

Internet Explorer 9 互換性検証

日本マイクロソフト株式会社

発行 : 2012 年 06月



[注意事項]

このドキュメントに記載されている情報 (URL 等の Web サイトに関する情報を含む) は、将来予告なしに変更することがあります。このドキュメントは情報提供のみを目的としており、明示または黙示に関わらず、これらの情報及び情報を使用した結果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。また、このドキュメントに記載された内容は作成日時点の情報に基づくものです。

お客様がこのドキュメント及び本製品を使用した結果については、すべてお客様が負うものとします。このドキュメントのご利用にあたっては、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う必要があります。このドキュメントのいかなる部分も、特に記載のない限り、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。

マイクロソフトは、このドキュメントの内容に関し、特許権、特許出願、商標権、著作権、またはその他の知的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に明示されない限り、このドキュメントはこれらの特許権、商標権、著作権、またはその他の知的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。

特に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、出来事などの名称は架空のものです。実在する会社名、組織名、商品名、個人名などとは一切関係ありません。

© 2011 Microsoft Corporation. All rights reserved.

Microsoft、Windows、Windows Vista、Windows Server、Active Directory は、米国 Microsoft Corporationまたはその関係会社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、各社の商標も含まれます。

目次

[1. はじめに 6](#_Toc326916506)

[1.1. 対象読者 6](#_Toc326916507)

[1.2. 対象システム 7](#_Toc326916508)

[2. 互換性検証 プロセス 8](#_Toc326916509)

[2.1. 検証プロセス 8](#_Toc326916510)

[2.2. インベントリ収集 9](#_Toc326916511)

[2.3. 優先順位付け 10](#_Toc326916512)

[2.4. Internet Explorer 9 構成検討 11](#_Toc326916513)

[2.5. パイロット テスト 12](#_Toc326916514)

[2.6. テスト 12](#_Toc326916515)

[2.7. 回避策検討 13](#_Toc326916516)

[2.7.1. Internet Explorer 9 構成変更 13](#_Toc326916517)

[2.7.2. 互換表示設定の利用 13](#_Toc326916518)

[2.7.3. ソースコードの修正 14](#_Toc326916519)

[2.7.4. 仮想化テクノロジの利用 15](#_Toc326916520)

[2.8. 対応実施 18](#_Toc326916521)

[3. 検証前の準備 (IT 管理者向け) 19](#_Toc326916522)

[3.1. Internet Explorer 9 構成検討 ポイント 19](#_Toc326916523)

[3.1.1. セキュリティ ゾーン 19](#_Toc326916524)

[3.1.2. 「ローカル イントラネット」ゾーン 判定 22](#_Toc326916525)

[3.1.3. 「ローカル イントラネット」ゾーン セキュリティ 設定 23](#_Toc326916526)

[3.1.4. 詳細設定オプション 24](#_Toc326916527)

[3.2. 互換表示 設定 26](#_Toc326916528)

[3.2.1. 互換表示 設定とは 26](#_Toc326916529)

[3.2.2. 互換表示 設定方法 26](#_Toc326916530)

[3.3. IEAK 27](#_Toc326916531)

[4. 検証ポイント (開発者向け) 28](#_Toc326916532)

[4.1. ブラウザーモード、ドキュメントモード 28](#_Toc326916533)

[4.1.1. DOCTYPE宣言とIE9のドキュメントモードについて 31](#_Toc326916534)

[4.1.2. ドキュメントモードを指定（変更）について 33](#_Toc326916535)

[4.1.3. 検証におけるモードについて 36](#_Toc326916536)

[4.2. 画面表示 41](#_Toc326916537)

[4.2.1. ユーザー インターフェイスの変更 41](#_Toc326916538)

[4.2.2. 強化されたタブ 51](#_Toc326916539)

[4.2.3. HTMLの機能強化（レンダリングエンジンの変更） 53](#_Toc326916540)

[4.2.4. CSSサポートの強化 63](#_Toc326916541)

[4.2.5. Windowの操作 74](#_Toc326916542)

[4.3. Java Script 79](#_Toc326916543)

[4.3.1. ECMA Script 5 標準準拠 79](#_Toc326916544)

[4.3.2. DOMイベントの変更 標準準拠 88](#_Toc326916545)

[4.3.3. DOM L2/3対応標準準拠 91](#_Toc326916546)

[4.4. セキュリティ 95](#_Toc326916547)

[4.4.1. セキュリティ ゾーン 95](#_Toc326916548)

[4.4.2. 保護モード 96](#_Toc326916549)

[4.4.3. LCIE 96](#_Toc326916550)

[4.4.4. DEP/NX 97](#_Toc326916551)

[4.5. ActiveX コントロール 97](#_Toc326916552)

[4.5.1. ActiveX 仕様変更 97](#_Toc326916553)

[4.5.2. ActiveX Opt-in 97](#_Toc326916554)

[4.5.3. ユーザーによるインストール 97](#_Toc326916555)

[4.5.4. サイト構成 ActiveX 98](#_Toc326916556)

[4.5.5. ActiveX Installer Service 98](#_Toc326916557)

[4.6. その他 99](#_Toc326916558)

[4.6.1. アドオンがデフォルトで無効に設定される 99](#_Toc326916559)

[4.6.2. アドオンのパフォーマンス アドバイザー 99](#_Toc326916560)

[5. まとめ 100](#_Toc326916561)

# はじめに

2009年10月に最新のクライアント OS である Windows 7 がリリースされたことに加え、Windows XP の延長サポートが 2014年4月17日に終了することもあり、Windows XP からの Windows 7 への移行を検討されるお客様が増えてきました。

Windows XP から Windows 7に移行する際に問題となるのがアプリケーションの互換性です。

アプリケーションの互換性は、組織が最新バージョンの Windows オペレーティング システムへのアップグレードを躊躇する主な理由の 1 つとなっています。近年ではシステムの Web 化が進んでいることもあり、Windows 7 など新しいオペレーティング システムへのアップグレード時にInternet Explorer 6 向けに構築された Web アプリケーションの互換性確保が重要なポイントです。

本書では、Windows XP 上で利用されている Internet Explorer 6 を最新の Windows 7 と Internet Explorer 9 に移行する際の互換性検証のプロセスや注意点について説明します。

本書の内容は次の通りです。

* Web アプリケーションの互換性検証プロセス
* Internet Explorer 9 構成定義
* Internet Explorer 9 における検証ポイント、変更点など

なお、本書では Windows 7 上で動作する Internet Explorer 9 の互換性に関してのみ記載しており、Windows 7 のアプリケーション互換性に関しては対象外としています。Windows 7 上のアプリケーション互換に関しては、参考資料をご確認ください。

また、HTML 5 や Internet Explorer 9 新機能など、Web アプリケーション開発技法についても一切触れていませんので、こちらも参考資料などをご確認ください。

## 対象読者

現在、Windows XP で Internet Explorer 6 を利用しており、Windows 7 + Internet Explorer 9 への移行や互換性検証を計画している IT 管理者、Web 開発者 を対象としています。

また、移行対象となる Web アプリケーションは企業内に展開している Web サイト、Web アプリケーションを想定しています。インターネット上に構築され、不特定多数のユーザーが利用するようなシステムは本書の対象ではありません。

## 対象システム

本書で対象としているシステムの前提は以下の通りです。

* Windows 7 Professional または Enterprise Edition の SP1
* Internet Explorer 9

Windows 7 は 32bit バージョンと 64bit バージョンがリリースされておりますが、Internet Explorer 9 に限っては 32bit バージョンを前提としています。Internet Explorer 9 の 64bit バージョンは対象外です。

# 互換性検証 プロセス

## 検証プロセス

Web アプリケーションの互換性検証や新しいプラットフォームへの移行をを行う場合、その成否に大きな影響を与えるのは検証プロセスに対する一貫したアプローチです。これには、環境内の資産の一覧（インベントリ）を作成し、その資産の組織にとっての価値を把握し、テストおよび修復リソースを効果的に割り当てることが含まれます。

成功するプロジェクトの多くが、次の図に示すように、多様なアプリケーション互換性テスト プロセスを行っています。



## インベントリ収集

インベントリ収集では組織内に存在する Web アプリケーションの情報を収集していきます。

検証対象の Web アプリケーション数を把握し、テスト計画や予算化、スケジュール立案に利用します。

インベントリ収集は今後の計画を立案する上で重要なプロセスですが、企業内に存在する Web アプリケーションの情報を収集することは一般的には非常に困難です。

クライアントにインストールされているアプリケーションの場合には、ソフトウェア管理システムや資産管理システムにより収集されたインベントリ情報を利用することで全体の情報を比較的容易に把握することが可能です。しかし、Web アプリケーションは一般的なソフトウェア管理システムで情報を収集することが困難なため、通常はシステム担当者、開発者、利用者にヒアリングを行って収集することになります。

以下に挙げる項目はインベントリ収集で確認する項目の一例です。

* Web アプリケーションの基本情報  
  アプリケーションの重要度、利用部門、利用者数といったアプリケーションの基本情報など
* 連絡先  
  今後の連絡先となるシステム オーナー、開発者、利用部門の責任者の情報など
* システム構成、利用環境  
  Web アプリケーションの構成、標準的な利用環境、サポートしているクライアント OS、Web ブラウザーのバージョン、Proxy 接続の有無、Single-Sign-On の導入状況、必要ソフトウェアなど
* コンポーネント  
  Web アプリケーションで Add-on, ActiveX, Java, Web フレームワーク, JavaScript ライブラリなどのコンポーネントを利用している場合、コンポーネント情報や Internet Explorer 9 の対応状況、サポート有無など
* システム化計画  
  今後のバージョンアップやシステム更新予定などのシステム計画情報

## 優先順位付け

インベントリ収集で収集された情報を基に、アプリケーションの優先順位付けを行います。

一般的には Web アプリケーションを “ビジネスに対する影響” と、”アプリケーションの利用範囲” の２つの要素によって優先度を設定していきます。

* ビジネスに対する影響  
  組織にとって、ビジネスインパクトの大きいアプリケーションを確認します。  
  アプリケーションの互換性問題によりアプリケーションが利用できず、ビジネスに直接的な影響が発生するものは優先度を高く設定します。
* アプリケーションの利用範囲  
  アプリケーションが影響を与えるユーザー数を考慮します。  
  組織外 (お客様、パートナー)、組織全体、部門、グループといったアプリケーションの利用範囲を特定、分類し、より広範囲な影響を与えるアプリケーションの優先度を高く設定します。

組織のビジネスに対する影響に基づいてアプリケーションに優先順位を付けることは、どこにリソースを割り当てる必要があるかを理解するうえで不可欠です。

次の一覧は、アプリケーションに優先順位を付ける際の分類例を示しています。

* **Unimportant (重要ではない)**: これらのアプリケーションはテストまたは修正のどのような投資対象にもなりません。
* **Nice to have (あると便利)**: これらのアプリケーションは簡単な方法として役立つことがあり、互換性がある場合には環境に残されることもありますが、それ以外の場合は削除されます。
* **Needed but not critical (必要だがクリティカルではない)**: これらのアプリケーションは "必要に応じて" 修正されることも多いですが、スモーク テスト以上の投資は行われません。ワークグループには、通常、場合によって必要になることがあるこれらのアプリケーションを修正が行われるまで使用できるように、Windows XP および Internet Explorer 6 の古いビルドを実行する予備のコンピューターがあります。
* **Important (重要)**: これらのアプリケーションは修正の投資に値しますが、指定された上限に達すると、投資は停止され、フォールバック オプションが追求されます。
* **Mission critical (ミッション クリティカル)**: これらのアプリケーションは、当面は重要な役割を果たすことを前提に、標準への準拠や将来に向けた準備などに関する大規模な投資の対象になります。

## Internet Explorer 9 構成検討

Internet Explorer 9 構成検討では、テストで利用する Internet Explorer 9 の構成を検討します。

実際のテストを行う前に、Internet Explorer 9 の構成を検討するのは以下の理由があります。

* セキュリティ強化に伴う互換性問題の回避  
  Internet Explorer はバージョンが上がるにつれて、既定のセキュリティ レベルが強化されています。  
  Internet Explorer 6 と同じ構成で利用していたつもりでも、Internet Explorer 9 の既定のセキュリティ強化による影響で、従来は問題のなかった操作やアドオンの利用が制限を受ける可能性があります。  
    
  セキュリティ強化に伴う互換性問題の例を以下に挙げます。
  + ActiveX コントロールが動作しない
  + Add-on が正常に動作しない
  + ポップアップ ブロックにより、ウィンドウ表示がブロックされる など

こういった問題を事前に回避する意味からも、Internet Explorer 9 の構成を事前に検討することは必須作業となります。

* 画面描画 (レンダリング) に伴う互換性問題の回避  
  Internet Explorer 9 はインターネット標準仕様に準拠していますが、標準準拠の影響で一部の HTML や CSS、JavaScript の解釈において Internet Explorer 6 との互換性が確保できていない部分があります。  
  しかし、Internet Explorer 9 では下位バージョンとの互換性確保と、Web 最新技術への対応を両立させるために、内部的に複数のレンダリング エンジンを搭載しています。  
  具体的には、Internet Explorer 5 以前、Internet Explorer 7, Internet Explorer 8 で利用していた複数のレンダリング エンジンを搭載しており、設定や HTML の指定によりレンダリング エンジンを切り替えることが可能です。  
  これにより、Web サイトや HTML ページ毎に最適なレンダリング エンジンを切り替えることで、互換性問題を回避もしくは低減することが可能になります。
* インターネットで不特定多数が利用する Web アプリケーションと異なり、企業など組織内の Web アプリケーションでは利用者の Internet Explorer の構成をグループ ポリシー、IEAK などを利用して既定の構成を変更する事が可能です。セキュリティやレンダリングに伴う互換性問題を意識して、Internet Explorer の構成を検討することで、テスト時における問題発生件数を減少させることが可能です。

