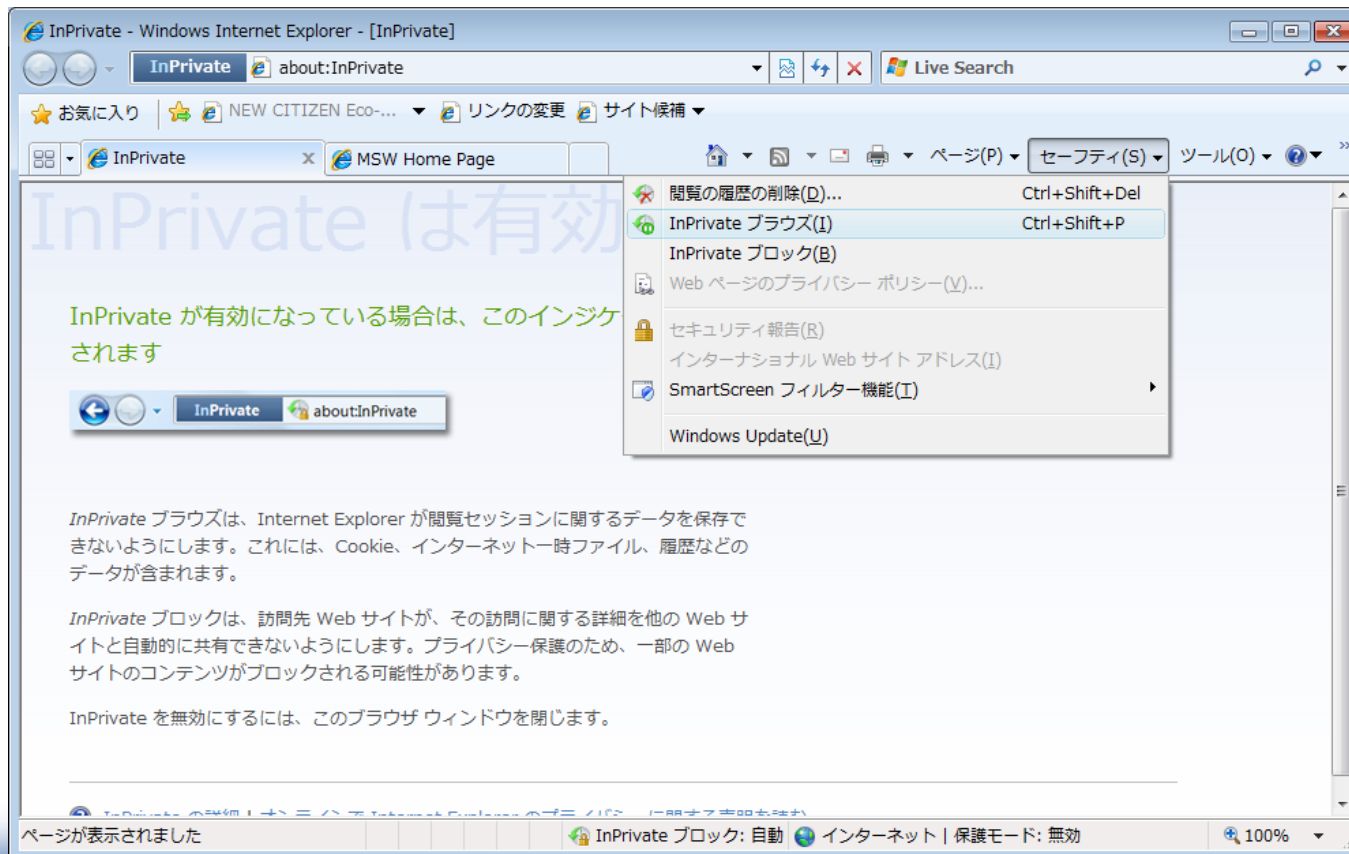
マルウェアブロック通知



インターネットを安全に利用するための新機能

InPrivateブラウズ

Cookie、履歴、キャッシュ(Temporary Internet Files) などの使用情報をコンピュータに記録せずにブラウジングを行ないます。



インターネットを安全に利用するための新機能

IE7 では、整合性レベルが異なるページは、別ウィンドウで表示



ieuser.exe

ブローカー
オブジェクト

ウィンドウとタブを
表示

整合性レベル：中

整合性レベル：低

ieexplorer.exe (保護モード ON)

UI フレーム

タブ 1

...

タブ n

ツールバー拡張

ブラウザー ヘルパー
オブジェクト

ActiveX コントロール

ieexplorer.exe (保護モード OFF)

UI フレーム

タブ 1

...

タブ n

ツールバー拡張

ブラウザー ヘルパー
オブジェクト

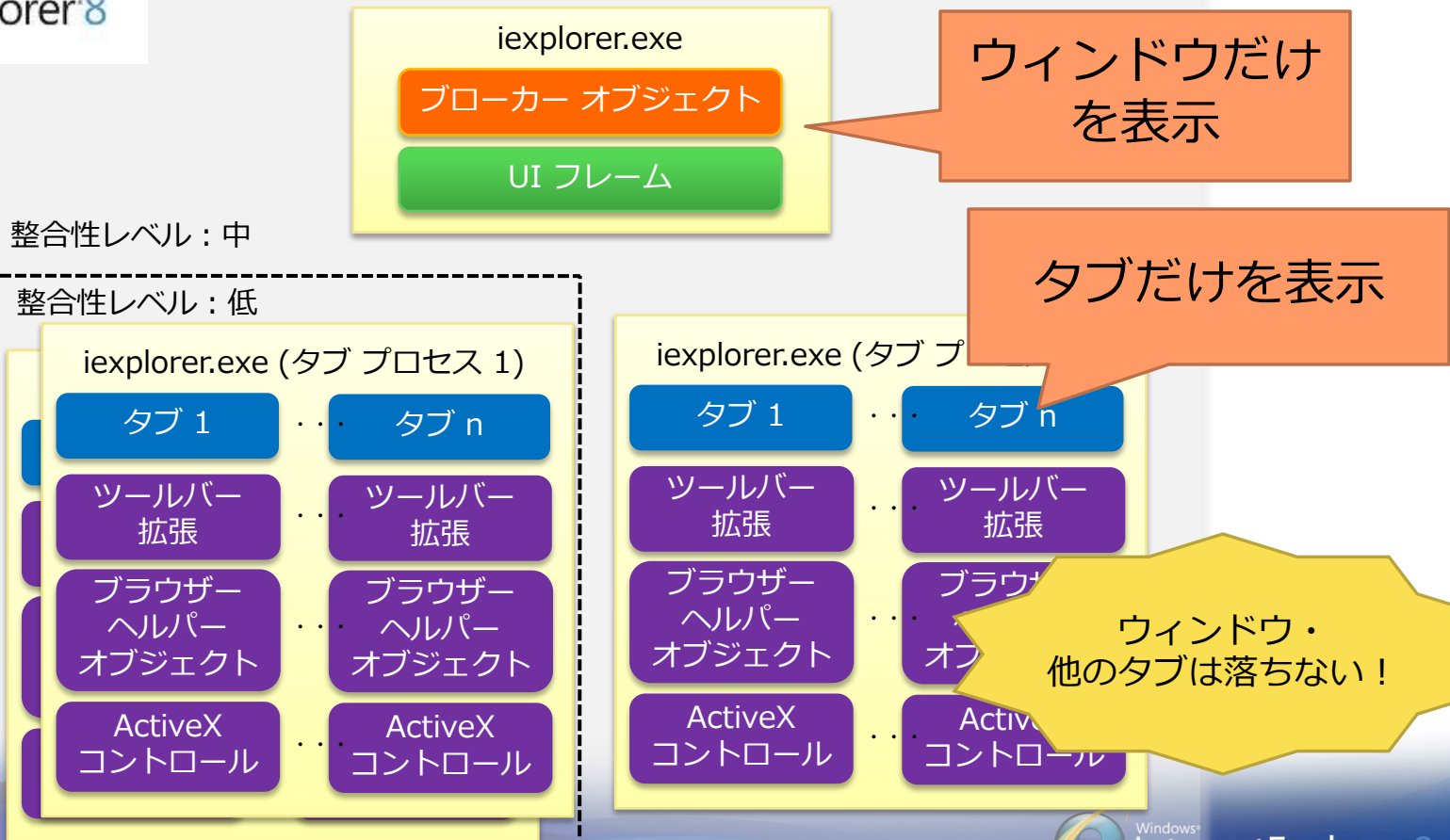
ActiveX コントロール

インターネットを安全に利用するための新機能

IE8（+Vista）では、フレームとタブを分離

保護モードの On/Off のサイトの同一ウィンドウ内の共存が可能

表示ページ内で異常が発生した場合にタブだけ終了させることが可能



Internet Explorer 8 の特徴



より速く

+



より便利に

+



より安心に



IE 8 Beta 版の位置づけ

- Internet Explorer 8 Beta 1
 - 3/5公開、英語版・ドイツ語版・中国語版のみ提供
 - 日本語Windows環境（XP SP2以降）へのインストール可能
 - Webデザイナー・Web開発者向けのコア機能評価用として提供
- Internet Explorer 8 Beta 2
 - 8/28公開、日本語版を含む20数カ国版提供
 - 外観も含めて、IE8 RTW版で提供予定のフル機能を実装
 - Webデザイナー・Web開発者だけでなく、先進的なユーザーもIE8の新機能を体感可能

IE 8 Beta 2 インストール要件

対象 オペレーティング システム

- Windows Vista x64 及び x86
- Windows Vista Service Pack 1 (SP1) x64及びx86
- Windows XP Service Pack 2 (SP2)
- Windows XP Service Pack 3 (SP3)
- Windows XP Professional x64 Edition
- Windows Server 2008 x64及びx86
- Windows Server 2003 Service Pack 2 (SP2) x64及びx86

