

プラットフォーム パフォーマンスの向上:
Windows® Internet Explorer® 8
Beta 1 for Developers



Web 作業の操作性を向上

2008 年 3 月

詳細の問い合わせ先 (報道関係者専用):
Rapid Response Team
Waggener Edstrom Worldwide
(503) 443-7070
rrt@waggeneredstrom.com

このドキュメントに記載された内容は、ドキュメント発行時点での記載内容に関する米国 Microsoft Corporation の見解を示すものです。マイクロソフトが市場状況の変化に対応する必要性から、このドキュメントの記載内容をマイクロソフトによる確約とみなすことはできません。また、マイクロソフトは、ドキュメント発行後、いかなる記載内容の正確性についても保証いたしません。

このドキュメントに記載された内容は情報の提供のみを目的としており、明示、黙示または法律の規定にかかわらず、これらの情報についてマイクロソフトはいかなる責任も負わないものとします。

お客様ご自身の責任において、適用されるすべての著作権関連法規に従ったご使用を願います。このドキュメントのいかなる部分も、米国 Microsoft Corporation の書面による許諾を受けることなく、その目的を問わず、どのような形態であっても、複製または譲渡することは禁じられています。ここでいう形態とは、複写や記録など、電子的な、または物理的なすべての手段を含みます。ただしこれは、著作権法上のお客様の権利を制限するものではありません。

Microsoft Corporation (以下「マイクロソフト」といいます) は、このドキュメントに記載されている内容に関し、特許、特許申請、商標、著作権、またはその他の無体財産権を有する場合があります。別途マイクロソフトのライセンス契約上に明示の規定のない限り、このドキュメントはこれらの特許、商標、著作権、またはその他の無体財産権に関する権利をお客様に許諾するものではありません。

別途記載されていない場合、このソフトウェアおよび関連するドキュメントで使用している会社、組織、製品、ドメイン名、電子メール アドレス、ロゴ、人物、場所、出来事などの名称は架空のものです。実在する商品名、団体名、個人名などとは一切関係ありません。

© 2008 Microsoft Corp. All rights reserved.

Microsoft、Windows、Windows Vista、Windows Server、ActiveX、Active Directory、Internet Explorer、Internet Explorer のロゴ、JScript、MSN、および MSN のロゴは、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

記載されている会社名、製品名には、各社の商標のものもあります。

概要

Windows Internet Explorer 8 Beta 1 for Developers では、Web サイトのパフォーマンスを向上させる複数の変更が行われました。これらの変更は、HTML パーサー、CSS ルールの処理、マークアップツリーの操作、Jscript ガベージ コレクタ、ランタイムなどのさまざまな Internet Explorer サブシステムに影響を及ぼします。

このドキュメントでは、これらの変更のいくつかと対応するベスト プラクティスについて説明します。これらのベスト プラクティスは、Internet Explorer 8 Beta 1 for Developers の機能向上を最大限に利用し、Internet Explorer 6 や Internet Explorer 7 に潜在していた問題を回避するうえで役立ちます。

互換性 : Internet Explorer 7 からの動作変更

これらの変更の多くはパフォーマンスにのみ影響を及ぼすものであり、大多数の Web サイト上の機能はほとんど影響を受けません。ただし、Web サイトの操作のタイミングやブラウザのパフォーマンスに依存する部分では影響が生じる場合があります。さらに、オブジェクトの解放後にそれと知らずにそれらのオブジェクトにアクセスする Web サイトは、Internet Explorer 8 Beta 1 for Developers ではこのようなアクセスができない可能性があります。

Internet Explorer 8 のパフォーマンスの向上は、選択されたレンダリング モードにかかわらず、Internet Explorer 8 で読み込まれた各 Web サイトに影響を及ぼします。

機能の詳細

Internet Explorer コンポーネントの機能向上

最適化により Internet Explorer 8 のさまざまなコンポーネントの機能が前のバージョンよりも向上しました。向上した主な機能は、以下のとおりです。

- コレクションのキャッシュの最適化

Internet Explorer により、メモリに HTML ドキュメントを表示するためのマークアップ ツリーが構築される場合、さまざまなドキュメント要素へのその後のアクセスを迅速化するために、多数のコンパニオン キャッシュも構築されます。以下にその例を示します。

- Internet Explorer では、データ無効化の機能向上によりコレクションの再構築にかかる時間が短縮されました。これにより、特に、スクリプトによってドキュメントが動的に変更される間、繰り返し同じコレクションにアクセスするサイトが高速になります。
- **childNodes** コレクションは、後続のアクセスに備えて戻り値を完全にキャッシュすることにより、高速になりました。