なお、具体的な設計の検討項目についての詳細は「3検証前の準備 (IT 管理者向け)」 で説明します。

## パイロット テスト

パイロット テストは、本格的な Web アプリケーションのテストを実施する前に、小規模なテストチームを編成して、実際に Internet Explorer 9 を利用して主要な Web アプリケーションのテストを行います。

テスト チームは IT 部門のスタッフやミッション クリティカルに位置づけられた Web アプリケーションの担当者により、実施されます。

パイロット テストの目的は Internet Explorer 9 における互換性問題の発生状況を大まかに把握することが一番の目的です。組織内の Web アプリケーションにおいて、互換性問題が発生しないのか、軽微な影響だけなのか、全く動作しないのかなど、互換性問題の影響を本格テストの実施前に把握します。

パイロット テストで良好な結果が得られた場合には、組織では不安なくアプリケーションとブラウザーの互換性の最終テストを行ってもらうことができます。

パイロット テストで不合格になったアプリケーションについては、IT 担当者がアプリケーション担当者と協力し、正式なテストを追加で実行して、問題を見つけて分類し、適切な修正を適用する必要があります。

問題の内容によっては、Internet Explorer 9 構成の再検討、またはテスト方法やテスト計画の見直しが必要になる場合もあります

## テスト

優先順位に従い Web アプリケーションのテストを開始します。

各業務システムのテストケースに従い、画面レイアウトの乱れや、動作上の問題が無いか確認を進めます。

Internet Explorer 9 の変更点に伴う検証ポイントを「4.検証ポイント (開発者向け)」を参照してください。

## 回避策検討

テスト時に互換性問題が検出された Web アプリケーションや Web ページについて、回避策を検討します。

Web アプリケーションのテストで問題が報告された場合、HTML や JavaScript などソース コードの修正だけが唯一の回避策ではありません。セキュリティ設定を変更する、互換表示設定を活用するなどの回避方法も選択することができます。

互換性問題の原因を追及し、改修コストやスケジュールの影響を踏まえて、それぞれの問題について最適な回避策を導き出す必要があります。

一般的には Web アプリケーションの互換性問題の回避策としては、以下のアプローチが挙げられます。

* Internet Explorer 9 構成変更
* 互換表示設定の利用
* ソースコードの修正
* 仮想化テクノロジの利用

### Internet Explorer 9 構成変更

セキュリティ強化に伴い互換性問題が発生しているようなケースにおいては、Internet Explorer 9 の構成を変更する、もしくはセキュリティ レベルを変更する (既定値から下げる) などの対応が考えられます。これにより、ソース コードの修正無く、設定を変更することで Web アプリケーションを動作させることが可能になります。

なお、セキュリティ レベルを既定のレベルから下げる場合には、設定項目によってはセキュリティ リスクやレベルを下げることの影響を慎重に判断する必要があります。

### 互換表示設定の利用

Internet Explorer 9 で互換表示を有効にすると Web サイトや Web アプリケーションの問題が解決される可能性があります。Internet Explorer 9 で互換表示を有効にした場合、Internet Explorer 7 互換のレンダリング エンジンが利用されます。Internet Explorer 7 互換のレンダリング エンジンは Internet Explorer 9 よりも Internet Explorer 6 との互換性が高いため、表示の乱れや画面描画 (レンダリング) による互換性問題を解決できる場合があります。

互換表示設定を利用する際には、以下のアプローチがあります。

* ローカル イントラネットで互換表示を有効にする
* X-UA-Compatibility Meta タグ の利用
* X-UA-Compatibility HTTP ヘッダーの利用

互換表示設定の詳細は「4.検証ポイント (開発者向け)」を参照してください。

### ソースコードの修正

問題の発生箇所が特定できており、且つソースコードの修正箇所が明確な場合には、ソースコードを修正することが最善のアプローチです。このように簡易に修正できる場合には、積極的にソース コードの修正を行うべきです。

ただし、修正箇所が膨大であったり、難易度が高くソースコードの修正に多額の費用が発生する場合もあります。このようなケースではソースコードの修正ではなく、後述の仮想化テクノロジの利用が望ましい場合があります。

ソースコードの修正に掛かる費用と後述する仮想化の運用コストを比較して、どちらが最適な回避策か十分に検討する必要があります。

### 仮想化テクノロジの利用

Internet Explorer 6 から最新バージョンの Internet Explorer への改修費用が膨大になる、スケジュール的な問題で一時的にでも Internet Explorer 6 環境を残す必要がある場合、仮想化テクノロジを利用して一時的に対応することができます。

一時的としているのは、仮想化による対応と並行して Web アプリケーションを新しいバージョンの Internet Explorer でネイティブに実行できるように修正する作業を進める必要があるためです。

Microsoft では、Internet Explorer 6 を一時的に動作させるための選択肢として、MED-V、Windows XP モード、およびターミナル サービスの 3 つの仮想化方式を提供しています。

#### Microsoft Enterprise Desktop Virtualization（MED-V）によるデスクトップの仮想化

この方法は、次期OS基盤上の仮想環境としてWindows XPを動作させ、その仮想化されたWindows XP上で既存のアプリケーションを動作させる事で、移行に伴うOSとアプリケーション間の互換性の問題を解消します。

MED-Vによるデスクトップ仮想化は、エンタープライズの環境に最適化されたデスクトップの仮想化テクノロジで、仮想環境の集中管理機能を搭載している為、仮想環境のデスクトップ、ロックダウン等のポリシーの一元管理が可能です。

MED-Vによるデスクトップの仮想化イメージ



前提条件：なし

メリット：

基本的にアプリケーションの改修の必要がなく、アプリケーションのライフサイクルへの影響がない。

オフラインでのアプリケーション利用ができる。

仮想OSの集中管理ができる。

デメリット：

MED-Vに対応していない特殊デバイスは動作しない可能性がある。

OS (物理/仮想) の2重管理による管理負荷が増加する。

物理PCにWindows XP環境を追加構築する為のスペックが必要となる。

管理サーバーの新規構築による投資が必要となる。

#### Windows XP Modeによるデスクトップの仮想化

この方法は、仮想環境としてWindows XPを動作させ、その仮想化されたWindows XP上で既存のアプリケーションを動作させる事で、移行に伴うアプリケーション間の互換性の問題を解消します。

Windows XP Modeによるデスクトップ仮想化は、比較的中小規模の環境に適したテクノロジで、仮想環境の集中管理機能を持っていない為、各PCの利用ユーザーによる仮想環境の管理が必要となります。

Windows XP Modeによるデスクトップの仮想化イメージ

前提条件：なし

メリット

基本的にはアプリケーションの改修の必要がなく、アプリケーションのライフサイクルへの影響がない。オフラインでのアプリケーション利用ができる。

デメリット

Windows XP Modeに対応していない特殊デバイスは動作しない可能性がある。

OS (物理/仮想) の2重管理による管理負荷が増加する。

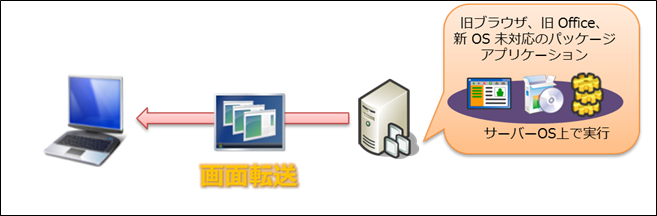
物理PCにWindows XP環境を追加構築する為のスペックが必要となる。

仮想OSを集中管理する仕組みがない。

#### Terminal Serviceによるプレゼンテーションの仮想化

この方法は、Windows XP上で動作している業務アプリケーションを集中管理されたサーバー上（Windows Server 2003）で稼働させ、ユーザーはリモートでアプリケーションを利用させる事で、移行に伴うOSとアプリケーション間の互換性の問題を解消します。

すべてのアプリケーションの処理は、ターミナル サービスを実行するサーバーOS上で集中的に実行されます。ユーザー セッションは互いに分離され、クライアントとサーバーの間では、キーボードやマウス入力などのプレゼンテーション情報や、ビデオの更新のみがクライアントへ送信されます。



Terminal Servicesによるアプリケーションの仮想化イメージ

前提条件：

ターミナルサービス（マルチセッション）で利用する事にライセンス上の問題がない事

業務アプリケーションがサーバーOS上で稼働する事ができる事

業務アプリケーションがマルチユーザーセッションによる利用を前提としたつくりになっている事

メリット：

基本的にアプリケーションの改修の必要がなく、アプリケーションのライフサイクルへの影響がない。

Virtual Desktop Infrastructureによる仮想化に比べ、比較的小規模な環境で実装ができる。

デメリット：

クライアントとサーバー間のネットワーク接続が必須要件となる。（オフラインでの利用はできない）

サーバーの新規構築に伴う投資が必要となる。

ビデオ/オーディオ/外部デバイスのパフォーマンスは物理環境より低下する。

## 対応実施

回避策検討のプロセスで検討した、互換性問題に対する対応を実施します。

# 検証前の準備 (IT 管理者向け)

「2.4. Internet Explorer 9 構成検討」で説明したように、Web アプリケーションの検証を行う前に、Internet Explorer 9 の構成を定義することが、互換性問題を事前に回避する上でも重要なことを説明しました。

ここでは、IT 管理者が検討すべき Internet Explorer 9 の構成ポイント、セキュリティ上の変更点について説明します。

## Internet Explorer 9 構成検討 ポイント

### セキュリティ ゾーン

セキュリティ ゾーンとは Web サイトに対して適切なレベルのセキュリティを提供するための構成情報です。

Internet Explorer は接続先に応じてセキュリティ ゾーンを識別して、定義されたセキュリティ設定に従って動作します。Internet Explorer 6, Internet Explorer 9 共に次の 5つのセキュリティ ゾーンが準備されています。

* インターネット ゾーン
* ローカル イントラネット ゾーン (※ Internet Explorer 6 では “イントラネット” と表記)
* 信頼済みサイト
* 制限付きサイト
* ローカル コンピューター ゾーン (※ レジストリやグループ ポリシーで定義)

各セキュリティ ゾーンの規定のセキュリティ レベルの比較を以下の表に示します。

Internet Explorer 9のセキュリティ ゾーンでは、従来のInternet Explorer 6と比べ、既定の設定がよりセキュアとなっています。Internet Explorer 6のセキュリティレベルでは「信頼済み」ゾーンのセキュリティ設定が最も低く設定されていました。しかしInternet Explorer 9では「ローカル イントラネット」ゾーンのセキュリティ設定が最も低く設定されています。

また、Internet Explorer 6、9それぞれで最もセキュリティ設定が低いゾーン同士を比較しても、Internet Explorer 9のセキュリティ設定はInternet Explorer 6に比べると厳しくなっています。具体例を挙げると、Internet Explorer 6では「信頼済み」ゾーンのセキュリティレベルが 「低」であるのに対し、Internet Explorer 9ではセキュリティレベルが「中」で設定されていることからも確認することができます。

表 Internet Explorer 6/9におけるセキュリティレベルの比較

|  |  |  |
| --- | --- | --- |
| セキュリティレベル  (上に行くほどセキュリティが強固) | Internet Explorer 9 | Internet Explorer 6 |
| 高 | 制限付き | 制限付き |
| 中高 | インターネット | - |
| 中 | 信頼済み | インターネット |
| 中低 | ローカル イントラネット | イントラネット |
| 低 | - | 信頼済み |

Internet Explorer 9 においては、組織内の Web サイトや Web アプリケーションを「ローカル イントラネット」ゾーンとして判定させることが非常に重要な意味を持ちます。

例えば、組織内の Web アプリケーションが 「イントラネット」や「信頼済」サイトとして判定された場合、Internet Explorer 6 よりも高いセキュリティ 設定が Internet Explorer 9 に適用されるため、アドインやコンポーネントの動作が制限されます。

Internet Explorer 9 では、組織内の Web アプリケーションは「ローカル イントラネット」としてセキュリティ ゾーンが判定されるように事前構成を行います。

参考資料: Internet Explorer セキュリティ ゾーンの説明

Internet Explorer では次の 4 つのセキュリティ ゾーンが利用できます。

* インターネット ゾーン  
  インターネット ゾーンは、他のゾーンに含まれない全ての Web サイトが適用されます。
* ローカル イントラネット ゾーン  
  一般的には企業内ネットワーク上の Web サイトで利用するセキュリティ ゾーンです。  
  明示的に DNS のサブドメインやホスト名を登録することで、ローカル イントラネット ゾーンとして識別する Web サイトを構成できます。
* 信頼済みサイト  
  信頼済みサイトとは、Web サイトの運営者が信頼でき、且つ インターネット ゾーンでの利用において問題がある場合に利用するセキュリティ ゾーンです。  
  既定では信頼済み サイトはインターネット ゾーンよりもセキュリティ設定が低いため、インターネット ゾーンとして識別されたサイトでセキュリティ設定の問題で動作しない Web サイトを信頼済みサイトに構成することで問題を回避することが可能になります。
* 制限付きサイト  
  制限付きサイトは非常に高いセキュリティ レベルが構成されており、最小限の操作のみが許可されています。この制限付きサイトは、信頼されていない Web サイトの利用時に構成することを想定しています。