インストールに必要な更新プログラム

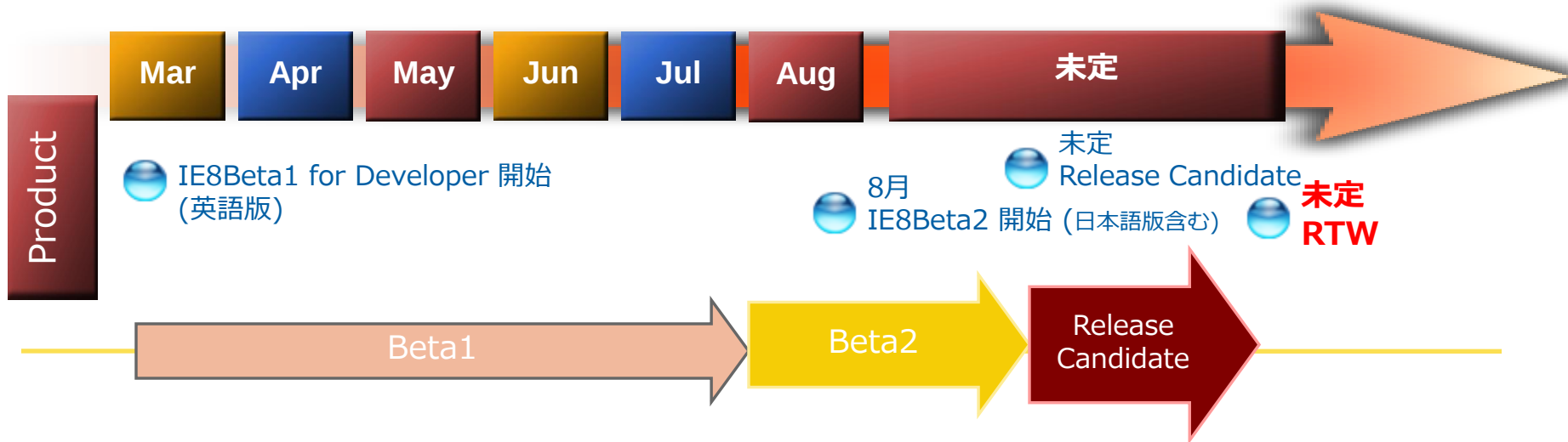
KB943302

この更新は **Windows Vista RTM 版のインストール のみに必要**です。
またこの更新は、IE8 のセットアップウィザードの中で、"最新のアップデートをインストールする" のチェックをオフにしない限り、自動的にインストールされます。IE8 を起動した時にこの更新がインストールされていない場合、手動でこの更新をインストールするようにプロンプトが表示されます。

KB946501

この更新は **multi-core Windows XP SP2 x86 の PC にのみ必要**です。
上記と同様に、この更新は IE8 のセットアップウィザードの中で、"最新のアップデートをインストールする" のチェックをオフにしない限り、自動的にインストールされます。
もしこの更新のインストールに失敗していたり、チェックがオフになっている場合、IE8 のインストールはできません。

IE 8 リリース スケジュール



対象 OS: Windows XP, Windows Vista,
Windows Server 2003, Windows Server 2008

IE8 互換性とその対策

相互運用性



Web標準に準拠したレンダリングモードをデフォルトで採用

CSS 2.1 サポート
HTML4.01 サポート

他のWebブラウザとの相互運用性の向上

CSSについては、
W3CへCSS Test Suiteを提案

相互運用性を重視するが、相互運用性と互換性のバランスも大切

互換性 3つのレンダリングモード

IE8 のレイアウトエンジンは、
「**Quirksモード**」、「**IE7標準モード**」、「**IE8標準モード**」
の3つのレンダリングモードをサポート
以前のInternet Explorerとの互換性を確保

レンダリングモード

IE <= 6

Quirksモード

IE 7

IE7 標準モード

デフォルト

IE 8

IE8標準モード

IE7レイアウト

IE8レイアウト

IE8レンダリングモード対応方法

IE8対応の必要性あり

IE7 で正常表示

Yes

No

ページの
IE7 対応化可能

Yes

No

別に標準準拠の
ページがある

Yes

No

対応方法3:
Web標準準拠のページを
新規作成

対応方法1:

コンテンツ又はサーバーの設定を行い、IE7標準モードのレンダリングモード指定

レンダリングモード指定例 :

- ・コンテンツ側で対応する場合

Meta タグ :

```
<meta http-equiv=
"X-UA-Compatible"
content="IE=EmulateIE7" >
```

- ・Webサーバー側で対応する場合

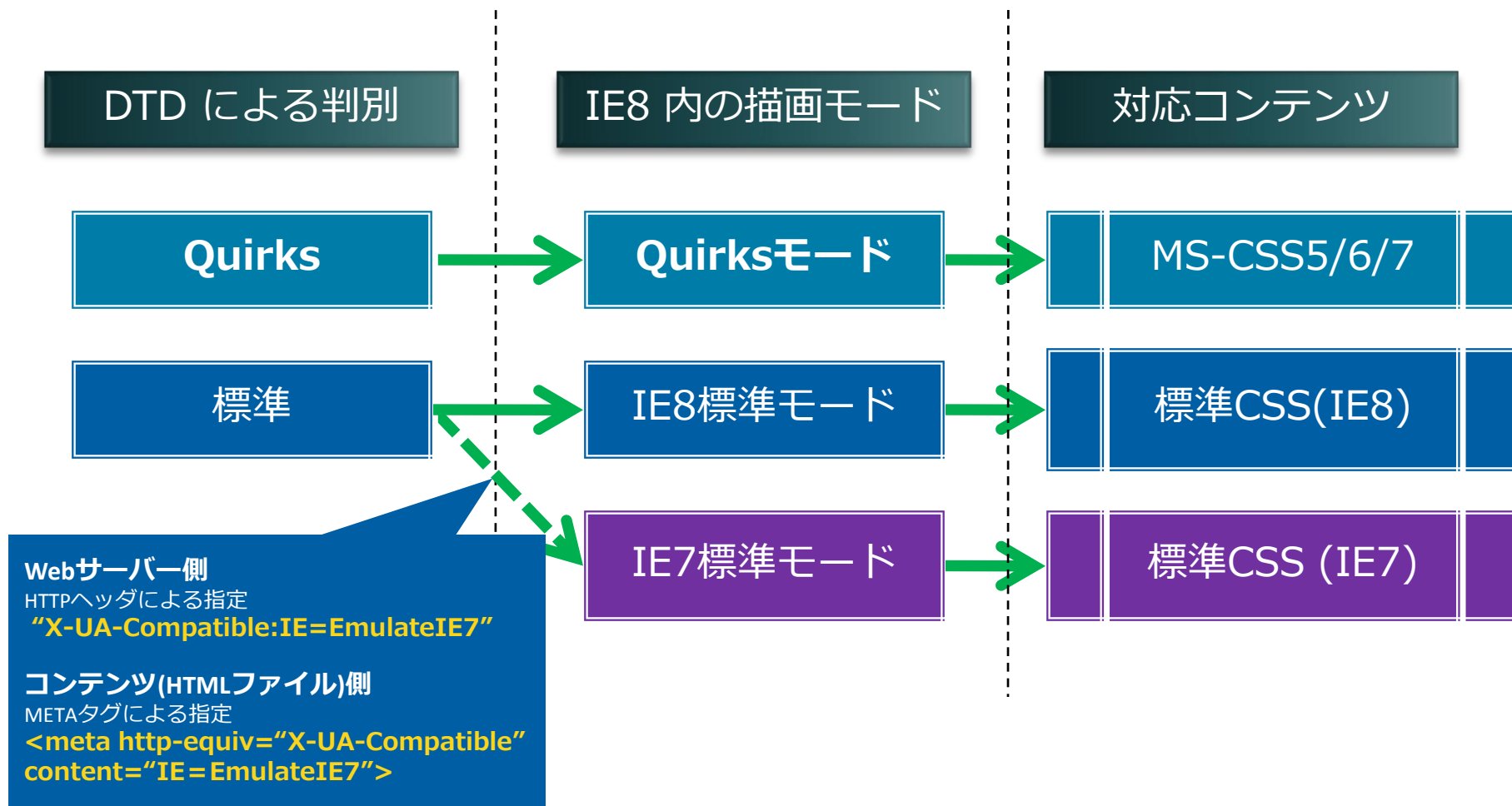
HTTP ヘッダ :

```
"X-UA-Compatible: IE=EmulateIE7"
```

対応方法2:

IE8 は標準準拠ページへ誘導するようにサーバーサイドで設定

IE8レンダリングモード対応



IE8のレンダリングモード スイッチ表1

METAタグの有無	DOCTYPEフォームの終わり	DOCTYPE内のURL表記の有無	適用されるレンダリングモード	DOCTYPEの例
Metaタグ無し	DOCTYPE なし	いずれの場合も	Quirksモード	-
	HTML (no version)	いずれの場合も	Quirksモード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML //EN">
	HTML 2.0	いずれの場合も	Quirksモード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 2.0 Final//EN">
	HTML 3.0	いずれの場合も	Quirksモード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
	HTML 4.0	いずれの場合も	IE8 標準モード(CSS 2.1)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
	HTML 4.0 Frameset	URLあり	IE8 標準モード(CSS 2.1)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN" "http://www.w3.org/TR/REC-html40/frameset.dtd">
		URLなし	Quirksモード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
	HTML 4.0 Transitional	URLあり	IE8 標準モード(CSS 2.1)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
		URLなし	Quirksモード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transition// EN">
	HTML 4.0 Strict	いずれの場合も	IE8 標準モード(CSS 2.1)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
	XHTML	いずれの場合も	IE8 標準モード(CSS 2.1)	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN">
	XML	いずれの場合も	IE8 標準モード(CSS 2.1)	<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
	Unrecognized DOCTYPE	いずれの場合も	IE8 標準モード(CSS 2.1)	-
IE8 指定あり (IE=8)	DOCTYPEの有無、内容にかかわらず	いずれの場合も	IE8 標準モード(CSS 2.1)	-
IE7 指定あり (IE=7)	DOCTYPEの有無、内容にかかわらず	いずれの場合も	IE7 標準モード	-
互換設定 (IE=5)	DOCTYPEの有無、内容にかかわらず	いずれの場合も	Quirksモード	-
指定有り (IE=Edge)	DOCTYPEの有無、内容にかかわらず	いずれの場合も	IE8 標準モード(CSS 2.1)	-