Internet Explorer 8 Beta 1 for Developers でこれらのコレクションのキャッシュ速度が向上する一方、以下のベスト プラクティスによって、特に Internet Explorer 6 および Internet Explorer 7 を対象とするコードがさらに高速になります。

ベスト プラクティス

最後のルックアップのみが繰り返し実行されるように、一連のルックアップが中間値をキャッシュするように変更します。

たとえば、以下のコードを変更します。

```
function badLoop(div1)
{
for(var i=0; i < div1.parentNode.childNodes.length; i++) {
var node = div1.parentNode.childNodes[i];

// Do something with node.

};
}
```

以下のように変更します。

```
function betterLoop(div1)
{
// Cache length and array to prevent multiple lookups.
var childLength = div1.parentNode.childNodes.length;
var childNodes = div1.parentNode.childNodes;

for(var i=0; i < childLength; i++) {
var node = childNodes[i];
// Do something with node.
};
}
```

ベスト プラクティス

nextSibling メソッドを使用して、要素の子をスキャンします。

たとえば、**betterLoop** を高速にするには、以下のようにします。

```
function bestLoop(div1)
{

for(node = div1.parentNode.childNodes[0]; node != null; node =
node.nextSibling) {
alert("yep");
// Do something with node.
};
}
```

ベスト プラクティス

ドキュメント コレクションを使用してドキュメント内の要素にアクセスすると、スクリプトがそのドキュメントを変更した後で Internet Explorer によってそれらのコレクションが再構築されるため、コレクションへのすべてのアクセスをバッチ処理し、ドキュメントに対する変更とは別に実行します。

- 循環参照によって発生するメモリ リークの軽減

[MSDN の記事 \(英語\)](#) で詳細に説明されているとおり、Internet Explorer の前のバージョンでは、JScript ガベージ コレクタによって JScript オブジェクトの有効期間が管理されますが、DOM オブジェクトの有効期間は管理されません。その結果、JScript ガベージ コレクタは DOM オブジェクトと JScript オブジェクト間の循環参照を中断できず、メモリ リークが発生します。Internet Explorer 6 では、Internet Explorer プロセスが終了すると循環参照が中断されません。Internet Explorer 7 では、ユーザーがメモリ リークを含むページから移動すると、循環参照が中断されます。

しかし、Internet Explorer 8 Beta 1 for Developers では、メモリ管理のアーキテクチャが変更され、JScript が参照する DOM オブジェクトを JScript ガベージ コレクタが管理できるようになりました。JScript ガベージ コレクタは、オブジェクトが循環参照によってのみメモリに保持されている時期を確認できるため、ページの有効期間中、それらのオブジェクトを復元できます。

Web 開発者は、Internet Explorer 7 での JScript の終了などのプログラミング パターンの使用により発生するメモリ リークについて認識する必要がありますが、Internet Explorer 8 Beta 1 ではこうしたプログラミング パターンによってメモリ リークが発生することはなくなりました。

- DOM 検索アルゴリズムの最適化

ID、タグ、名前、またはその他の特性によりドキュメントの要素を検索する場合、Web サイトが頻繁に検索を繰り返す不経済な運用になります。Internet Explorer 8 では、一般的な `getElementById` と `getElementsByTagName` メソッドにおける検索アルゴリズムが向上し、前のバージョンよりも迅速に実行できるようになりました。

さらに、Internet Explorer 8 Beta 1 for Developers では CSS セレクタ API が実装され、JScript が CSS セレクタの強力で柔軟な言語を使用して DOM 要素にアクセスできるようになりました。このような API は、Internet Explorer の CSS セレクタと一致するコンポーネントを使用するため、新しいセレクタの追加と長期的なパフォーマンスの向上という利点が得られます。状況によっては、このような API を使用することにより、セレクタ API を実装していない他のブラウザに必要な従来のメソッドよりも格段にスピードが向上します。

ベスト プラクティス

セレクタ API を実装していないサイトでは、以下のスクリプトを使用できます。

```
function doValidation() {
    // Retrieve all the required elements by ID
    var name = document.getElementById('fld_name');
    var empnum = document.getElementById('fld_emp');
    var ssn = document.getElementById('fld_ssn');
    var ccnum = document.getElementById('fld_ccNum');
    var bribe = document.getElementById('fld_cost');
    // Set the flag to false by default
    var missingRequiredField = false;
    // Validate each of the elements
    if (name.childNodes[1].value == "")
        missingRequiredField = true;
    if (empnum.childNodes[1].value == "")
        missingRequiredField = true;
    if (ssn.childNodes[1].value == "")
        missingRequiredField = true;
    if (ccnum.childNodes[1].value == "")
        missingRequiredField = true;
    if (bribe.childNodes[1].value == "")
        missingRequiredField = true;
}
```