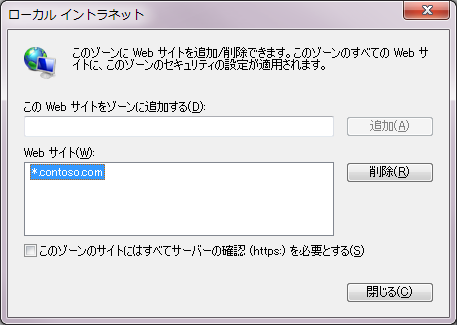
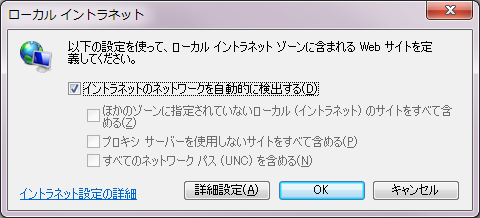
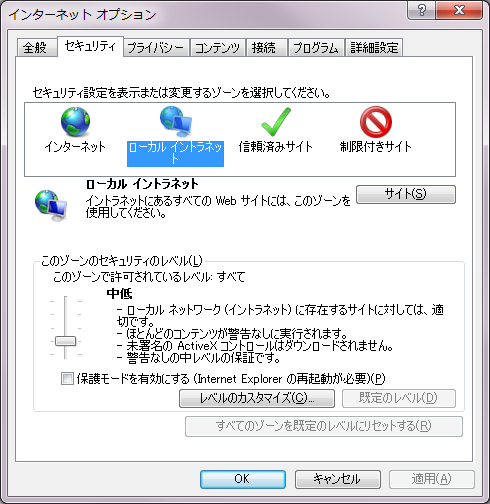
### 「ローカル イントラネット」ゾーン 判定

Internet Explorer 9 における「ローカル イントラネット」ゾーンの構成方法について説明します。

「ローカル イントラネット」の構成を行うには、ツール メニューより＜インターネット オプション＞を選択し、”セキュリティ” タブを選択します。

「ローカル イントラネット」を選択し、”サイト” ボタンをクリックすると、”ローカル イントラネット”のダイアログが現れます。ここで” 詳細設定” ボタンを押してください。

Web サイトをゾーンに追加できるダイアログが表示されますので、組織で利用する Web サイトのドメイン サフィックスや FQDN 名、IP アドレスなどを登録します。



Internet Explorer 9のローカル イントラネットの識別設定では「イントラネットのネットワークを自動的に検出する」が有効になり、Internet Explorer 6と比較してローカル イントラネット ゾーンの判定方法がより厳格に行われるようになっています。しかし、上記のように明示的にドメイン サフィックスなどを登録することで、管理者が「ローカル イントラネット」ゾーンとして識別すべきサイトを明示的に構成することができます。

### 「ローカル イントラネット」ゾーン セキュリティ 設定

ここまでの検討で、組織内の Web サイトが「ローカル イントラネット」ゾーンとして認識されるように構成しました。続いて、「ローカル イントラネット」ゾーンのセキュリティの詳細設定値について、定義します。

セキュリティ設定の詳細を検討する際のアプローチは以下の通りです。

1. Internet Explorer 6 の「イントラネット」セキュリティ設定値の確認  
   Internet Explorer 6 の「イントラネット」サイトのセキュリティ設定を確認し、標準設定から変更している箇所をピックアップしてください。  
   設定変更の有無が分からない場合には、「バージョン別 セキュリティ設定値一覧」を参考にして、標準設定から変更している箇所を洗い出してください。可能であれば、変更している理由 (セキュリティ上の理由、アプリケーションの動作要件など) が分かれば合わせて確認してください。
2. Internet Explorer 6 「イントラネット」 と Internet Explorer 9 「ローカル イントラネット」のセキュリティ設定値の比較  
   1 で抽出した「イントラネット」のセキュリティ設定情報と、Internet Explorer 9 の「ローカル イントラネット」のセキュリティ設定値を全て比較してください。  
   何カ所か設定値が異なる箇所があると思います。設定値がずれている項目について、項目の意味や過去の変更理由などを考慮して、Internet Explorer 9 の「ローカル イントラネット」のセキュリティ設定値を検討します。  
   ※ アプリケーションの動作を優先させるのであれば、Internet Explorer 6 の設定値をそのまま Internet Explorer 9 の構成値とします。
3. Internet Explorer 9 「ローカル イントラネット」のセキュリティ設定値の決定  
   2 の検討結果を踏まえて、Internet Explorer 9 の「ローカル イントラネット」のセキュリティ 設定値を決定します。既定値から変更した箇所は、変更の経緯が分かるように設計書などに明記しておくことを推奨します。

参考資料: Internet Explorer 各バージョン セキュリティ設定値一覧

バージョン別セキュリティ既定値一覧表

<http://msdn.microsoft.com/ja-jp/ie/cc844005>

### 詳細設定オプション

最後に Internet Explorer 9 全体の動作を制御する詳細設定オプションを定義します。

詳細設定オプションの設定値を検討する際のアプローチは以下の通りです。

1. Internet Explorer 6 の「詳細設定オプション」の確認  
   Internet Explorer 6 の「詳細設定オプション」を確認し、標準設定から変更している箇所をピックアップしてください。  
   設定変更の有無が分からない場合には、「バージョン別 セキュリティ設定値一覧」を参考にして、標準設定から変更している箇所を洗い出してください。可能であれば、変更している理由 (セキュリティ上の理由、アプリケーションの動作要件など) が分かれば合わせて確認してください。
2. Internet Explorer 6 と Internet Explorer 9 の「詳細設定オプション」の比較  
   1 で抽出した Internet Explorer 6 の「詳細設定オプション」の定義情報と、Internet Explorer 9 の定義情報を比較してください。  
   設定値がずれている項目について、項目の意味や過去の変更理由などを考慮して、Internet Explorer 9 の「詳細設定オプション」の設定値を検討します。  
   ※ 設定に迷う場合には、Internet Explorer 9 の既定の状態としてください。
3. Internet Explorer 9 「詳細設定オプション」設定値の決定  
   2 の検討結果を踏まえて、Internet Explorer 9 の「詳細設定オプション」の設定値を決定します。  
   既定値から変更した箇所は、変更の経緯が分かるように設計書などに明記しておくことを推奨します。

詳細設定オプションで構成するセキュリティ関連の項目について、補足しておきます。

#### SSL 3.0 / TLS 1.0 対応

Web サーバーで SSL を利用している場合、Internet Explorer 9 からSSL を利用してアクセスできるか確認してください。これは、Internet Explorer 6 と 比較して Internet Explorer 9 では SSL の既定値が異なります。SSL v2 にしか対応していない Web サーバーなどを利用している場合には、Internet Explorer 9 の既定の状態で SSL 通信が行えるか確認を行ってください。

SSL で通信できない場合には SSL 設定値の見直し、または Web サーバーの SSL 設定の変更を検討してください。

#### 信頼されていないサーバー証明書の利用など

SSL 利用時に信頼された証明機関からサーバー証明書が発行されていない、または有効期限が切れている証明書を利用しているといった場合には、SSL 接続時にユーザーに対して警告メッセージが表示されます。

これまでの Internet Explorer 6 でも警告メッセージが表示されて、ユーザーが警告内容を承諾する必要がありましたが、Internet Explorer 9 では警告メッセージの表示方法などが変更されています。

警告メッセージの表示方法などがユーザーの操作性などに影響をきたす場合には、信頼されたサーバー証明書を利用するなど、対応を検討してください。

#### サーバー証明書の失効確認 (OCSP)

Internet Explorer 9 では OCSP プロトコルのサポートが追加され、サーバー証明書の失効確認が既定で有効になっています。

OCSP では証明機関が指定する証明書の失効リスト (CRL: Certificate Revocation Lists) にアクセスして、証明書の失効を確認します。このため、インターネットへのアクセスが制限されている環境などでは、CRL の配布ポイントにアクセスできないために、SSL 接続時に遅延が発生するなどの影響が懸念されます。

Internet Explorer 9 端末がインターネットにアクセスできる、CRL 配布ポイントにアクセスできることを確認してください。

#### Proxy サーバーの HTTP 1.1 対応

Proxy サーバーを運用している場合、Internet Explorer 9 では既定で HTTP 1.1 で通信を行います。

Proxy サーバーが HTTP 1.1 に対応していることを確認してください。

また、必要に応じて Proxy サーバー経由のテストの必要性を考慮し、テストケースとして Proxy サーバー経由のテストを含めることを検討してください。

参考資料: Internet Explorer 各バージョン セキュリティ設定値一覧

バージョン別詳細設定一覧

<http://msdn.microsoft.com/ja-jp/ie/cc844005?lc=1041#IE_SecuritySettingsDetailed_by_version>

## 互換表示 設定

### 互換表示 設定とは

Internet Explorer 9 には過去のバージョンとの互換性を確保するために、HTML の描画を行う描画エンジンを複数 搭載しています。既定では Internet Explorer 9 は最新の HTML 5 などに対応した “IE9 Standards” と呼ばれる動作モードが選択されます。

しかし、"IE9 Standards” モードは最新の HTML 5 などの標準規格に対応している反面、Internet Explorer 6 を前提にした HTML や CSS, JavaScript とは幾つかの相違点があり、完全な互換性が確保されていません。また、Internet Explorer 6 を利用している組織の Web アプリケーションでは最新の HTML 5 などを利用しておらず、従来の HTML や CSS が Internet Explorer 6 と同様に動作することが期待されています。

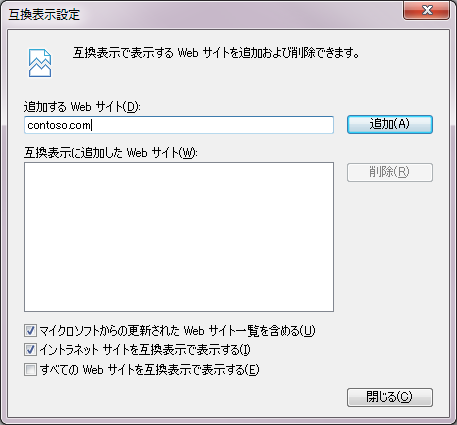
Internet Explorer 6 との互換性を確保するために、Intenet Explorer 9 では「Internet Explorer 7 互換モード」という機能が用意されています。Internet Explorer 7 互換モードとして動作した場合、描画エンジンは Internet Explorer 6 と互換性が高い Internet Explorer 7 のエンジンが利用されます。

これにより、Internet Explorer 6 向けの Web アプリケーションを Internet Explorer 9 でも互換性を確保して動作させることが可能になります。

### 互換表示 設定方法

互換表示を設定するには、メニューより ＜ツール＞－＜互換表示設定＞を選択します。

互換表示設定のダイアログが表示されるので、”イントラネット サイトを互換表示で設定する” のチェックボックスを確認してください。



“イントラネット サイトを互換表示で表示する” が有効になっている場合 (Internet Explorer9 既定で有効)、「ローカル イントラネット」サイトにアクセスする際には互換表示設定が有効になります。

これは、「ローカル イントラネット」サイト つまり、組織内の Web アプリケーションにアクセスする際には Internet Explorer 9 互換表示 (Internet Explorer 7 モード)が利用されると言うことを意味します。

繰り返しになりますが、これまで Internet Explorer 6 を利用している組織の Web アプリケーションの大部分は HTML 5 などの最新規格を利用しておらず、Internet Explorer 6 と同様に Internet Explorer 9 が動作することが求められています。

Internet Explorer 9 の互換表示と「ローカル イントラネット」ゾーンの構成を組み合わせることで、組織の Web アプリケーションに対して Internet Explorer 6 と互換性の高い Internet Explorer 7 モードでアクセスすることが可能になります。

これにより、大幅に Internet Explorer 9 における互換性問題を低減することが可能です。

## IEAK

Internet Explorer のセキュリティ ゾーン、インターネット オプションなどの構成情報が決定したら、それらの情報を検証機にどのように配布するのか検討を行います。

構成情報が Internet Explorer の既定の構成からほとんど変更がない場合には、テスターに変更箇所を伝える、もしくは手順書を配布して対応するといったアプローチでも可能です。

しかし、構成情報が既定値から大きく変更されている場合には、構成情報を効率よく検証機に配布するための方法について検討します。

* IEAK (Internet Explorer Administration Kit)

Internet Explorer の構成情報をセットアップ パッケージとして配布できる IEAK とよぶツールがあります。

本ツールを利用することで、管理者が構成した Internet Explorer の構成情報をセットアップ形式で配布することが可能になるため、テスターは個々の設定を変更するのではなく、セットアップツールを実行することにより、構成情報を容易に適用することが可能になります。

* Windows 7 のキッティング

Windows 7 のキッティング手法を利用して、Internet Explorer 9 および Windows 7 の構成を展開する方法です。

OS 構成と Internet Explorer 9 の構成を全て含めて検証機の構成を準備できるため、検証機を新規に調達するような場合には有効な手段になります。

# 検証ポイント (開発者向け)

Internet Explorer 6からInternet Explorer 9への移行時の検証ポイントは、主として、画面表示、JavaScript、セキュリティ、ActiveXコントロール、その他（新機能など）があげられます。

また、Internet Explorerには既存のバージョンと互換性を保つため、いくつかのモードがあります。

このため、検証を行うにあたり既存のアプリケーションがInternet Explorer 9のどのモードで動くか、またどのモードで動かすかによりその結果が異なります。