IE8のレンダリングモード スイッチ表2

METAタグの有無	DOCTYPEフォームの終わり	DOCTYPE内のURL表記の有無	適用されるレンダリングモード	DOCTYPEの例
IE7標準/ Quirksモード (IE=EmulateIE7)	DOCTYPE なし	いずれの場合も	Quirksモード	-
	HTML (no version)	いずれの場合も	Quirksモード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML //EN">
	HTML 2.0	いずれの場合も	Quirksモード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 2.0 Final//EN">
	HTML 3.0	いずれの場合も	Quirksモード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
	HTML 4.0	いずれの場合も	IE7標準モード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
	HTML 4.0 Frameset	URLあり	IE7標準モード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN" "http://www.w3.org/TR/REC-html40/frameset.dtd">
		URLなし	Quirksモード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
	HTML 4.0 Transitional	URLあり	IE7標準モード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" ""http://www.w3.org/TR/html4/loose.dtd">
		URLなし	Quirksモード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
	HTML 4.0 Strict	いずれの場合も	IE7標準モード	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN" "http://www.w3.org/TR/html4/strict.dtd">
	XHTML	いずれの場合も	IE7標準モード	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
	XML	いずれの場合も	IE7標準モード	<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
	Unrecognized DOCTYPE	いずれの場合も	IE7標準モード	-

META タグによるレンダリングモード指定

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

<html xmlns:web xmlns="http://www.w3.org/1999/xhtml" lang="en"
xml:lang="en" class="liveApp la_en lo_us IE IE_8">
<head>
  <title>Test Page</title>
  <meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
  <meta http-equiv="Content-Type" content="text/html;
    charset=utf-8" />
  <meta name="description" content="Find exactly what you are
    looking for – FAST!" />
  <meta name="ROBOTS" content="NOODP" />
  <link rel="stylesheet" type="text/css"
    ref="stylesheets/standards.css" />
</head>
```

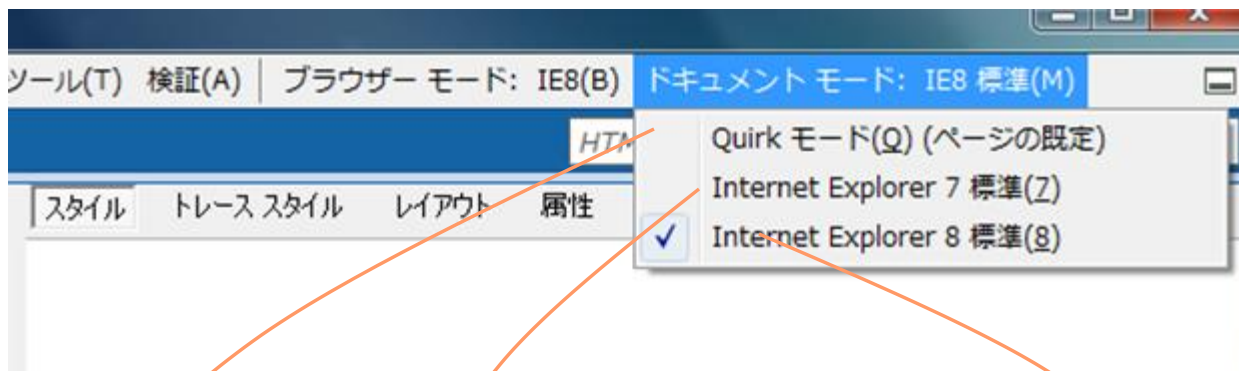
レンダリングモード指定の補足

モード	指定する値
Quirks	IE=5
IE7標準	IE=7
IE8標準	IE=8
常にIE最新モード	IE=edge
標準DOCTYPEはIE7標準, Quirks DOCTYPEはQuirks	IE=EmulateIE7

- » META タグの指定は
HTTPレスポンスヘッダ、
DOCTYPEを上書き
- » 今後もずっと最新のモードを
使う場合は edge を指定

開発者ツールの活用

- 簡単にレンダリングモードを切り替えて確認可能



MODES

IE <= 6

- Quirks モード

IE 7

- IE7標準モード

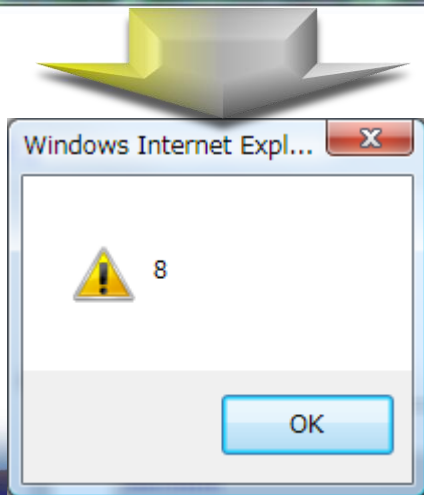
IE 8

- IE8標準モード

レンダリングモードの判定方法

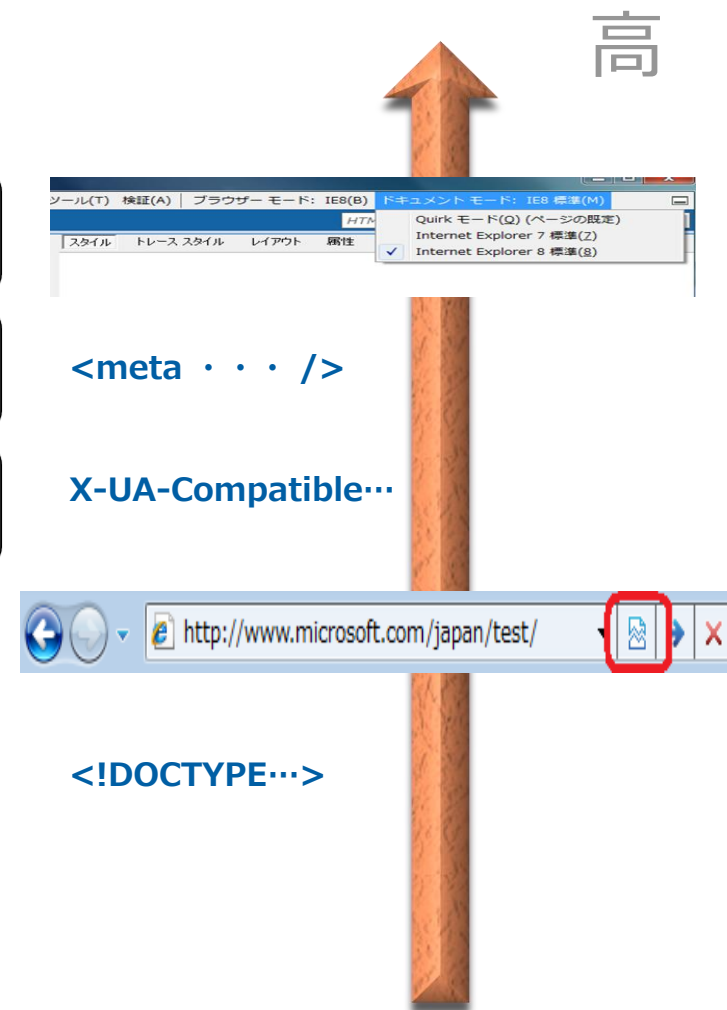
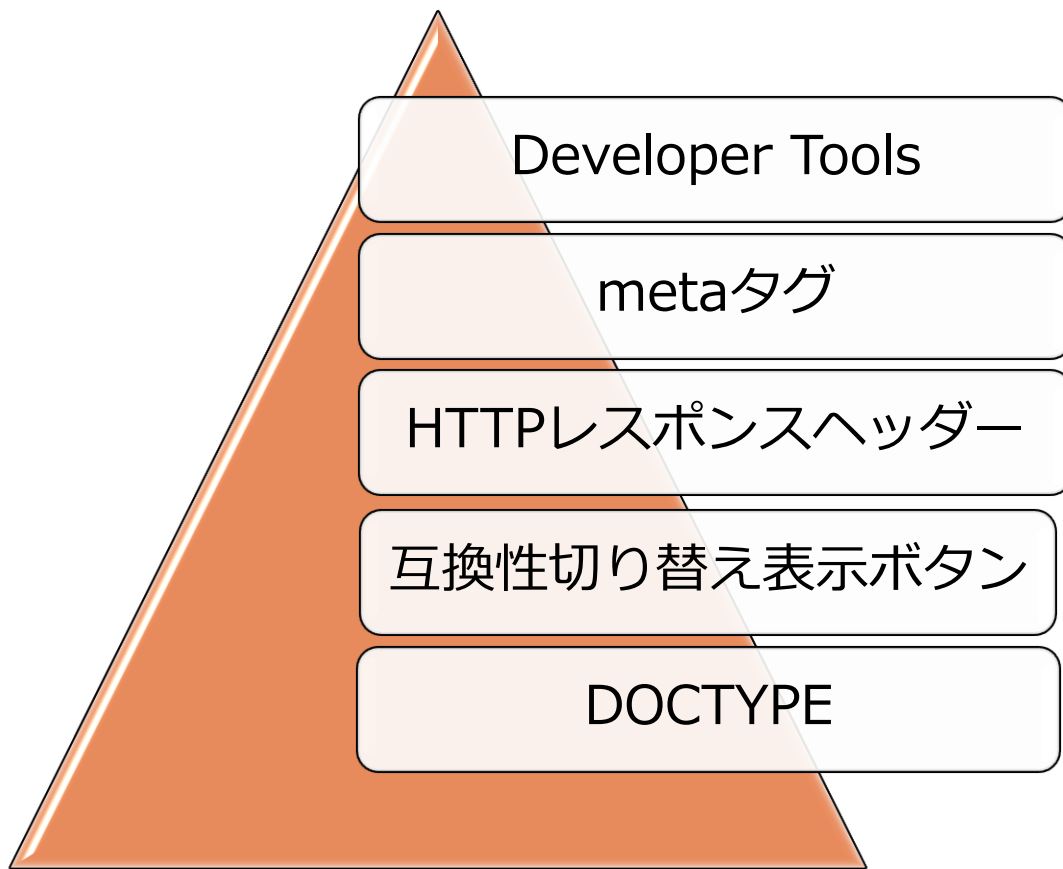
- JavaScriptのdocument.documentModeプロパティにより判定ができる

