

循環メモリ リークの緩和：  
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](mailto: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 ロゴ、JScript、MSN および MSN 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、メモリ管理機能が向上しました。この機能向上により、Microsoft® JScript® オブジェクトと DOM オブジェクトとの間の循環参照によって発生していたメモリ リークが緩和されます。

## 互換性 : Internet Explorer 7 からの動作変更

この機能は、Internet Explorer の前のバージョンで存在していた、メモリ リークのみによって生じるガベージ メモリに依存する Web ページの動作に影響を及ぼします。Internet Explorer 8 Beta 1 for Developers では、これらのページは未割り当てのメモリを参照し、エラーになります。

この機能は、サイトの選択されたレンダリング モードにかかわらず、Internet Explorer 8 Beta 1 for Developers で表示される各 Web サイトに影響を及ぼします。

## 機能の詳細

[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、この循環参照が中断されます。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。

Internet Explorer 8 Beta 1 for Developers では、JScript ガベージ コレクタは JScript オブジェクトが参照する DOM オブジェクトを他の JScript オブジェクトとして処理します。Internet Explorer 7 の場合のページ移動や、Internet Explorer 6 の場合のプロセスの終了を待たずに、ガベージ コレクタはこれらの DOM オブジェクトの有効期間を管理し、サイトの有効期間内であれば循環参照を中断できます。

Web 開発者は、Internet Explorer 7 以前での 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 for Developers 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。

ここでは、以前はメモリ リークが発生したが、Internet Explorer 8 Beta 1 for Developers ではリークが発生しない簡略化したコード パターンの例を紹介します。

### 直接メモリ循環参照

- 自身を参照したりツリー外にあるオブジェクトを参照する DOM オブジェクトの場合の循環参照

```
function leaktest1()
{
    var elem1 = document.createElement("DIV");
    elem1.thing = elem1;
}
```

- ツリー外の DOM オブジェクト間の循環参照

```
function leaktest2()
{
    var elem1 = document.createElement("DIV");
    var elem2 = document.createElement("DIV");
    elem1.thing = elem2;
    elem2.item = elem1;
}
```

- DOM オブジェクトと JScript オブジェクト間の循環参照

```
function leaktest3()
{
    var x = new Object();
    x.obj = document.createElement("DIV");
    x.obj.jsobj = x;
}
```

- 一時マークアップ時の DOM オブジェクト間の循環参照

```
function leaktest4()
{
    var elem1 = document.createElement("DIV");
    var elem2 = document.createElement("DIV");
    elem1.appendChild(elem2);
    elem1.thing = elem2;
    elem2.item = elem1;
}
```

- removeNode を使用してツリーからオブジェクトが削除された場合の DOM オブジェクト間の循環参照

```
function leaktest5()
{
    var elem1 = document.createElement("DIV");
    document.body.appendChild(elem1);
    elem1.thing = elem1;
    elem1.removeNode(true);
}
```

- innerHTML を使用してツリーからオブジェクトが削除された場合の DOM オブジェクト間の循環参照

```
function leaktest6()
{
    var elem1 = document.createElement("DIV");
    document.body.appendChild(elem1);
    elem1.thing = elem1;
    elem1.parentElement.innerHTML = "";
}
```

#### 終了により発生するメモリ循環参照

- 終了と共に要素が動的に作成される場合

```
function leaktest7()
{
    var elem1 = document.createElement("DIV");
    elem1.onload = function () {
        var y = elem;
    }
}
```

- 終了と共に要素がツリーに追加され、removeNode を使用してツリーから削除される場合

```
function leaktest8()
{
    var elem = document.createElement("DIV");
    document.body.appendChild(elem);
    elem.onload = function () {
        var y = elem;
    }
    elem.removeNode();
}
```

- 終了と共に要素がツリーに追加され、innerHTML を使用してツリーから削除される場合

```
function leaktest9()
{
    var elem = document.createElement("DIV");
    document.body.appendChild(elem);
    elem.onload = function () {
        var y = elem;
    }
    elem.parentElement.innerHTML = "";
}
```

- 関数ポインタにより発生するリーク

```
function leaktest10()
{
    var elem = document.createElement("DIV");
    elem.thing = elem.setAttribute;
}
```