まず、具体的な検証のポイントを記述する前にInternet Explorer 9の互換性機能について簡単に説明します。

## ブラウザーモード、ドキュメントモード

これは、Internet Explorer 8から採用された機能です。

ブラウザーモードは、ブラウザーがどのバージョンとして振る舞うかを決定する機能です。具体的には、以下の３点についてそのモードごとに振る舞いが異なります。

　・サーバーに送信するUser-Agent文字列

　・バージョン ベクター

　・デフォルトのドキュメントモード

* サーバーに送信するUser-Agent文字列

Internet Explorer が自信を識別するためにWebサーバーに送信する値（文字列）です。

サーバーアプリケーションの作りにもよりますが、この値を基にアプリケーションの動作、作成するページをクライアントのブラウザーに合わせて行います。

例：Mozilla/5.0(compatible; MSIE 9.0; WindowsNT6.1; Trident 5.0; …)

詳細については、以下のサイトなどを参照してください。

ユーザー エージェント文字列を理解する

<http://msdn.microsoft.com/ja-jp/library/ms537503(v=vs.85).aspx>

* バージョン ベクター

バージョン ベクターは、ブラウザーの起動時に読み取られるレジストリ キーに格納されている、Internet Explorerの内部バージョン番号を表します。

この値は、条件付きコメントを処理するときなどに使用されます。

条件付きコメントは、ページのHTMLソース内のコメントのブロックであり、Internet Explorerでのみ解釈されます。

例：

　<!—[if IE 9]>

　～

　<![endif]-->

詳細については、以下のサイトなどを参照してください。

　バージョン ベクター

　http://msdn.microsoft.com/ja-jp/library/cc817577.aspx

　条件付きコメント

　http://msdn.microsoft.com/ja-jp/library/ms537512(v=VS.85).aspx

* ドキュメントモード

ブラウザーのレンダリングエンジンなどを切り替える機能です。

モードの種類については、後述します。

ブラウザーモードは、Internet Explorer 9では以下のようになっています。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | 種類 | User－Agent文字列 | バージョン ベクター | ドキュメントモード | 備考 |
| 1 | IE9 | Mozilla/5.0(compatible; MSIE 9.0; WindowsNT6.1; Trident 5.0; …) | IE 9 | IE9 Standards | Internet Explorer 9の既定のモード |
| 2 | IE9 互換表示 | Mozilla/4.0(compatible; MSIE 7.0; WindowsNT6.1; Trident 5.0; …) | IE 7 | IE7 Standards |  |
| 3 | IE8 | Mozilla/4.0(compatible; MSIE 8.0; WindowsNT6.1; Trident 4.0; …) | IE 8 | IE8 Standards |  |
| 4 | IE7 | Mozilla/4.0(compatible; MSIE 7.0; WindowsNT6.1; …) | IE 7 | IE7 Standards |  |

ドキュメントモードは、Internet Explorer 9では、以下のようになっています。

|  |  |  |
| --- | --- | --- |
| No. | 種類 | 説明 |
| 1 | IE9 Standards | 最新の規格に標準準拠した動作モード。  デフォルトのドキュメントモード。  HTML5、CSS3、ECMAScript5、DOM L2/L3などに対応。 |
| 2 | IE8 Standards | IE8の標準のドキュメントモードと互換性を持ったモード。  CSS2.1に対応している。 |
| 3 | IE7 Standards | IE7の標準のドキュメントモードと互換性を持ったモード。  CSS1.0に対応している。 |
| 4 | Quirks | IE5.0、5.5、6.0と互換性を持ったドキュメントモード。 |

### DOCTYPE宣言とIE9のドキュメントモードについて

　Internet Explorer は、DOCTYPE宣言により標準モード、Quirksモードを切り替えて表示します。

IE6とIE9のDOCTYPE宣言に対応したドキュメントモードは、以下のようになります。

表：DOCTYPE宣言とドキュメントモード

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | DOCTYPE宣言 | | | IE6のモード | IE9のモード |  |
| 1 | なし | | | Quirks | Quirks | DOCTYPEの宣言なし |
| 2 | HTML 4.01 | Transitional | システム識別子なし | Quirks | Quirks | <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> |
| 3 | システム識別子あり | IE6標準 | IE9標準 | <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> |
| 4 | Strict | システム識別子なし | IE6標準 | IE9標準 | <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"> |
| 5 | システム識別子あり | IE6標準 | IE9標準 | <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"> |
| 6 | XHTML 1.0 | Transitional | XML宣言なし | IE6標準 | IE9標準 | <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> |
| 7 | XML宣言あり | Quirks | IE9標準 | <?xml version="1.0" encoding="文字エンコーディング"?>  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> |
| 8 | Strict | XML宣言なし | IE6標準 | IE9標準 | <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> |
| 9 | XML宣言あり | Quirks | IE9標準 | <?xml version="1.0" encoding="文字エンコーディング"?>  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> |
| 10 | XHTML 1.1 |  | XML宣言なし | IE6標準 | IE9標準 | <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"> |
| 11 | XML宣言あり | Quirks | IE9標準 | <?xml version="1.0" encoding="文字エンコーディング"?>  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"> |

※DOCTYPEの変更を行わずにInternet Explorer 9で表示した場合、ほとんどのケースでドキュメントモードがIE9標準となります。このため、互換性のないモードで動作するので画面の崩れやJavaScriptの処理結果が異なることが考えられます。DOCTYPE宣言を行っていないページでは、Quirtksモード（後方互換モード）になる場合が多くなります。

### ドキュメントモードを指定（変更）について

　ドキュメントモードを指定（変更）する方法には、以下の３通りがあります。

* metaタグで指定する
* HTTPレスポンスヘッダで指定する
* DOCTYPEで指定する
* Metaタグで指定する

以下のようなmetaタグをヘッダに記述することで、ドキュメントモードを指定することができます。

例：

　<meta http-equiv="X-UA-Compatible" content="IE=5">

　<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7; IE=EmulateIE9">

互換モードの概要は以下になります。

|  |  |  |
| --- | --- | --- |
| No. | 互換モード値 | レンダリング動作 |
| 1 | IE=5 | Quirks（クワークス）モード |
| 2 | IE=7 | Internet Explorer 7標準準拠モード |
| 3 | IE=EmulateIE7 | !DOCTYPE宣言によりモードが決定されます  ・Quirksモードの!DOCTYPE宣言の場合はQuirksモード  ・標準モードの!DOCTYPE宣言の場合はInternet Explorer 7標準準拠モード |
| 4 | IE=8 | Internet Explorer 8標準準拠モード |
| 5 | IE=EmulateIE8 | !DOCTYPE宣言によりモードが決定されます  ・Quirksモードの!DOCTYPE宣言の場合はQuirksモード  ・標準モードの!DOCTYPE宣言の場合はInternet Explorer 8標準準拠モード |
| 6 | IE=9 | Internet Explorer 9標準準拠モード |
| 7 | IE=EmulateIE9 | !DOCTYPE宣言によりモードが決定されます  ・Quirksモードの!DOCTYPE宣言の場合はQuirksモード  ・標準モードの!DOCTYPE宣言の場合はInternet Explorer 9標準準拠モード |
| 8 | IE=edge | Internet Explorer 9と将来のすべてのバージョンのブラウザーがサポートする最新の標準を使用。実稼働サイトには非推奨。 |

※Quirksモード／標準モードのDOCTYPE宣言については、「表：DOCTYPE宣言とドキュメントモード」を参照してください。

モードの複数指定について

例にもあるように、metaタグでは複数のレンダリングモードを指定することが可能です。

例：<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7; IE=EmulateIE9">

Metaタグをサポートする今後のInternet Explorerでは、指定された複数のモードの中から一番適したモードを選択します。上記の例の場合、IE7互換モードとIE9互換モードを指定しています。これをInternet Explorer 8で表示した場合、IE7互換モードでレンダリングします。

metaタグの位置について

metaタグのX-UA-Compatibleの記述位置についてですが、head要素の先頭の方で書くようにしてください。これは、metaタグのX-UA-Compatibleでドキュメントモードを指定する前に、script要素やstyle要素などをInternet Explorerが読み込むと、この時点で最適と判断したドキュメントモードでこのページを処理（レンダリング）します。つまり、ドキュメントモードが指定する前に確定してしまいます。

そのため、ドキュメントモードが確定した後にmetaタグのX-UA-Compatibleで指定してもInternet Explorerは、無視して処理を続けます。

Head要素の中で、metaタグは先頭の方にまとめて書くことをお勧めします。

* HTTPレスポンスヘッダーで指定する

HTTPレスポンスヘッダーを利用してドキュメントモードを指定することもできます。

この方法はWebサーバー単位やサイト単位、ディレクトリ単位でドキュメントモードを指定するときに便利です。以下のようなメタタグをヘッダに記述することで、ドキュメントモードを指定することができます。

以下のようなヘッダーをクライアントに送信することにより、ドキュメントモードを指定することができます。X-UA-Compatibleヘッダーに指定する値はmetaタグのcontent属性の値と同じです。

例：X-UA-Compatible：IE=5

Microsoft Inernet Information Services（IIS）7.0を構成するには、Web.configファイルを以下のように設定します。

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<system.webServer>

<httpProtocol>

<customHeaders>

<clear />

<add name="X-UA-Compatible" value="IE=7" />

</customHeaders>

</httpProtocol>

</system.webServer>

</configuration>

* DOCTYPE宣言で指定する

DOCTYPE宣言で、標準モードとQuirksモードに切り替えることができます。

DOCTYPE宣言では、他のバージョンの標準モードを指定することはできません。

他のバージョンの標準モードで表示するためには、metaタグ、またはHTTPレスポンスヘッダーで指定します。

DOCTYPE宣言の例については、「表：DOCTYPE宣言とドキュメントモード」を参照してください。

　ドキュメントモードの指定方法の優先度は、以下のようになります。

　Metaタグ　＞　HTTPレスポンスヘッダー　＞　DOCTYPE宣言

　ただし、metaタグやHTTPレスポンスヘッダーのcontentにて、

　　IE=EmulateIE9、IE=EmulateIE8、IE=EmulateIE7

　が、指定されている場合には、DOCTYPE宣言が優先されます。

　（DOCTYPE宣言によりQuirksモードか標準モードかが決まります）

### 検証におけるモードについて

ブラウザモード、ドキュメントモードが異なると、User-Agent文字列、処理を行うレンダリングエンジンやJavaScriptエンジンが異なり、画面表示や処理結果が異なる場合があります。

検証を行っているときに、このモードの違いによる影響により画面表示が壊れたり、処理結果が異なるようなことがあったら、他のモードに切り替えて、互換性が保たれるようなモードを探すことをお勧めします。

これは、移行（migration）において、あるモードを指定することでそのページ（サイト）の互換性が保たれるならば、そのページの移行には、そのモードを利用することをお勧めするからです。

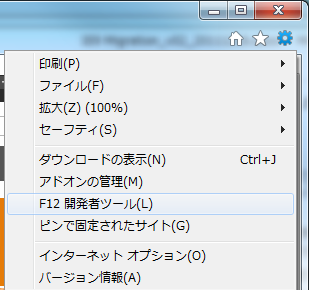
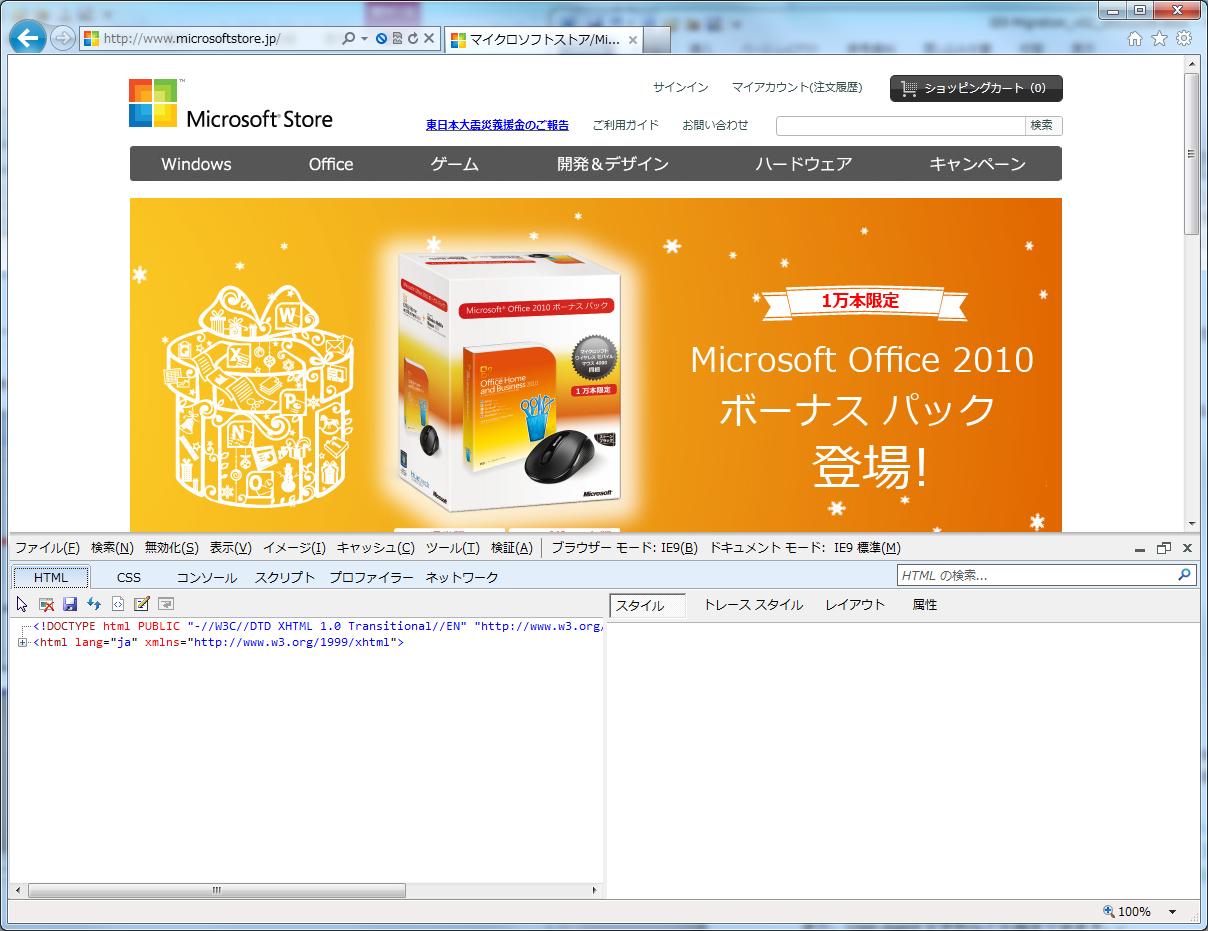
モードを指定することにより互換性が保てるページは、まずはこのような形で移行を行い、その後、そのページを変更（更改）するときに、改めて最新のブラウザーのモードで動作するように作成することをお勧めします。