```
javascript:alert(document.documentMode)
```



モード	表示される値
Quirksモード	5
IE7標準モード	7
IE8標準モード	8

レンダリングモード判定の優先順位



User Agent文字列

- User Agent文字列とは、ブラウザからWebサーバーへ送信する文字列
- 使っているブラウザの種類、バージョン番号、システム詳細（OSのバージョンなど）を表す

例) Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0)

User Agent文字列を利用した判定コード

```
if(ver == 7.0)  
    msg = "Edge Agent";
```

```
if(ver >= 8.0)  
    msg = "Edge Agent";
```

Version Vector

- Version Vectorとは、IEの内部バージョン番号。IE8では、「**8.0**」。
- レジストリキーに格納されており、IE起動時に読み込まれる。
- 条件付きコメントを処理する際に利用される。

Version Vector を利用したCSSの切り替え例：

```
<head>
<title>IE8 Seminar</title>
<meta http-equiv="X-UA-Compatible" content="IE=8" />
<!--[if gte IE 8]>
    <link rel="stylesheet"
        type="text/css" href="ie8.css" />
<![endif]-->
<!--[if IE 7]>
    <link rel="stylesheet"
        type="text/css" href="ie7.css" />
<![endif]-->
</head>
```

メモリ保護オプションの変更 IE向けアドオン開発者が注意すべき点

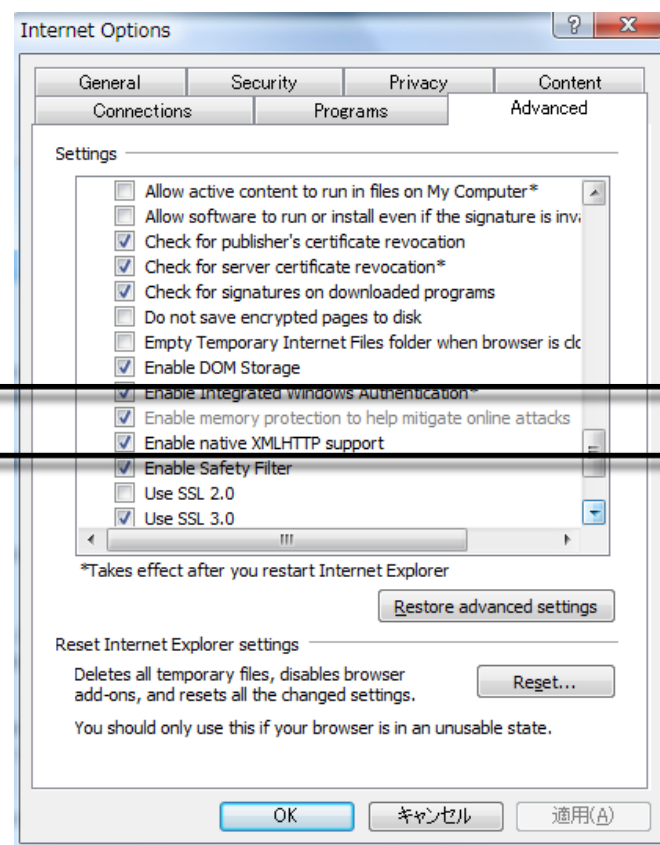
- » バッファオーバーラン
 - 実行不能なデータ型を実行しようとすることで異常終了やセキュリティホールを引き起こす
 - IE の異常終了で最も多い原因の1つ
- » DEP (Data Execution Prevention) 、NX (No-Execute) 機能
 - ActiveX で実行可能なデータのみ実行する構造
 - Vista/WS2008 で、デフォルトでオン。



メモリ保護オプションの変更 IE向けアドオン開発者が注意すべき点

対応策 (Call to Action)

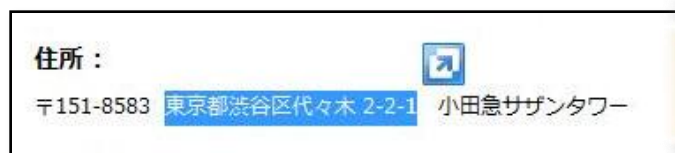
1. ATL (Active Template Library) のバージョン 7.1 SP1 以降でリビルド
(VS2005 には、ATL 8.0 が含まれる)
2. リンカオプションに “/NXCompat” を加える
3. IE8 Beta1 (DEP/NX 機能オン) でテスト
4. その他オプションによる最適化
/GS (スタック呼び出し制御)、
/SafeSEH (安全な例外処理)、
/DynamicBase (ASLR: Address Space Layout Randomization)



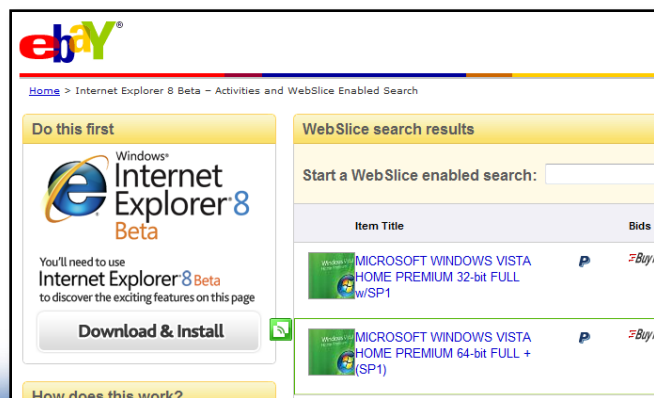
IE8 新機能について アクセラレータ WebSlice

オンラインサービスの操作性を向上させる機能

- 「アクセラレータ」
選択したページ内の文字列をキーワードに、ページを移動せずにWeb アプリを利用可能



- 「WebSlice」
Web ページの一部を切り取って保存、同じページを表示しなくてもいつでも最新情報を確認可能



アクセラレータ

Webページ上のコンテンツと様々なオンラインサービスを
シンプルな操作で連携させる機能

簡単に実装可能（クライアント側の変更は必要ない）





Demo

アクセラレータの実装 OpenService フォーマット

```
<?xml version="1.0" encoding="utf-8" ?>
<openServiceDescription
  xmlns="http://www.microsoft.com/schemas/openservicedescription/1.0">
  <homepageUrl>http://maps.live.com</homepageUrl>
  <display>
    <name>Live Mapsの地図</name>
    <icon>http://maps.live.com/favicon.ico</icon>
  </display>
  <activity category="Map">
    <activityAction context="selection">
      <execute method="get"
        action="http://maps.live.com/default.aspx?where1={selection}" />
      <preview method="get" action="http://maps.live.com/geotager.aspx">
        <parameter name="b" value="{selection}" />
        <parameter name="clean" value="true" />
        <parameter name="w" value="320" />
        <parameter name="h" value="240" />
        <parameter name="format" value="full" />
      </preview>
    </activityAction>
  </activity>
</openServiceDescription>
```



アクセラレータの実装

OpenService フォーマット

» アクセラレータのXML 名前空間

```
<openServiceDescription  
  xmlns="http://www.microsoft.com/schemas/openservicedescription/1.0">
```

» homepageUrl : アクセラレータによってアクセスするWebサービスのURL

```
<homepageUrl>http://maps.live.com</homepageUrl>
```

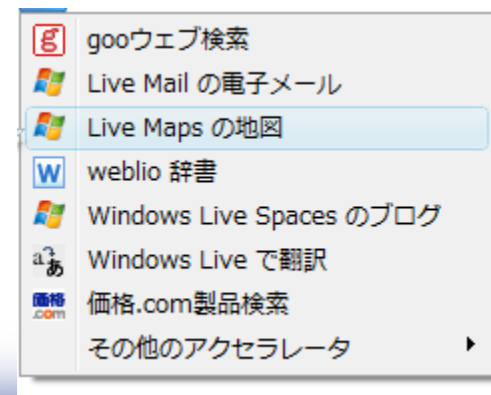
» diplay : アクセラレータのメニューに表示される項目