セレクタ API を使用すると、以下のスクリプトで同じタスクを実行できます。

```
function doValidation() {
  // Retrieve the required elements by using Selectors
  // Selects all the form fields with 'required' classes
  var reqs = document.querySelectorAll(".required");
  // Set the flag to false by default
  var missingRequiredField = false;
  // Validate that the form data is not empty
  for (var i = 0; i < reqs.length; i++) {
    if (reqs[i].value == "")
      missingRequiredField = true;
  }
}
```

JScript エンジンの向上

Internet Explorer 8 Beta 1 for Developers の JScript エンジンには、JScript 実行の全体的なパフォーマンスを向上させる複数の変更が行われました。実施された主な変更には、以下のものがあります。

- 文字列の連結とその他の組み込み文字列操作の最適化

特に AJAX シナリオで最も一般的な JScript 演算の 1 つである文字列の連結が、Internet Explorer の前のバージョンから大幅に向上しました。たとえば、演算子 + および += の使用や組み込み `String.concat` メソッドが挙げられます。現在、Internet Explorer 7 および Internet Explorer 6 で `Array.join` を使用して文字列を連結している JScript 作成者は、Internet Explorer 8 で文字列の連結に演算子 + を使用してパフォーマンスの向上を図ることができます。

ベスト プラクティス

以下のように `Array.join` を使用して文字列を追加する代わりに、

```
<script language=javascript>
var smallStrings = new Array();
// Fill array with smaller strings
var largerString = smallStrings.join('');
</script>
```

以下のように演算子 + を使用します。

```
<script language=javascript>
var smallString1 = "string1";
var smallString2 = "string2";
// String concatenation using the "+" operator
var largerString = string1 + string2;
</script>
```

Internet Explorer 8 Beta 1 for Developers の JScript エンジンに加えられた設計変更によって、その他の組み込み **String** メソッドも向上します (さまざまな組み込み **String** オブジェクトメソッドについては、[MSDN \(英語\)](#) を参照してください)。

文字列操作では、ガベージ コレクタのチューニング機能を利用することもできます。スクリプトによって大量の **String** オブジェクトの作成や削除が行われる場合、このチューニング機能を使用するとガベージ コレクタ サイクルが短縮されるため、JScript コードのパフォーマンスが向上します。このような最適化は、Internet Explorer 7 で JScript ガベージ コレクタに対して行われた変更に対しても適用されます。これらの変更は [Windows Script 5.7 \(英語\)](#) でも行われています。

- **組み込み配列操作の最適化**

配列の記憶域および JScript エンジン内でのアクセスの最適化によって、ほとんどの組み込み **Array** メソッドでパフォーマンスが向上します (さまざまな組み込み **Array** オブジェクトメソッドについては [MSDN \(英語\)](#) を参照してください)。組み込み **Array** メソッドのより迅速な代替メソッドとして独自の **Array** メソッド (**Array.push** や **Array.pop** など) を実装している JScript 作成者は、組み込みメソッドを使用してコード パフォーマンスを向上させることを検討してください。

ベスト プラクティス

以下のように独自の **Array.push** や **Array.pop** 演算を実装する代わりに、

```
<script language=javascript>

//user defined array type called 'prototypeArray'
var prototypeArray = function()
{
  this._array = [];
  this.length = 0;
  this.start = 0;
  this.end = 0;

  //push method for user defined array type 'prototypeArray'
  prototypeArray.prototype.push = function()
  {
    var l = arguments.length;
    for (var i=0;i<l;i++){
      this.length++;
      this._array[this.end++] = arguments[i];
    }
  }

  //pop method for user defined array type 'prototypeArray'
  prototypeArray.prototype.pop = function()
  {
    var obj = null;
    if (this.length>0){
      this.length--;
      obj = this._array[--this.end];
      deletethis._array[this.end];
    }
    return obj;
  }
}
```

```
//creating an object of user defined array type 'prototypeArray'  
var myArray = new prototypeArray();  
//accessing push and pop methods of user defined array type  
'prototypeArray'  
myArray.push("Test String")  
myArray.pop();  
  
</script>
```

開発者は、以下に示す組み込みの **Array.push** および **Array.pop** 演算を使用してください。

```
<script language=javascript>  
//creating an object of standard Array type  
var standardArray = new Array();  
//accessing built-in push and pop methods of standard Array type  
standardArray.push("Test String");  
standardArray.pop();  
  
</script>
```

- 実行エンジンの最適化

Internet Explorer 8 Beta 1 for Developers 内の JScript 実行エンジンによって、以下のパフォーマンスの最適化と向上が実現されます。

- 関数呼び出し
- オブジェクトのプロトタイプ プロパティへのアクセス
- window.var (var は変数) と this.var (this はグローバル window オブジェクトを示す) を使用した JScript 変数の特定のルックアップ パターン