モードの切り替え方法ですが、F12の開発者ツールでモードを切り替えることができます。

開発者ツール

開発者ツールは、[F12]キーを押すと起動されます。（または、[ツール(alt + X)] － [F12 開発者ツール(L)]）より起動されます。）

　　図　F12開発者ツールメニュー　　　　　　図　F12開発者ツール



開発者ツールのメニューバーにある[ブラウザーモード]、[ドキュメントモード]からそれぞれモードを指定できます。また、現在のモードがメニューバーに表示されています。

図　F12開発者ツールのメニュー

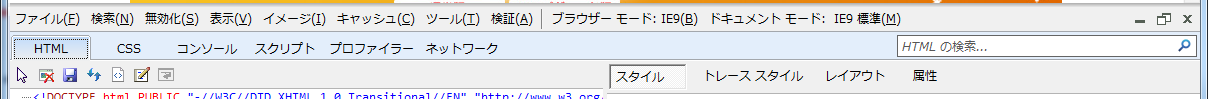


図　F12開発者ツールのブラウザーモードメニュー

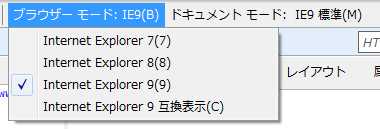
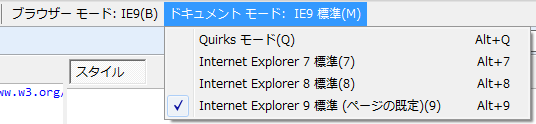
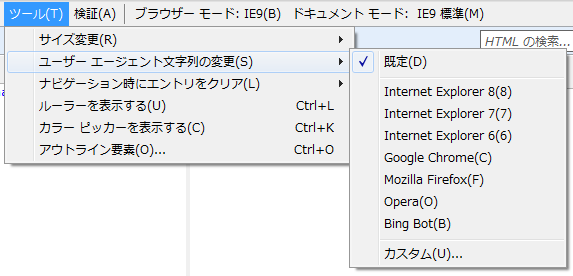


図　F12開発者ツールのブラウザーモードメニュー



　また、User-Agent文字列なども指定できます。

図　ユーザー　エージェント文字列の変更



ここでは、いくつかの基本的な軽減手順について説明し、現在の主要な問題に対処しつつ、将来のアプリケーションの互換性に備えられるようにします。

企業内の Web システムに対するソリューションを検討する際の重要な点は、これまで利用してきた Internet Explorer 6 とこれから導入される Internet Explorer 9 の複数バージョンをサポートする必要がある事です。

一般的に、Windows 7 の導入は段階的に進むことから、端末のリプレースが完了するまでの期間 (数年に及ぶ場合もある) は複数バージョンの Internet Explorer への対応を考慮する必要があります。

互換表示は Internet Explorer 9 の機能 の 1 つで、Internet Explorer 7 とほとんど同じ表示でブラウザーに Web ページをレンダリングできるようにします。

#### 互換表示で問題が解決される場合

Internet Explorer 8 または Internet Explorer 9 で互換表示を有効にすると Web サイトや Web アプリケーションの問題が解決される場合は、グループ ポリシーを使用して互換表示を有効にすることが最善の方法です。次に、Web アプリケーションや Web サイトの次のリリースで、採用されている最新の Web 標準を使用してコードを更新するための計画を立てることをお勧めします。Internet Explorer 8 と Internet Explorer 9 は、製品のライフサイクル全体を通じて Microsoft によってサポートされますが、古いレンダリング モードだけに依存しないことをお勧めします。Meta タグやヘッダーを使用して標準モードへの新しい適応すべて選択する必要もあります。将来の互換性確保のためには、現在採用されている Web 標準に合わせて Web アプリケーションをコーディングする必要があります。

Web 標準は常に進化し、変化しています。そのため、互換性は組織の進展に合わせて取り組まなければならない問題です。ブラウザーは最新の Web 標準に準拠するために常に改良され、準拠の度合いが低下することはありません。したがって、Web ベースのアプリケーションと Web サイトも、ブラウザーの最新の標準に合わせて更新する必要があります。

#### 互換表示で問題が解決されない場合

特に Internet Explorer 6 用にコーディングされている古いアプリケーションの中には、互換表示で動作しないものがあります。これらのアプリケーションについては、Web サイトを Quirks モードに設定するという代替の修正オプションがあります。Quirks モードは、Internet Explorer 5.5 のレンダリング動作とほぼ同等の下位互換モードです。Web サイトが Quirks モードで正しくレンダリングされるかどうかを確認するために、テスターは開発者ツールを使用して Internet Explorer 9 を Quirks モードに設定できます。Quirks モードによってサイトの互換性問題が軽減される場合、IT プロフェッショナルは、グループ ポリシーを使用してそのサイトを Quirks モードのポリシー一覧に追加できます。Quirks モードによって互換性の問題が軽減されない場合は、IT プロフェッショナルはサイトを修正する別の方法を探す必要があります。この方法としては、サイトの開発者に Web サイトの修正を依頼したり、MED-V、ターミナル サービス、または Windows XP モードを使用した次善策として仮想化を使用することが考えられます。

#### 将来の互換性を確保するための Meta タグや HTTP ヘッダーの使用

開発者は、Meta タグの content 属性を使用して、Web ページのコンテンツのレンダリングに使用するモードを指定できます。コンテンツが Internet Explorer 8 標準レンダリング モードで表示されるようにするには、値に IE=EmulateIE8 または IE=8 を指定します。コンテンツが Internet Explorer 9 標準レンダリング モードで表示されるようにするには、値に IE=EmulateIE9 または IE=9 を指定します。コンテンツが Internet Explorer 7 のレンダリング動作を使用して Internet Explorer 9 で表示されるようにするには、値に IE=EmulateIE7 または IE=7 を指定します。コンテンツが Internet Explorer 5.5 (Quirks) のレンダリング動作で表示されるようにするには、値に IE=5 を指定します。

互換性と X-UA-Compatible ヘッダーの詳細については、「[ドキュメント互換性の定義](http://go.microsoft.com/fwlink/?LinkId=214509)」を参照してください。

もちろん、Web サイトには何千もの (あるいは何万もの) ページが含まれることもあるため、この値をドキュメントごとに設定するのは現実的ではありません。サイトのすべてのページ、またはフォルダー単位で選択したページのコレクションに Meta タグを設定してサイトが機能するようになる場合は、サーバーの構成を調整し、HTTP ヘッダーに X-UA-Compatible メタデータを追加することをお勧めします。

次の表は、Internet Explorer 9 のドキュメントの互換性を示しています。

|  |  |
| --- | --- |
| content の値 | 意味 |
| IE=5 | "Quirks" モード。 |
| IE=7 | 常に Internet Explorer 7 モードを使用する。 |
| IE=EmulateIE7 | サイトの DOCTYPE が Quirks の場合を除いて Internet Explorer 7 モードで表示する。 |
| IE=8 | 常に Internet Explorer 8 モードを使用する。 |
| IE=EmulateIE8 | サイトの DOCTYPE が Quirks の場合を除いて Internet Explorer 8 モードで表示する。また、クライアント コンピューターの互換表示を上書きして Internet Explorer 8 モードを使用する。 |
| IE=Edge | 最新のモードで表示する (Internet Explorer 9 では IE=9 と同等)。 |
| IE=9 | 常に Internet Explorer 9 モードを使用する。 |
| IE=EmulateIE9 | サイトの DOCTYPE が Quirks の場合を除いて Internet Explorer 9 モードで表示する。また、クライアント コンピューターの互換表示を上書きして Internet Explorer 9 モードを使用する。 |

## 画面表示

### ユーザー インターフェイスの変更

Internet Explorer 6と9を比較すると、まず最初に感じるのがユーザーインターフェイスの違いになります。

* ウィンドウレイアウトの変更

IE6とIE9では、ルック＆フィールが大きく変わっています。それに伴い、描画領域も変わっております。表示するページの画面レイアウトに問題がないことを確認する必要があります。

確認するときの条件ですが、Internet Explorer 9への移行の場合モニターも同時に変更されることが多いかと思います。モニター、解像度などを移行後の仕様に合わせて確認をしてください。また、表示に問題がある場合は、解像度を変更することで問題が解決できるか確認をしてください。解像度を変更することにより解決できる問題であれば、解像度を変更することをお勧めします。

ただし、解像度を上げると文字が小さく表示されます。そのため「文字が見づらい」という問題が発生する場合もあります。このような場合には、文字サイズを変更することを検討してください。しかし、文字サイズを変更すると、「メニューの文字の位置が変わる」、「[OK]ボタンの表示位置が画面からはみ出す」などといった問題も発生する場合があります。

解像度とあわせて文字サイズを調整して、なるべく問題が起きないような適正な値を設定してください。

図　IE6のイメージ

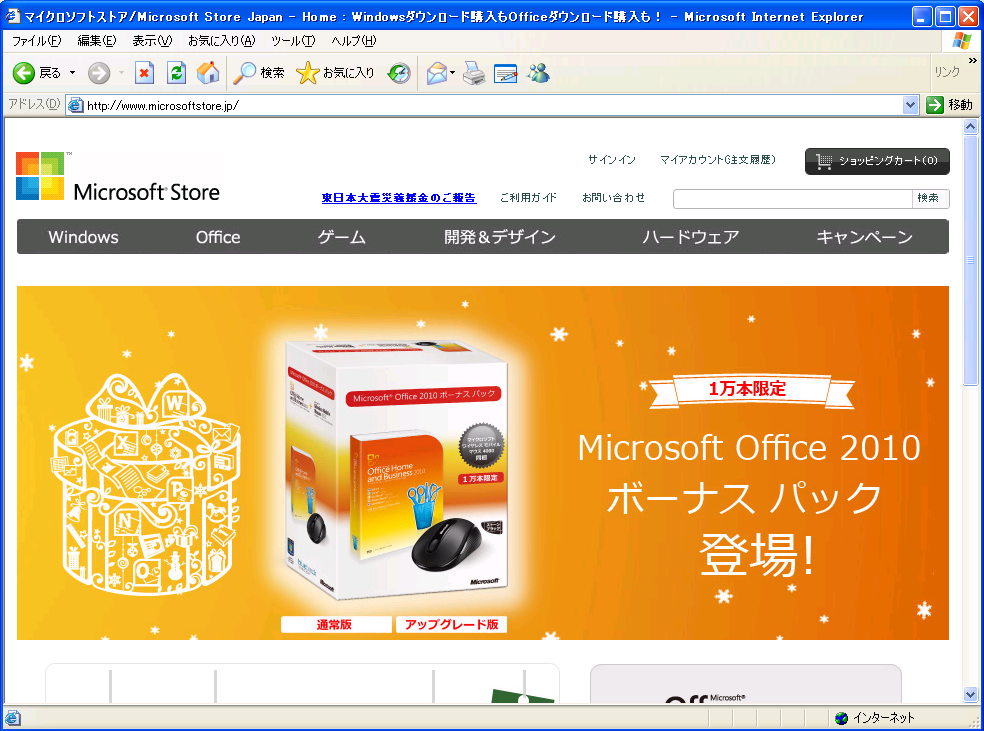


図　IE9のイメージ



タブ表示として不要なメニューは隠しました。

ステータスバーは、非表示にしました。

図 IE9のイメージ（メニュー、ステータスバーを表示）



メニューをすべて表示した状態です。

ステータスバーを表示した状態です。

＜参考資料＞

画面の解像度を変更する

http://windows.microsoft.com/ja-JP/windows7/Change-your-screen-resolution

　画面上のテキストを大きくするまたは小さくする

　http://windows.microsoft.com/ja-JP/windows7/Make-the-text-on-your-screen-larger-or-smaller

* ステータス バーが初期設定では非表示

Internet Explorer 9では、ステータス バーが初期設定では非表示になります。

ステータス バーにメッセージを表示させているアプリケーション（ステータス バーに「次の操作（ナビゲーション）」や「マウスカーソルがフォーカスされているコントロールのヘルプ」などの表示）は、対応が必要になります。