```
<display>
```

```
  <name>Live Mapsの地図</name>
```

```
  <icon>http://maps.live.com/favicon.ico</icon>
```

```
</display>
```



アクセラレータの実装

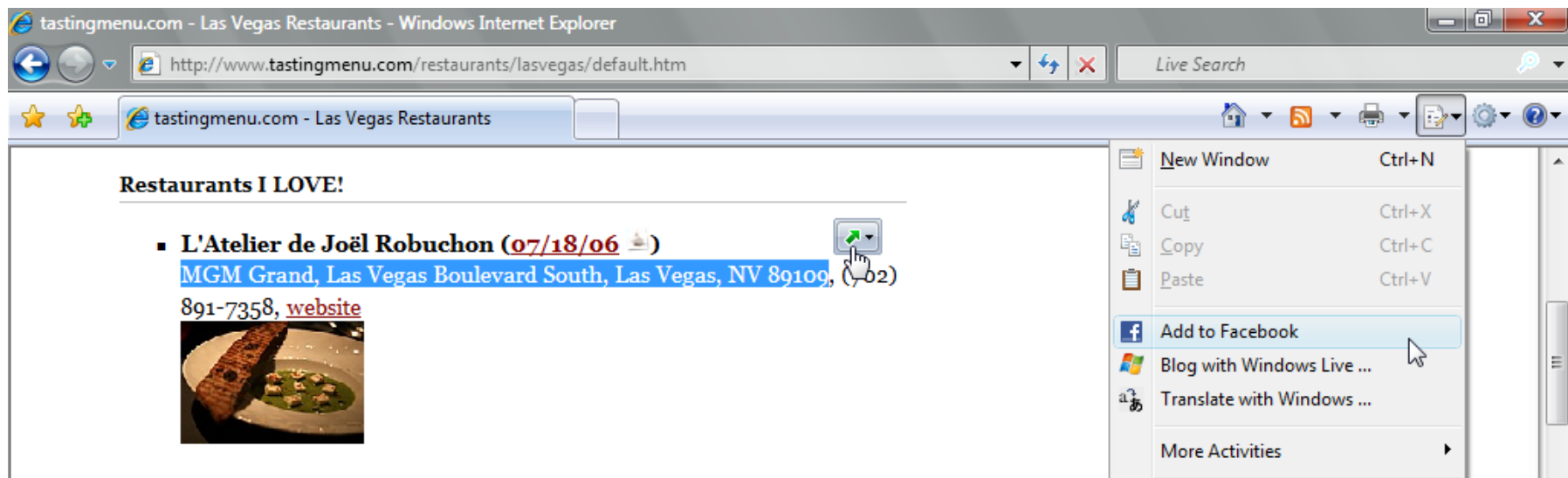
OpenService フォーマット

- » activity : アクセラレータの動作を設定
 - category : アクセラレータが提供する機能によりグループ分け
 - Map: Windows Live Map, Google Maps
 - Blog: Windows Live Spaces, Blogger
 - Define: Encarta, Wikipedia
 - Add: Del.icio.us
 - . . .

<activity category="**Map**">

アクセラレータの実装 OpenService フォーマット

- » context : アクセラレータが動作する際のデータタイプを指定
 - デフォルト「selection」
 - 「selection」、「document」、「link」指定可能
- <activityAction context="selection">



アクセラレータの実装

OpenService フォーマット

» preview : ユーザーがアクセラレータを選択したときに表示される小型ウィンドウ (マウスカーソルオンの時の動作)

methodは、「get」 or 「post」 (デフォルト「get」)

actionで指定するURIに対してインラインパラメータ

(この場合、where1={selection}) で変数を送信、または、

フォームベースパラメータで指定した変数

(nameとvalueで指定) を送信

```
<preview method="get" action="http://maps.live.com/geotager.aspx">
```

```
<parameter name="b" value="{selection}" />
```

```
<parameter name="clean" value="true" />
```

```
<parameter name="w" value="320" />
```

```
<parameter name="h" value="240" />
```

```
<parameter name="format" value="full" />
```

```
</preview>
```



アクセラレータの実装

OpenService フォーマット

- » execute : ユーザーがアクセラレータを選択（クリック）したときに実行される機能（マウスでクリックした時の動作）
 - methodは、「get」 or 「post」（デフォルト「get」）
 - actionで指定するURIに対して、インラインパラメータ（この場合、where1={selection}）で変数を送信、または、フォームベースパラメータで指定した変数（nameとvalueで指定）を送信

```
<execute method="get"  
  action="http://maps.live.com/default.aspx?where1={selection}" />
```

アクセラレータの実装

アクセラレータの展開方法

1. アクセラレーターを提供する側は、OpenServiceフォーマットのXMLファイルを準備
2. ブラウザがIE8かどうか、User-Agent Stringなどでチェック
3. 提供するアクセラレーターが既にインストールされているかどうかをチェック
(コーディング例)

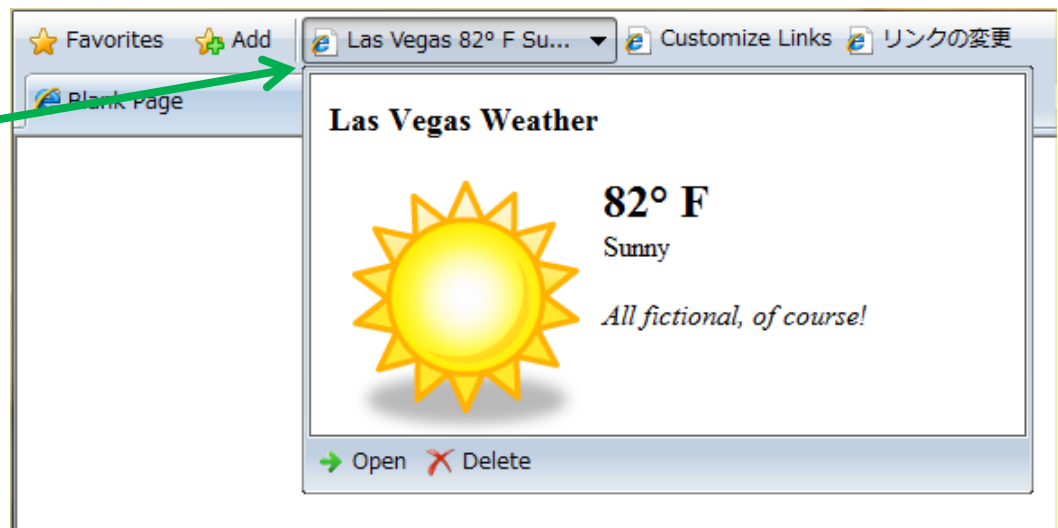
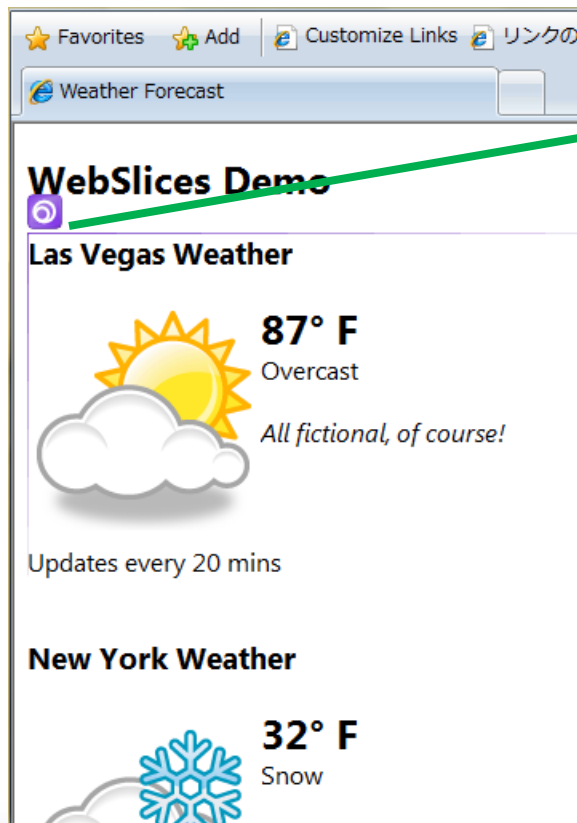
```
window.external.IsServiceInstalled(http://maps.live.com/livemaps.xml, 'map');
```

4. 提供するアクセラレーターをインストールする
(コーディング例)

```
window.external.AddService(http://maps.live.com/livemaps.xml) ;
```

WebSlice

頻繁に更新されるWebページの一部を切り取り、
簡単に更新情報をチェックできる機能





Demo

WebSlice

お気に入り バー

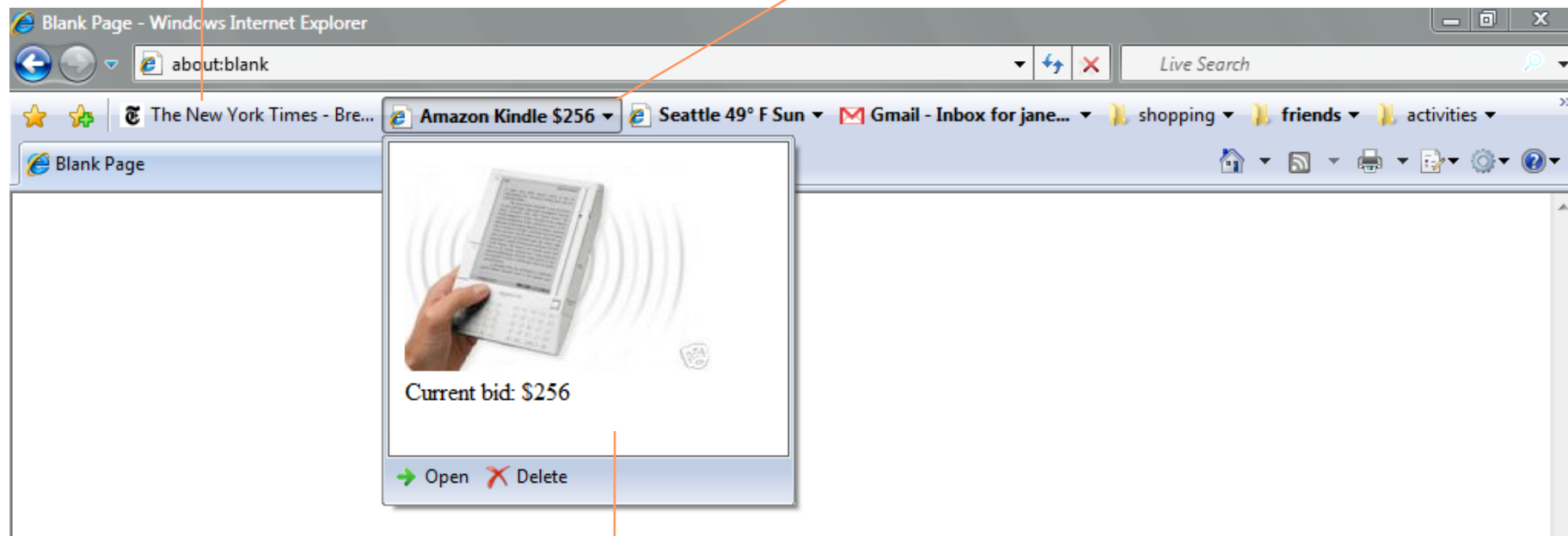
お気に入り リンク、Feeds、WebSlices
フォルダーからコンテンツを管理可能

コンテンツの状態表示

ボード：アップデート

イタリック：終了間近

グレー：終了もしくは、エラー



詳細のプレビューウィンドウ

Web ページからの HTML 引用 (static コンテンツ)

基本的な HTML スタイルをインポート

クリックすることで web サイトへアクセス

WebSliceの実装

WebSliceフォーマット

WebSliceを有効にするには、WebSliceのHTML属性をWebページに記述する
WebSliceは、hAtomマイクロフォーマットとWebSliceフォーマットの組み合わせを使用して記述する

```
<html>  
<body>
```

切り取りアイコン表示領域

更新時データ同期領域

```
<div class="hslice" id="1" >  
  <p class="entry-title">Amazon Kindle, Unopened</p>  
  <div class="entry-content">  
      
    <p>Current bid: US $282.11</p>  
    <p>This clip updates every <span class="ttl">15</span> minutes </p>  
  </div>  
</div>
```

```
...  
</body>  
</html>
```



WebSliceの実装

WebSlice フォーマット

- » hAtom は **static** コンテンツを表示できる
- » Web Slices は **dynamic** コンテンツを表示する
- » hAtom プロパティの再利用と、Subscribing 用のプロパティを追加

- » hAtomプロパティ（基本プロパティ）

entry-title – feed アイテムのタイトル名

```
<p class="entry-title">Item - $66.00</p>
```

entry-content – feed アイテムの内容

```
<div class="entry-content">high bidder: buyer1
```

...

```
</div>
```

WebSliceの実装

WebSlice フォーマット

» Web Slicesに追加したプロパティ（オプション）

ttl – （オプション）有効期間

`<div>Updates every <span class="ttl">60</span>mins</div>`

feedurl – （オプション）更新情報を取得するための代替 URL

`<a rel="feedurl" href="auction.microsoft.com/item.xml">Subscribe to WebSlice</a>`

endtime – （オプション）終了時刻

`<p>End time:<abbr class="endtime" title="2008-02-28T17:00:00">1 day 18 hours</abbr></p>`

WebSliceの実装

WebSliceの展開方法

AddToFavoritesBar API

ユーザーがこのメソッドを呼び出すと、WebSlice の購読を許可するダイアログが表示される。

window.external.addToFavoritesBar(string URL, string Title, [optional] string Type)

URL : WebSliceのURL (必須)

Title : WebSlice の名前 (必須)

Type : リンクのタイプ (Slice) (オプション)

(コーディング例)

`<button`

```
onclick="javascript:window.external.addToFavoritesBar('http://auction.microsoft.com/item  
#1', 'Item - $66.00', 'slice')">Add WebSlice</button>
```

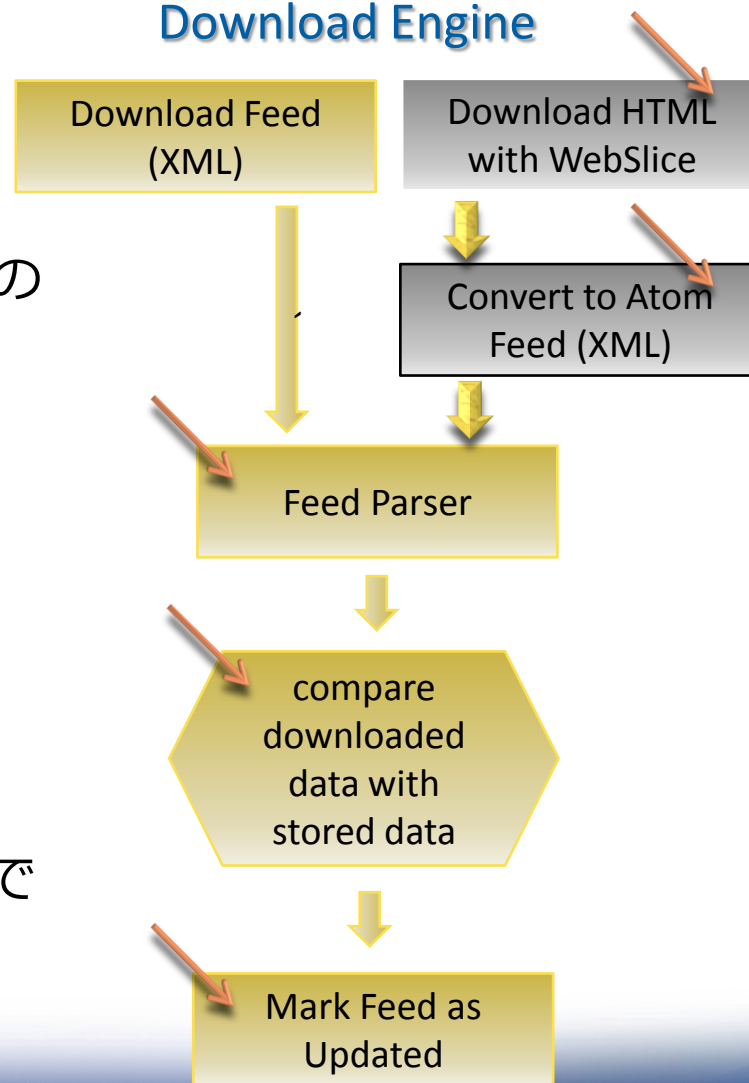
WebSlice ブラウザでの処理プロセス

WebSliceは、フィードと同様に、
Windows Feeds プラットフォームで
処理される
Feed Parser にデータが持ち込まれるまでの
プロセスが異なる

- フィードの場合は フィード (XML) をダウンロードして Feed Parser へ
- WebSlice は WebSlice の記述がある HTML をダウンロードして Atom Feed (XML) にコンバートしてから Feed Parser へ

ダウンロードしたコンテンツは、
ローカルに保存されて、フィードAPI 経由で
アクセスすることが可能

Download Engine



WebSlice ネットワーク管理

フィードのダウンロードエンジンはデフォルトでは1日1回更新を確認

- 更新間隔の最小単位は 15 分
 - 「ツール」->「インターネットオプション」->「コンテンツ」->「フィード」->「設定」で変更可能
-
- 最小更新間隔 (time-to-live) の指定
この ttl 設定値はユーザー設定値より優先される
`<div>Updates every <span class="ttl">60</span>mins</div>`
 - 代替 フィードの提供
更新を異なるソース(Feed URL)から取得する設定が可能
`<a rel="feedurl" href="www.foo.com/feed.xml">Subscribe</a>`

WebSliceの認証

- » WebSlice は フィードと同じ扱い
- » データはユーザーのローカルマシンに保存される
- » プライベートなデータは認証を通して公開

- » Cookiesベースの認証
 - Cookies の有効期間が過ぎた場合、再認証が必要になる
ログインのための Link を WebSlice コンテンツで
提供することを推奨

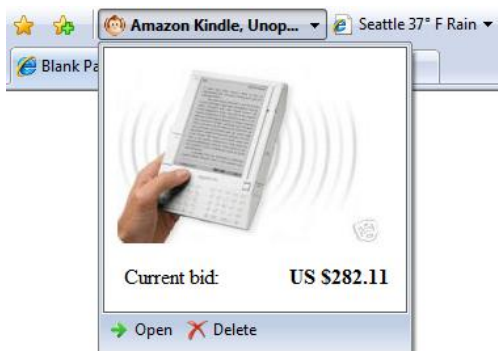
- » HTTPベースの認証
 - HTTP基本認証、HTTPダイジェスト認証をサポート
 - HTTP基本認証は、HTTPS(SSL) 経由でのみサポート

フィードとWebSliceの使い分け



フィードは新しいアイテムをプロモーションするのに便利

例：ニュース記事, ブログの投稿, 検索結果など



WebSliceは刻々と変化するWeb ページの一部を表示するのに理想的

例：天気予報, オークションアイテムやプロフィールなど

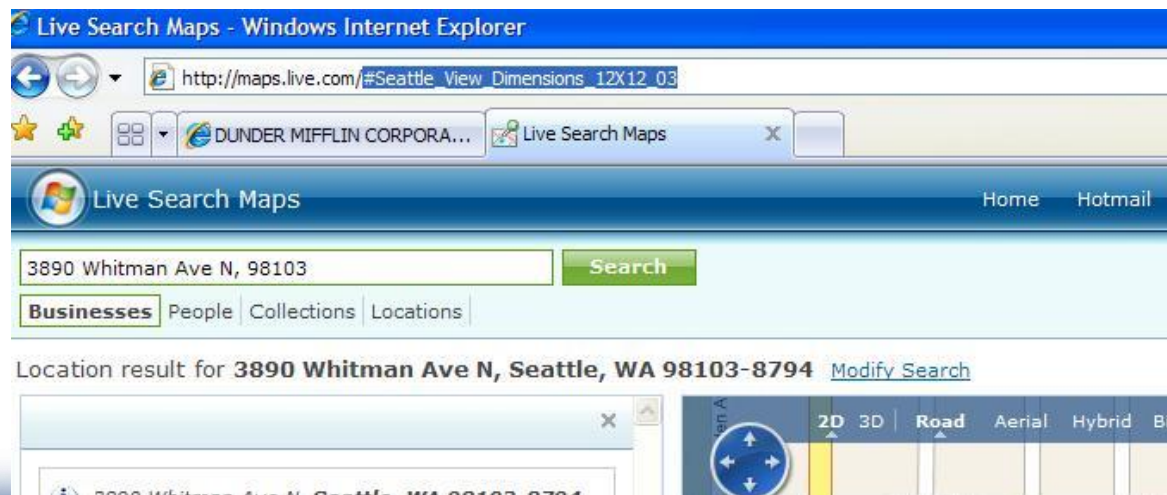
IE8 新機能について その他の新機能

IE8のその他の新機能

- AJAX ナビゲーション
- Selectors API
- DOM Storage
- Connectivity Event
- Cross Domain Communication

AJAX ナビゲーション

- W3C HTML5 Working Draft 準拠
- » AJAX による非同期通信と部分更新の履歴（パラメータ）を保存し、ブラウザの「戻る」「進む」ボタンを AJAX 処理で利用可能とする
 - `window.location.hash` を使って AJAX 処理におけるパラメータを URL として保存
 - `window.onhashchange` イベントで、保存したパラメータを使って AJAX 処理を戻す
 - Hash URL のコピー & ペーストが可能



Selectors API

- » W3C Working Draft 準拠 : Selectors API 仕様
- » Document Object Model の操作 (DOM オブジェクトの取得など) を簡単にする
- » **querySelectorAll()** – StaticNodeList* を返す
 - StaticNodeList* – 取得後の要素の変化を反映しない要素リスト
- » **querySelector()** – 最初にマッチした要素を返す
- » document 要素だけでなく、DOM サブツリーの各 element 要素でも使用できる
- » DOM API (getElementById など) より高速に DOM ツリーを検索できる

Selectors API 使用例

- CSS -