＜対応案＞

　・ブラウザーの設定を変更して、ステータス バーを表示させます。

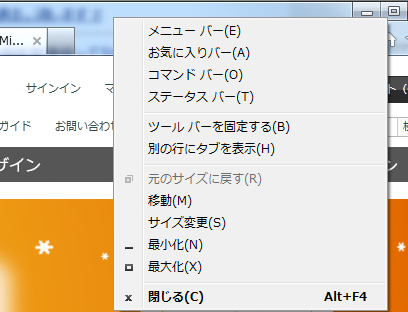
　　例：上部のタイトル バーのあたりを右クリックして、ポップアップ メニューにある

　　　　[ステータス バー(T)]のチェックをオンにする。

　・window.openの時に「status=1」または「status=yes」を指定する。

　・グループポリシーで「ステータス バーを非表示」の設定を「無効」にする。

タイトルバーのあたりを右クリックします。



表示／非表示を切り替えたいメニューを選択します。

* アンカータグ（<A>タグ）の遷移先情報（href属性）の表示

Internet Explorer 9では、ステータス バーが非表示の場合、アンカータグ（<a>タグ）のリンク先情報などが、windowの左下にツールチップで表示されます。

Internet Explorer 6では、ステータス バーが非表示の場合、何も表示されません。

図　IE9 の遷移先情報表示

マウスカーソルが、リンク先情報を持ったアンカータグの上にある状態です。



アンカータグの情報が表示されます。

　　図　IE6の遷移先情報



上図と同様にマウスカーソルが、リンク先情報を持ったアンカータグの上にある状態です。

ステータスバーを非表示の場合、何も表示されません。

* ブラウザーのWindowサイズを固定して表示するアプリケーション

Internet Explorer 9では、アドレス バーのデザインの見直しなどを行ったため、画面描画領域が変更されています。

そのため、画面に余白が発生したりします。

＜対応案＞

　・画面レイアウトの見直しを行う。

* タイトルタグ（<title>タグ）の非表示

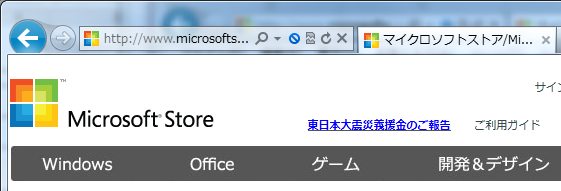
Internet Explorer 9では、タイトル バーにタイトルタグ（<title>タグ）の情報が表示されなくなります。代わりに、タブにその情報は表示されます。

ただし、タブ ブラウズを無効にしてもタイトル バーに情報は表示されません。

タブにtitleタグの情報が表示されているイメージを挿入

タイトルバーには何も表示されません。

図　IE 9のタイトルバー



タブにタイトルタグの内容が表示されます。

図　IE6 のタイトルバー

タイトルバーにタイトルタグの内容が表示されます。



* フォーム部品の変更

Internet Explorer 9では、ボタン表示のLook&Feelが変更されています。

* 適応型ページズーム

この機能は、Internet Explorer 8から採用されています。Internet Explorer 7のズームの動作とは異なります。適応型ページズームは、レイアウト前に要素サイズを変更しているのに対して、IE7のズームはレイアウト後に要素のサイズ変更を行っています。IE 6にはズーム機能はありません。

図　IE9の画面表示（倍率100％）

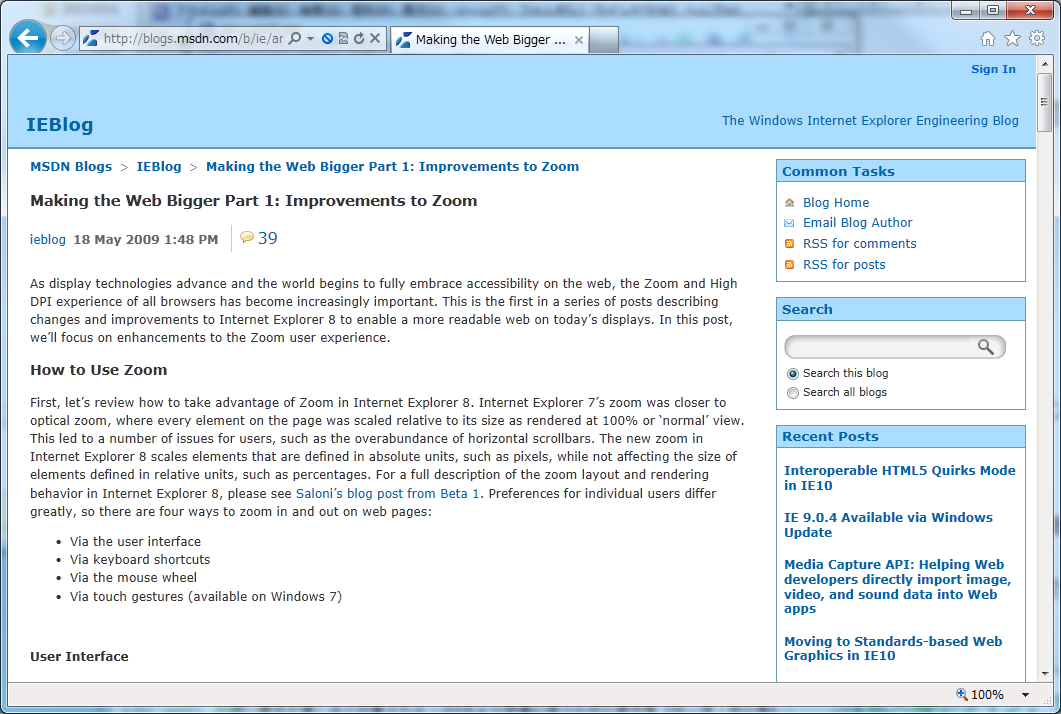
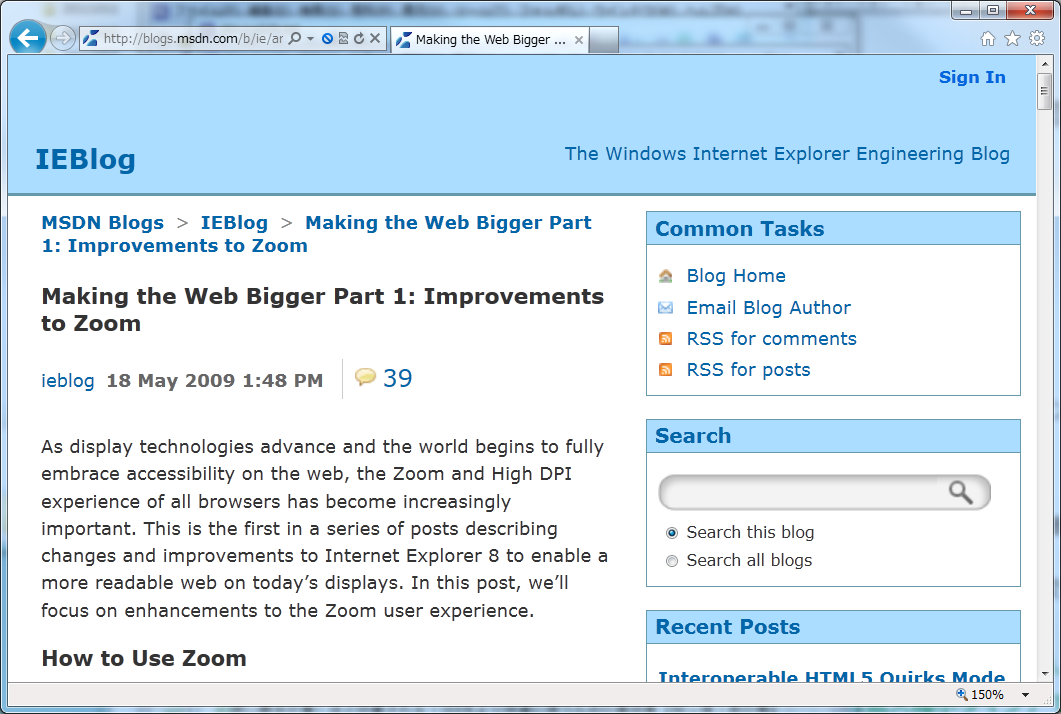


図　IE9の画面表示（倍率 150％）



改行位置などが大きさに合わせて変わっています。

水平スクロールバーは表示されていません。

ただし、上記のようなズームになるためには、自動または相対サイズ指定されている必要があります。絶対単位（例：in、cm、mmなど）またはデバイスとフォントに依存する相対単位（例：px、ex、emなど）の固定サイズで指定されるものは、IE7のズームと同じように拡大縮小されます。

　例：200%ズームの場合

　　　　相対サイズ：　50%　→　50%

　　　　固定サイズ：　50px　→　100px

このためズームにより以下のことが発生する可能性があります。

　例：

　相対サイズ： ズームの倍率を変更した場合、テキストの再表示する前に、利用可能なスペースに

基づいて行の長さを計算し直して、改行を挿入します。

　　　　　　 　ズーム前と改行位置が変わる場合があります。

　　固定サイズ：ズームの倍率を変更しても、行の長さや改行位置は再計算されません。

　　　　　　　　全体的に拡大または縮小されます。

　　　　　　　　そのため、大きな余白が表示される場合やテキストが画面の外にはみ出して、

　　　　　　　　スクロールバーが表示される場合があります。

＜対応案＞

　・画面レイアウトの見直しを行う

　・ズーム機能を無効にして、100%固定にする。

　　（ズーム機能は、グループポリシーなどで無効にすることが出来ます。）

### 強化されたタブ

Internet Explorer 6と比べて、Internet Exporer 9は、タブブラウズ機能が追加されています。

* IE 9のタブのの強化機能

Internet Explorer 9では、ポップアップを開く方法をInternet Explorerで自動的に判定するため、新規ウィンドウではなく、新規タブで表示される場合があります。

影響を受ける条件は以下になります。

＜条件＞

アンカータグ（<A>タグ）にtarget=”\_blank” や target=”任意の名前”属性を設定しているリンクを使用し、かつリンク先のセキュリティ ゾーンがリンク元と同じ場合

＜対応案＞

・アンカータグ（<A>タグ）のリンクをJavaScriptの“window.open”で新規ウィンドウを起動するように書き換える。

・ブラウザーの設定を変更する。

　[インターネット オプション] → [全般] → タブ項目の[設定] → [ポップアップの発生時] の中で

　[常に新しいウィンドウでポップアップを開く]を選択する。

＜タブの設定とセキュリティ ゾーンの組み合わせによる表示結果＞

|  |  |  |  |
| --- | --- | --- | --- |
| No. | ポップアップの発生時の設定 | リンク先のセキュリティ ゾーン | 表示結果 |
| 1 | IEで自動的に判定する（初期設定） | リンク元と同じゾーン | 新規タブ |
| 2 | リンク元と違うゾーン | 新規ウィンドウ |
| 3 | 常に新しいウィンドウで開く | リンク元と同じゾーン | 新規ウィンドウ |
| 4 | リンク元と違うゾーン | 新規ウィンドウ |
| 5 | 常に新しいタブで開く | リンク元と同じゾーン | 新規タブ |
| 6 | リンク元と違うゾーン | 新規ウィンドウ |

* タブブラウズの設定

Internet Explorer 9のタブブラウズの有効化／無効化など、タブブラウズ機能の設定を変更したい場合は、以下のメニューより行うことになります。

＜手順＞

|  |  |  |
| --- | --- | --- |
| 1 |  | メニュー[ツール]から、[インターネット オプション]を選択します。 |
| 2 |  | [インターネット オプション]が開きますので、次に[全般タブ]の[タブ]のところにある、ボタン[設定]をクリックします。 |
| 3 |  | [タブ ブラウズの設定]が開きます。  タブブラウズを無効にしたい場合は、[タブブラウズを有効にする]のチェックボックスを外します。 |

### HTMLの機能強化（レンダリングエンジンの変更）

Internet Explorer 9では、Direct2D、DirectWriteによる描画、Natural Metricsの採用により、高速でかつ正確な画面描画が可能になりました。しかし、このことにより表示の一部が変更され、既存のInternet Explorerと違いが生じることになりました。

ただし、ドキュメントモードを変更することにより、互換性を保てるようになっています。

* リスト表示（間隔、リストのマーク）

リスト表示で行間やリストのマークのサイズが変わります。行間が変わるため表示位置が変わる可能性があります。

＜条件＞

　以下のタグをドキュメントモードが“IE9 Standards”で使用する場合

　・<ul>タグ：Unordered List（順序のあるリスト）

　・<ol>タグ：Ordered List（順序のあるリスト）

　・<li>タグ：List Item

＜対応案＞

　・ドキュメントモードをQuirks モードにする。

|  |  |  |
| --- | --- | --- |
| No | IE6 | IE9 |
| 1 |  | マーカーのイメージが異なります。  行間が異なります。 |

* 複数行表示（行間）

複数行にわたる文章で、行間のサイズが変わります。そのため表示位置が変わる可能性があります。

＜条件＞

　ドキュメント モードが“IE9 Standards”で、複数行にわたる文章がある場合

＜対応案＞

　・ドキュメントモードをQuirks モードにする。

* テーブル表示（テーブルサイズ、文字間隔、枠、<body text = “color”>）

Internet Explorer 9では、テーブル表示について以下の項目で変更があります。

・テーブルのサイズ、文字間隔、マージン（空白）が変更されます。

・テーブルの枠線、罫線の見ためが変更されます。

・テーブルの枠線、罫線の色が<body text = “color”>の指定で変更されます。

＜対応案＞

　・ドキュメントモードをQuirks モードにする。

|  |  |  |
| --- | --- | --- |
| No | IE6 | IE9 |
| 1 |  |  |

* スタイルシート（{white-space:pre;}）

Internet Explorer 9では、スタイルシートの{white-space:pre;}で、表示が変わります。

これは、Internet Explorer 6では、スタイルシート{white-space:pre;}のpreが無効になるためにInternet Explorer 9とは表示に違いがあります。