```
Selector {  
  property: value; property2: value;  
}
```

- HTML -

```
<body>  
  <div class='Selector'> ... </div>  
  ...  
</body>
```

- JavaScript -

```
var oneElement = document.querySelector('Selector');  
var elementList = document.querySelectorAll('Selector');  
for (var i = 0; i < elementList.length; i++) {  
  elementList[i].style.display = 'none';  
}
```

Selectors API と DOM API

// Selectors API の場合

```
var elementList = document.querySelectorAll('Selector');  
for (var i = 0; i < elementList.length; i++)  
{  
    elementList[i].style.display = 'none';  
}
```

// DOM API の場合

```
var elementList = document.getElementsByTagName('div');  
for (var i = 0; i < elementList.length; i++)  
{  
    if (elementList[i].className.indexOf('Selector') != -1)  
        elementList[i].style.display = 'none';  
}
```

DOM Storage

- » W3C HTML 5 Working Draft 準拠
- » 現在、Web ページの多くがローカルマシン上にデータを保存するために document.cookie を使用
- » Cookies は保存容量が限定されている
(1ドメイン毎に 20 key/value ペア、または 4,096 byte)
- » DOM Storage オブジェクトは、とてもシンプルなストレージを提供
(Key/Value ストリングペア)
- » ドメイン毎 (サブドメイン含む) に 10MB のローカルストレージを持つ
- » ブラウザのタブ毎にそれぞれのセッションストレージを持つ
- » いずれも Public Store ではない

DOM Storage サンプル

1a. ドメイン毎のグローバルストレージオブジェクトを生成

```
var storage = globalStorage["http://www.contoso.com"];
```

1b. または、セッションストレージオブジェクトを生成

```
var storage = sessionStorage["http://www.contoso.com"];
```

2. アイテム (Key/Value ペア) をストア

```
storage.setItem("John","Public");
```

3. Key に対応した Value 文字列を取得

```
var cptname = storage.getItem("John");
```

4. アイテムを削除

```
storage.removeItem("John");
```

5. ストレージ全体を削除

```
storage.clear();
```

Connectivity Events

Online / Offline イベント

- » W3C HTML 5 Working Draft 準拠
- » Web ブラウザの接続状態（オンライン/オフライン）を検出するイベントが追加

// online, offline イベントハンドラの追加

```
<body online="online()" onoffline="onoffline()">;
```

// ブラウザのオンライン状態の取得

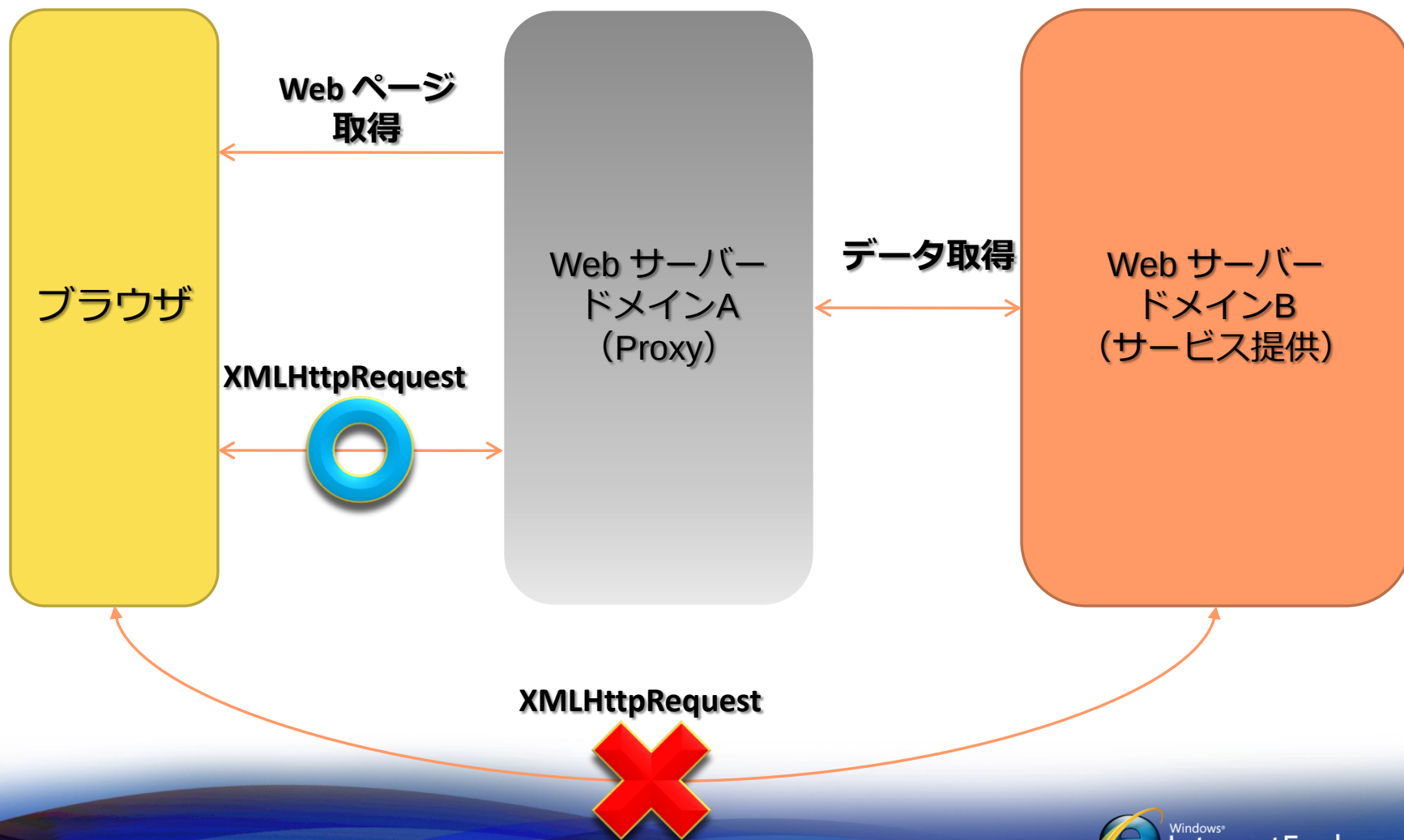
```
online = window.navigator.onLine;
```

Cross Domain Communication

- » ドメインを超えて、データやコンポーネントを活用することが今日の課題
 - 現状
 - 安全のために利用を制限している
 - XMLHttpRequest も同様に制限されている
- » 次の方法で抜け道を作ることができるが...
 - JavaScript の利用
 - 信頼できるスクリプトの作成者でないと危険
 - サーバーサイドで Proxy をかける
 - 困難な上にコストがかかる
 - IE のセキュリティオプションの変更

Cross Domain Communication

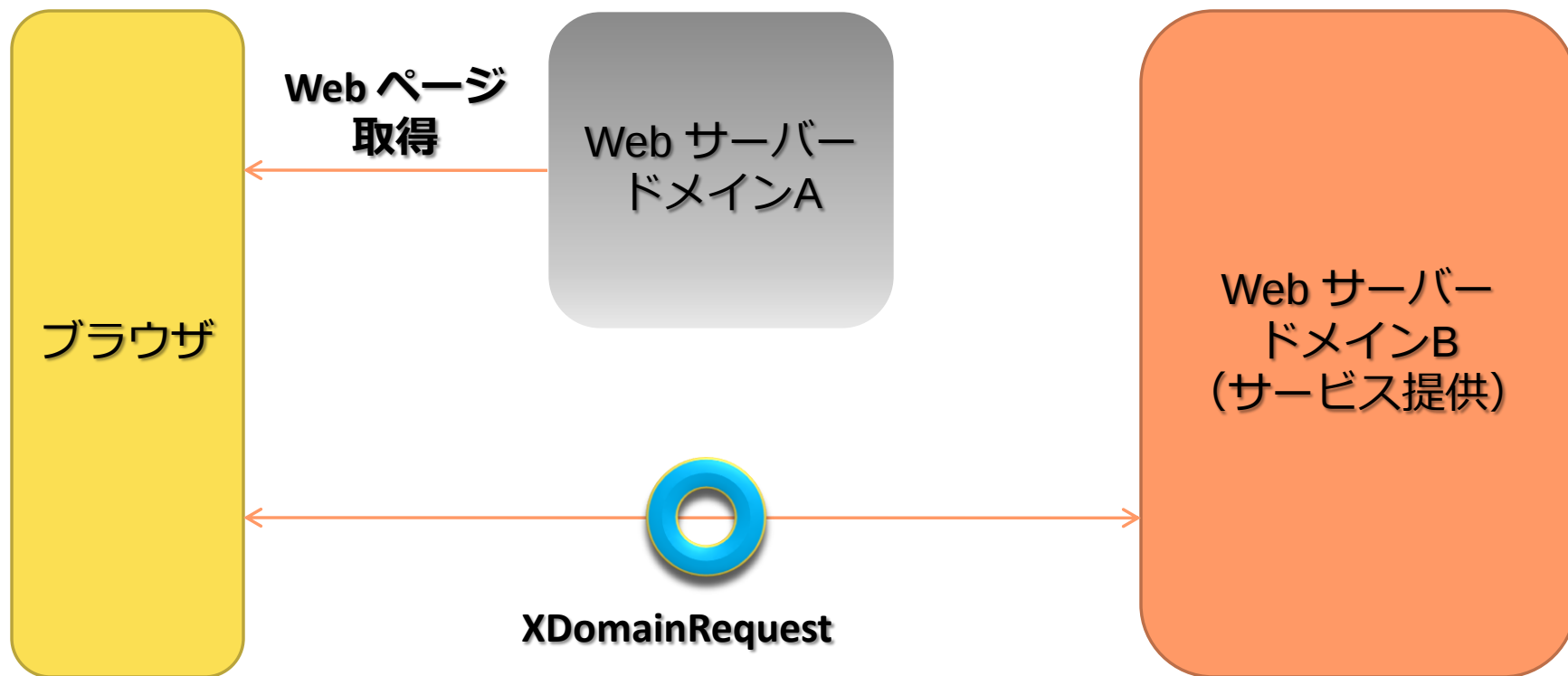
クロスドメイン制約



Cross Domain Request (XDR)

- » 新しいオブジェクト **XDomainRequest** を使用することで、異なるドメイン間のリクエストが安全に実現できる
- » **XDomainRequest (XDR)**
 - XDomainRequest オブジェクトを作成
 - XDR は HTTP リクエストヘッダで **XDomainRequest:1** を送信してデータを要求
 - XDR は、サーバーが HTTP レスポンスヘッダで **XDomainRequestAllowed:1** を返してきたときのみ、レスポンスデータを処理する
 - XDR に対応するサーバでは XDomainRequestAllowed:1 をレスポンスヘッダで返す必要がある
 - ASP –
`Response.AppendHeader("XDomainRequestAllowed","1");`
 - IIS – HTTP 応答ヘッダーで指定
 - XDR は anonymous (no cookies/auth) リクエスト

Cross Domain Request (XDR)



リクエストヘッダ [XDomainRequest:1]

レスポンスヘッダ [XDomainRequestAllowed:1]

Cross Domain Request 使用例

1. XDRオブジェクトの生成

```
var xdr = new XDomainRequest();
```

2. イベントハンドラの設定

```
xdr.onload = new function() {  
    var response = xdr.responseText;  
    ...  
}  
xdr.onerror = new function() { ... }
```

3. POST メソッドで接続をオープン

```
xdr.open("POST", "http://www.contoso.com/xdr.asmx")
```

4. データ送信

```
xdr.send("argument=value")
```

Cross Domain Request のセキュリティ

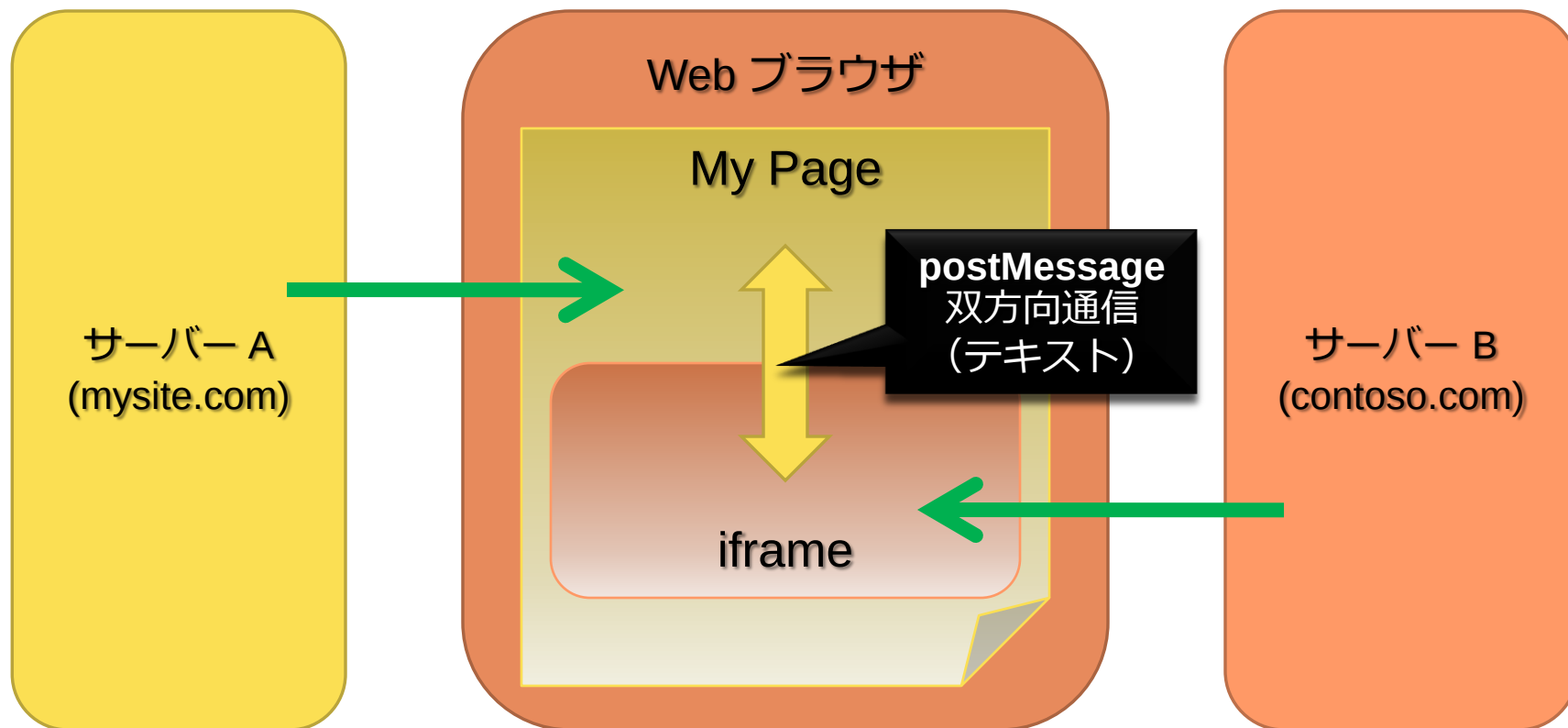
現在のページとデータを要求するページのゾーンに基づいて、XDR によるクロスドメインリクエストの実行が許可される

		Web ページがデータを要求する URL のゾーン					
		ローカル	イントラネット	信頼済み (イントラネット)	信頼済み (インターネット)	インターネット	制限付き
Web ページが存在するゾーン	ローカル	許可	許可	許可	許可	許可	ブロック
	イントラネット	ブロック	許可	許可	許可	許可	ブロック
	信頼済み (イントラネット)	ブロック	許可	許可	許可	許可	ブロック
	信頼済み (インターネット)	ブロック	ブロック	ブロック	許可	許可	ブロック
	インターネット	ブロック	ブロック	ブロック	許可	許可	ブロック
	制限付き	ブロック	ブロック	ブロック	ブロック	ブロック	ブロック

Cross Document Messaging (XDM)

- » W3C HTML 5 Working Draft 準拠 : e メソッド、その関連プロパティおよびイベント `postMessage`
- » Cross Document Messaging (XDM) が提供する `window.postMessage` メソッドは、異なるドメインとのフレーム間の双方向データ通信を安全に実現する
- » シンプルかつパフォーマンスの高い方法でクロスドキュメント通信が可能

Cross Document Messaging (XDM)



XDM サンプル

メインページ側 (<http://mysite.com>)

```
<html>
<head>
  <script>
    document.onmessage = receiver;
    function receiver() {
      if (window.event.data == "World") {
        alert("success!");
      }
    }
    function postToIframe() {
      var ff = document.getElementsByTagName("iframe")[0];
      ff.contentWindow.postMessage("Hello", "http://contoso.com");
    }
  </script>
</head>
<body>
  <iframe id="myIframe" src="http://contoso.com/gadget.htm"
    onload="postToIframe()"></iframe>
</body>
</html>
```

XDM サンプル

iframe ページ側 (<http://contoso.com>)

```
<html>
<head>
  <script>
    document.onmessage = receiver;
    function receiver() {
      var e = window.event;
      if (e.domain == "mysite.com" && e.data == "Hello") {
        e.source.postMessage("World");
      }
    }
  </script>
</head>
<body>
  <p>Child Frame</p>
</body>
</html>
```

Resources

- » Internet Explorer 8 Beta 1 for Developers ホワイトペーパー（日本語）
<http://msdn.microsoft.com/ja-jp/ie/cc216837.aspx>
- » Windows Internet Explorer 8 ベータ 1 リリースノート（日本語）
<http://support.microsoft.com/kb/949787>
- » Welcome to Windows Internet Explorer 8 Beta 1
<http://www.microsoft.com/windows/ie/ie8/welcome/en/default.html>
- » Internet Explorer 8 Beta 1 Whitepapers
<http://code.msdn.microsoft.com/ie8whitepapers>
- » Internet Explorer 8 Readiness Toolkit
<http://www.microsoft.com/windows/products/winfamily/ie/ie8/readiness/Install.htm>
- » Internet Explorer Developer Center
<http://msdn.microsoft.com/en-us/ie/default.aspx>
- » Internet Explorer 8 Activities Gallery
<http://ie.microsoft.com/activities/en-en/Default.aspx>
- » OpenService Format Specification for Activities version 0.8
[http://msdn.microsoft.com/en-us/library/cc304163\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc304163(VS.85).aspx)
- » WebSlice Format Specification version 0.8
[http://msdn.microsoft.com/en-us/library/cc304073\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc304073(VS.85).aspx)



Microsoft®

© 2008 Microsoft Corporation. All rights reserved.

本資料の第三者開示はご遠慮くださいますようお願いいたします。

製品情報、製品・機能内容、開発スケジュール、マーケティング施策などの全ての情報は予告無く変更されることがあります。あらかじめご了承ください。