特に、表示には関係なくソースコードを見やすくするために入れていた「改行」などが表示上も改行されるようになるため、表示が変わります。

＜条件＞

　スタイルシートで{white-space:pre;}を使っていて、ソースの中に半角スペース、タブ、改行がある場合

＜対応案＞

　・ドキュメントモードをQuirks モードにする。

　・スタイルシート、ソースコードを修正する。

|  |  |
| --- | --- |
| スタイルシート | |
| p.sample1 {width:40px; background-color:#33ccff; white-space: normal;}  p.sample2 {width:40px; background-color:#33ccff; white-space: pre;}  p.sample3 {width:40px; background-color:#33ccff; white-space: nowrap;} | |
| ソースコード | |
| <html>  　<head>  　　<title>スタイルシート white-space</title>  <link rel="stylesheet" href="sample.css" type="text/css">  </head>  <body>  <p class="sample1">Internet Explorer 9</p>  <p class="sample2">Internet Explorer 9</p>  <p class="sample3">Internet Explorer 9</p>  </body>  </html> | |
| IE6の結果 | IE 9の結果 |
|  | バックグラウンドの塗りつぶし領域が異なります。 |

|  |  |
| --- | --- |
| スタイルシート | |
| p.sample1 {width:40px; background-color:#33ccff; white-space: pre;}  スタイルシートは同じです。  p.sample2 {width:40px; background-color:#33ccff; white-space: pre;}  p.sample3 {width:40px; background-color:#33ccff; white-space: pre;} | |
| ソースコード | |
| <html>  　<head>  　　<title>スタイルシート white-space</title>  <link rel="stylesheet" href="sample.css" type="text/css">  </head>  改行があります。  <body>  <p class="sample1">Internet Explorer 9</p>  <p class="sample2">  Internet Explorer 9</p>  改行があります。  <p class="sample3">Internet Explorer 9  </p>  </body>  </html> | |
| IE6の結果 | IE 9の結果 |
| 改行がありません。 | それぞれ、改行があります。 |

* <input>タグ

Internet Explorer 9では、<input>タグで表示するボタン、テキストボックスなどのサイズが変わります。

＜条件＞

　<input>タグを使用している。

　テキストボックスのサイズ（<input type = “text” ~>）

＜対応案＞

　・ドキュメントモードをQuirksモードにする。

　・スタイルシートで、ボタンのサイズを指定する。

* テキストエリア（高さ、幅）

Internet Explorer 9では、<textarea>タグで表示するテキストエリアのサイズが変わります。

＜条件＞

　<textarea>タグを使用している。

＜対応案＞

　・ドキュメントモードをQuirksモードにする。

* フォントサイズ（サイズ）

Internet Explorer 9では、フォントのサイズが若干変わります。

また、boldの指定（<b>タグ、font-waight=boldなど）の表示についても同様に若干変わります。

＜対応案＞

　・ドキュメントモードをQuirksモードにする。

＜補足＞

　・スタイルシートで、詳細に設定を調整すればある程度までは同じような表示に近づけることは

　　可能です。

　　ただし、フォントのサイズを絶対単位（px、mmなど）で指定するとInternet Explorerで

　　文字のサイズを変更しても、表示が変わらないので気を付ける必要があります。

図　フォントサイズの比較

|  |  |
| --- | --- |
| IE6 |  |
| IE9 |  |

* セレクトボックス（スクロールバー）

Internet Explorer 9では、セレクトボックス（<select>タグ）で、縦スクロールバーが表示される場合があります。また、枠線やコントロールのボタンなどのデザインの見ためが変更されます。

＜条件＞

<select>タグで表示する選択肢の数（行数）（<select size=”X”>のX）と選択肢の数が等しい場合

※ただし、スタイルシートでセレクトボックスの幅（width）、高さ（height）を指定していた場合、

　指定していない場合と動作が異なります。

＜対応案＞

　・ドキュメントモードをQuirksモードにする。

|  |  |
| --- | --- |
| ソースコード | |
| <form>  <div>  選択肢の数と同じです。  <select name="BROWSER" size=8 >  <option value="IE9">Internet Explorer 9.X  <option value="IE8">Internet Explorer 8.X  <option value="IE7">Internet Explorer 7.X  <option value="IE6">Internet Explorer 6.X  <option value="IE5">Internet Explorer 5.X  <option value="IE4">Internet Explorer 4.X1234567890  <option value="IE3" selected>Internet Explorer 3.X  <option value="OTHER">その他  </select>  </div>  </form> | |
| IE6の結果 | IE 9の結果 |
| 縦スクロールバーはありません。 | 縦スクロールバーが表示されます。 |

|  |  |
| --- | --- |
| ソースコード | |
| <form>  <div>  選択肢の数より大きい数です。  <select name="BROWSER" size=9 >  <option value="IE9">Internet Explorer 9.X  <option value="IE8">Internet Explorer 8.X  <option value="IE7">Internet Explorer 7.X  <option value="IE6">Internet Explorer 6.X  <option value="IE5">Internet Explorer 5.X  <option value="IE4">Internet Explorer 4.X1234567890  <option value="IE3" selected>Internet Explorer 3.X  <option value="OTHER">その他  </select>  </div>  </form> | |
| IE6の結果 | IE 9の結果 |
|  | 縦スクロールバーはありません。 |

* テキストと下線の間隔

Internet Explorer 9では、下線、打ち消し線などの位置や太さが若干変更されます。

＜条件＞

　<u>タグやtext-decorationなどを使い、下線や打ち消し線などを表示している場合

＜対応案＞

　・ドキュメントモードをQuirksモードにする。

### CSSサポートの強化

CSSの標準準拠が強化されたため、Internet Explorer以外のブラウザー向けCSSハックがInternet Explorer 9でも適用されるようになりました。Internet Explorer 6以外のブラウザーにも対応しているアプリケーションでは、ドキュメントモードの変更や条件付きコメントなどを利用する必要があります。

また、CSS3のサポート、CSS2.1への準拠のため、特にいくつかの要素、タグを使用している場合に画面表示を確認する必要があります。

* CSSハック

Internet Explorer 9 では、CSSの標準準拠が強化されたため、以下のCSSハックを使われている場合にはドキュメントモードの変更などが必要になります。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | CSSハック名 | 例 | IE6 | IE9 |
| 1 | ルートセレクタハック | :root p{ color: #f00;} | × | ○ |
| 2 | ファーストオブタイプハック | body:first-of-type p{ color: #f00;} | × | ○ |
| 3 | 否定Lang疑似クラスハック | html:not([lang\*=""]) p{ color:#f00;} | × | ○ |
| 4 | ノットターゲットハック | html:not(:target) p{ color:#f00;} | × | ○ |
| 5 | 構造的な疑似クラスハック | body:nth-of-type(1) p{ color:#f00;} | × | ○ |
| 6 | 否定Nth Child疑似クラスハック | html:not(:nth-child(n)) p{ color:#f00;} | × | ○ |
| 7 | モズエニーリンクハック | p, x:-moz-any-link, x:only-child{color:#f00;} | ○ | × |
| 8 | スターハック | \* html p{ color: #f00;} | ○ | × |
| 9 | ２スターハック | \* html\*p{ color:#f00;} | ○ | × |
| 10 | インポータントハック | p{  /\* 通常 \*/  color:#000 !important;  /\* IE 6以下 \*/  color:#f00;  } | ○ | × |
| 11 | アンダスコアハック | p { \_color: #f00;} | ○ | × |
| 12 | チャイルドセレクタハック | html > body p { color: #f00;} | × | ○ |
| 13 | 隣接セレクタハック | head + body p { color: #f00;} | × | ○ |
| 14 | オーウェンハック | head:first-child + body p { color: #f00;} | × | ○ |
| 15 | 属性セレクタハック | html[xmlns] p { color: #f00;} | × | ○ |
| 16 | スター７ハック | html\*p { color: #f00;} | ○ | × |
| 17 | ハッシュハック | p { #color: #f00;} | ○ | × |
| 18 | アスタリスクハック | p { \*color: #f00;} | ○ | × |
| 19 | 全角スペースハック | p{  /\* 通常 \*/  color:#000;  /\* IE 6/7 \*/  　color:#f00;  } | ○ | × |

＜対応案＞

　・ドキュメントモードをQuirksモードにする。

　・条件付きコメントを利用して、Internet Explorer 6の場合と同様の処理になるようにする。

　・CSSを書き換える。

* float要素でのmargin

Internet Explorer 9 では、CSSの標準準拠が強化されたため、float要素の表現が変わります。

＜対応案＞

　・ドキュメントモードをQuirksモードにする。

　・CSSを書き換える。

|  |  |
| --- | --- |
| 変更点 | |
| ボックスの高さの算出からフロート要素が除外されるようになりました。 | |
| ソースコード | |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html;charset=Shift\_JIS" />  <title>SAMPLE</title>  <style type="text/css">  #container{ background-color:#33ccff; width:360px; }  #left{ float: left; width: 160px; }  floatにしています。  #right{ float: right; width: 160px; }  </style>  </head>  <body>  <div id="container">  <p id="left">  左Internet Explorer 6  左Internet Explorer 6  左Internet Explorer 6  </p>  <p id="right">  右Internet Explorer 9  右Internet Explorer 9  右Internet Explorer 9  </p>  </div>  </body>  </html> | |
| IE6の結果 | IE 9の結果 |
| 親要素の高さにフロートが含まれているために背景が塗りつぶされています。 | 親要素の高さにフロートが含まれてないために背景が塗りつぶされません。 |

|  |  |
| --- | --- |
| 変更点 | |
| フロート要素の子要素の上下のデフォルトスタイルシートによるマージンが消えないようになりました。 | |
| ソースコード | |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html;charset=Shift\_JIS" />  <title>SAMPLE</title>  <style type="text/css">  h1{ background-color: orange; margin:0; }  #footer{ clear:both; font-size: 2em; background-color: yellow; margin:0; }  #container{ width:360px; }  #left{ float: left; width: 160px; background-color: limegreen; }  #right{ float: right; width: 160px; background-color: skyblue; }  </style>  </head>  float要素には、マージンを指定していないため、デフォルトスタイルシートの設定が適用されます。にしています。  <body>  <div id="container">  <h1>へッダー</h1>  <p id="left">  左Internet Explorer 6  左Internet Explorer 6  左Internet Explorer 6  </p>  <p id="right">  右Internet Explorer 9  右Internet Explorer 9  右Internet Explorer 9  </p>  <div id="footer">フッター</div>  </div>  </body>  </html> | |
| IE6の結果 | IE 9の結果 |
| マージンがありません。 |  |
| 変更点 | |
| 複数の要素にfloatやclearを指定した時の並び順が変わりました。 | |
| ソースコード | |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />  <title>SAMPLE</title>  <style type="text/css">  div{ width: 100px; font-size: 4em; text-align: center; }  float要素が複数あります。  #a{ float:left; background-color:pink }  #b{ float:left; background-color:red; clear:left }  #c{ float:left; background-color:orange }  #d{ float:left; background-color:yellow; clear:left }  #e{ float:left; background-color:skyblue }  </style>  </head>  <body>  <div id="a">A</div>  <div id="b">B</div>  <div id="c">C</div>  <div id="d">D</div>  <div id="e">E</div>  </body>  </html> | |
| IE6の結果 | IE 9の結果 |
|  |  |

CとEの位置が異なります。

* テーブル要素の優先度の変更

Internet Explorer 9では、CSS2.1への準拠のためにテーブル要素の優先度が変わっています。

（Internet Explorer 8から準拠）

＜対応案＞

　・ドキュメントモードをQuirksモードにする。

　・CSSを書き換える。

＜テーブル要素の線種の優先度＞

テーブル要素に border-collapse: collapse; を設定し、枠線の重複が発生した場合、Internet Explorer 6では、 border-collapse: collapse;に部分的にしか対応していないため、Internet Explorer 9と描画の結果に差異が生じる場合があります。

CSS 2勧告では隣接するセルの異なる枠線が重なった場合、下記の優先順位に基づいて描画するように規定されています。

1. border-style: hidden が設定されている要素が最優先となる (逆に none は最も優先度が低い)
2. border-style: hidden が設定されていない場合、線の太さが太いものが優先される
3. 線の太さが同じ場合、線の形式が double のものが優先される  
   (double、solid、dashed、dotted、ridge、outset、groove、inset の順に優先されます)
4. 線の形式が同じ場合、th もしくは td 要素に対する設定値が優先される  
   (thと td要素、tr要素、thead とtbody と tfoot 、col要素、colgroup要素、table要素 の順に優先されます)
5. border-style: none が設定されている要素の優先度が最も低い

１．～５．の条件が全く同じで色が違うというような場合は、左もしくは上のセルの線種が優先されます。

|  |  |
| --- | --- |
| IE6 | IE9 |
|  |  |

上図は、以下のような基準で描画されています。

|  |  |  |
| --- | --- | --- |
|  | IE6 | IE9 |
| dotted と dashed の接線 | dotted が左にあるので dotted を描画 | dotted と dashed では dashed に優先権があるので dashed を描画 |
| dotted と solid の接線 | dotted のセルが上にあるので、dotted を描画 | solid と dotted では solid に優先権があるので solid を描画 |
| solid と double の接線 | solid のセルが左側にあるので solid を描画 | solid と double では double に優先権があるので double を描画 |
| double と dashed の接線 | dashed のセルが上にあるので、dashed を描画 | double と dashed では double に優先権があるので、double を描画 |

線種の優先度は、CSS2にて下記のように規定されています。

1. double（二重線）
2. solid（実線）
3. dashed（破線）
4. dotted（点線）
5. ridge（凸線）
6. outset（上面）
7. groove（凹線）
8. inset（底面）

|  |  |
| --- | --- |
| IE6 | IE9 |
|  |  |

（比較のため、優先度の高い線種を下にしています。）

＜テーブル要素の優先度＞

Internet Explorer 9ではCSS 2.1勧告に基づき、テーブル要素ごとに優先度を設定しており、border-style:hiddenの有無や線の太さ、線の種類によって優先度が決定できない場合（色指定のみが異なる場合など）の優先度の決定に利用されます。

テーブル要素の優先度は、以下のように規定されています。

1. td要素（th要素）
2. tr要素
3. thead要素、tbody要素、tfoot要素
4. col要素
5. colgroup要素
6. table要素

仮に下記のようなCSSをテーブルに適用した場合、Internet Explorer 9とInternet Explorer 6では適用されるものが異なります。

|  |  |  |
| --- | --- | --- |
|  | IE6 | IE9 |
| table{ border: red; } | 優先順位 1 - border: red; を適用 | 優先順位 6 |
| colgroup {border: green;} | 適用されず | 優先順位 5 |
| col {border: yellow;} | 適用されず | 優先順位 4 |
| tbody {border: black;} | 適用されず | 優先順位 3 |
| tr {border: white;} | 適用されず | 優先順位 2 |
| td {border: blue;} | 優先順位 2 | 優先順位 1 - border: blue; を適用 |

* before疑似要素、after疑似要素

Internet Explorer 9 では、CSS2.1への準拠のためにbefore疑似要素、after疑似要素を処理できるようになりました。（Internet Explorer 8から準拠）

＜対応案＞

　・ドキュメントモードをQuirksモードにする。

　・CSSを書き換える。

|  |  |
| --- | --- |
| スタイルシート | |
| p.sample1:before {content: "●Before●" }  p.sample2:after {content: url( "sample.png") } | |
| ソースコード | |
| <html>  <head>  <link rel="stylesheet" href="test\_content.css" type="text/css">  </head>  <body>  <p class="sample1">先頭にテキストを挿入しました。</p>  <p class="sample2">末尾に画像を挿入しました。</p>  </body>  </html> | |
| IE6の結果 | IE 9の結果 |
| ソースコードにあるテキストしか表示されません。 | スタイルシートで指定した、テキストと画像が表示されています。 |

### Windowの操作

Internet Explorer のユーザーインターフェイスの変更や、JavaScriptの動作が一部変更になったため、Windowの操作を行っている処理について確認する必要があります。

* ウィンドウサイズ（横方向の最小サイズ、サイズの考え方）

Internet Explorer 9 では、ウィンドウサイズの最小値やサイズの考え方が変わりました。

Internet Explorer 6では、外枠を含むところでサイズを指定するのに対して、Internet Explorer 9では、表示領域のサイズを指定するようになります。横方向の指定可能な最低サイズもInternet Explorer 9では、「250pixel」となっております。

そのため、JavaScriptなどでサイズを指定してウィンドウを表示している場合、動作が異なります。

※Internet Explorer 6でもアドレスバー、ステータスバーを表示させない場合は、外枠を含みません。

＜対応案＞

　・レイアウトを変更する。

　・指定するウィンドウのサイズを変更する

|  |  |
| --- | --- |
| モーダルウィンドウ | |
| IE6 | IE9 |
|  | dialogHeight  （外枠も含む） |
| 新ウィンドウ（サイズ 200×200 を指定） | |
| IE6 | IE9 |
|  | dialogHight  （外枠は含まない）  dialogWidth  （外枠は含まない） |

* アドレスバーを非表示にしているしているアプリケーション

Internet Explorer 9 では、セキュリティ対策のため基本的にアドレスバーが表示されます。

そのため、アドレスバーを画面レイアウトが変わる場合があります。

ただし、セキュリティの設定を変更することにより、スクリプトなどからアドレスバーを非表示にして開くことができます。規定では、ローカルイントラネットゾーン、信頼済みサイトはアドレスバーがないウィンドウを開くことが出来ます。

しかし、セキュリティ対策を考えるとアドレスバーが表示されることを前提として画面レイアウトを変更することをお奨めします。

＜対応案＞

　・レイアウトを変更する。

|  |  |
| --- | --- |
| アドレスバー表示 | アドレスバー非表示 |
|  |  |

* ステータスバーへのメッセージの表示

Internet Explorer 9では、ステータスバーは初期設定で非表示になっています。

そのため、ナビゲーションやメッセージなどをステータスバーに表示しているアプリケーションでは、ウィンドウを開くときにステータスバーを表示させる必要があります。

またステータスバーが非表示にした時に、今までステータスバーに表示されていたアンカー（<A>）タグの遷移情報（href属性）がWindowの左下にツールチップで表示されます。

このため、ステータスバーを非表示にしてユーザーから隠していた遷移情報が、表示されるようになります。これについては、対処方法はありません。

＜対応策＞

　・ステータスバーにメッセージを表示するアプリケーションは、ステータスバーを表示させる

スクリプト

<a href="http://www.microsoftstore.jp/Form/Product/ProductCategory.aspx?cat=100" target="\_top" title="Windows">

図　IE6のステータスバー

図　IE9のステータスバー



図　IE9のツールチップ（ステータスバー 非表示）



* スクリプトでのウィンドウのクローズ

Internet Explorer 9では、スクリプトでウィンドウをcloseする場合に、Internet Explorer の確認ダイアログが表示される場合があります。

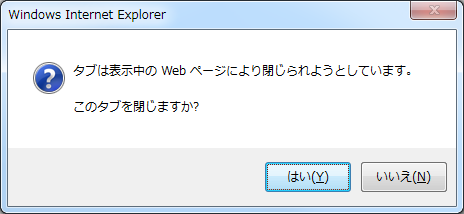
一般的に、スクリプトから開かれたWindowは確認ダイアログなしで、スクリプトでクローズできます。

メニューやショートカットから開かれたWindowはスクリプトでクローズする場合に確認ダイアログが表示されます。

＜対応案＞

　・ありません。

図　確認ダイアログ



## Java Script

近年、WebにおけるJavaScriptの重要性はますます高まってきています。

現在のWebアプリケーションでは、従来のデスクトップ アプリケーションと同等な機能や複雑さを実現するために、何千（あるいわ何十万）ものラインのJavaScriptコードが実行されています。またAJAXとスクリプト ライブラリの出現により、今日では、ほぼすべての新しいサイトはJavaScriptを利用しています。

このため、ブラウザーとそのScript Engineは高速に処理を行うことが必要になっています。

今回、コード名“Chakra”という新しいJavaScript EngineをInternet Explorer 9に実装し、リリースしています。

Internet Explorer 9は、従来のInternet Exploereのバージョンと同等の処理結果になるようにしています。

しかし、JavaScriptの仕様をECMAScript 5に標準準拠したために一部の処理において、処理結果に差が生じる場合があります。

ただし、互換性が取れるようにしております。

### ECMA Script 5 標準準拠

* getYearメソッド

Internet Explorer 9では、DateクラスのgetYearメソッドの戻り値が変更されました。

Internet Explorer 8までは、西暦を返しますが、Internet Explorer 9では、1900年からの差分を返します。

＜対応策＞

　・ドキュメントモードをQuirksモードにする。

　・ブラウザーの判別を行って、Internet Explorer 9の場合は、戻り値に1900を加算する。

　・getFullYearメソッド使用するように変更する。

|  |  |  |  |
| --- | --- | --- | --- |
| No | スクリプト | IE6 | IE9 |
| 1 | var dt = new Date();  alert( dt.getYear());  ※実行時が、2012年1月1日の場合 | 2012 | 112 |
| 2 | var dt = new Date();  alert( dt.getFullYear());  ※実行時が、2012年1月1日の場合 | 2012 | 2012 |

* toFixedメソッド

Internet Explorer 9では、toFixedメソッドの小数点以下の処理が変更されました。四捨五入した値を返します。

Internet Explorer 6では、以下のルールになります。

・指定された桁数で切り捨てを行った時に結果が「0（ゼロ）」になる場合、切り捨てた結果を文字列

にして返します。

　　・指定された桁数で切り捨てを行った時に結果が「0（ゼロ）」にならない場合、四捨五入を行った

結果を文字列にして返します。

＜対応策＞

　・IE6の上記のルールの通りにしたい場合、上記ルールになる処理に置き換えます。

　・IE6の場合に上記ルールではなく、常に四捨五入されるように処理を行っていた場合、

特に変更の必要はありません。

|  |  |  |  |
| --- | --- | --- | --- |
| No | スクリプト | IE6 | IE9 |
| 1 | alert( (0.5).toFixed(0));  alert( (-0.5).toFixed(0)); | 0  -0 | 1  -1 |
| 2 | alert( (0.55).toFixed(1));  alert( (0.54).toFixed(1));  alert( (-0.55).toFixed(1)); | 0.6  0.5  -0.6 | 0.6  0.5  -0.6 |
| 3 | alert( (0.05).toFixed(1));  alert( (0.04).toFixed(1));  alert( (-0.05).toFixed(1)); | 0.0  0.0  -0.0 | 0.1  0.0  -0.1 |

* 配列末尾の空要素のlengthプロパティ

Internet Explorer 9では、配列の末尾に空の要素がある場合、lengthプロパティの値がInternet Explorer 6から変更されています。

Internet Explorer 9では、配列の末尾にある空要素だけは存在しない要素として扱われ、lengthのプロパティの値にカウントされません。

＜対応策＞

・ドキュメントモードをQuirksモードにする。

・配列末尾に空要素がある場合は、明示的に配列末尾に要素を指定する。

|  |  |  |  |
| --- | --- | --- | --- |
| No | スクリプト | IE6 | IE9 |
| 1 | var aryobj = [1,2,3,4,5,,,];  alert( aryobj.length ); | 8 | 7 |
| 2 | var aryobj = new Array(8);  alert( aryobj.length ); | 8 | 8 |
| 3 | var aryobj = [1,2,3,4,5,,,];  aryobj[9] = 0;  alert( aryobj.length ); | 10 | 10 |

* 変数の宣言と初期化

Internet Explorer 9では、変数を変数を宣言した場合の初期値とそうでない場合の初期値がInternet Explorer 6から変更されています。

＜対応策＞

・ドキュメントモードをQuirksモードにする。

・明確に変数の宣言と初期化を行うように修正する。

|  |  |  |  |
| --- | --- | --- | --- |
| No | スクリプト | IE6 の結果 | IE9 |
| 1 | <script type="text/javascript">  try {  result;  } catch (e) {  if (e instanceof ReferenceError) {  alert("result: undefined variable");  　}  }  </script> | エラーなし | “result: undefined variable”  エラーが発生して例外処理が走ります。 |
| 2 | <script type="text/javascript">  try {  result;  } catch (e) {  if (e instanceof ReferenceError) {  alert("result: undefined variable");  　}  }  var resut;  </script> | エラーなし | エラーなし |
| 3 | <script type="text/javascript">  alert( result );  </script> | エラー | エラー |
| 4 | <script type="text/javascript">  var result;  alert( result );  </script> | エラー | “undefined”エラーにはなりません。 |
| 5 | <script type="text/javascript">  var result;  result = “”;  alert( result ); | “” | “” |

* 継承されたプロパティのdontenum属性

Internet Explorer 9では、オブジェクト「Object」に関してはDontEnum属性が付与されなくなりました。

具体的には、ObjectクラスのメンバであるvalueOfプロパティとtoStringプロパティは、for inループで列挙されるようになりました。

＜対応策＞

・ドキュメントモードをQuirksモードにする。

・スクリプトを修正する。

|  |  |  |  |
| --- | --- | --- | --- |
| No | スクリプト | IE6 の結果 | IE9 |
| 1 | var obj = { valueOf:0, toString:1, foo:2 };  var s = “”;  for( var p in obj ) s+=p+”,”;  alert( s ); | “foo,” | “valueOf,toString,foo,” |
| 2 | var obj = new Object();  var s = “”;  for( var p in obj ) s+=p+”,”;  alert( s ); | “”  データを設定していないので、属性はない。 | “”  データを設定していないので、属性はない。 |
| 3 | var obj = new Object();  obj.valueOf = 0;  obj.toString = 1;  obj.foo = 2;  var s = “”;  for( var p in obj ) s+=p+”,”;  alert( s ); | “foo,” | “valueOf,toString,foo,” |
| 4 | var ary = new Array();  var s = “”;  for( var p in ary ) s+=p+”,”;  alert( s );  alert( ary.length ); | “”  “0”  データを設定していないので、属性はない。  lengthもゼロ。  length属性は列挙されない。 | “”  “0”  データを設定していないので、属性はない。lengthもゼロ。  length属性は列挙されない。 |
| 5 | var ary = new Array();  ary[0] = 1;  ary[1] = 2;  ary[2] = 3;  ary.prop1 = “配列”;  var s = “”;  for( var p in ary ) s+=p+”,”;  alert( s );  alert( ary.length ); | “0,1,2,prop1,”  “3”  データのあるところまで、列挙される。  length属性は列挙されない。 | “0,1,2, prop1,”  “3”  データのあるところまで、列挙される。  length属性は列挙されない。 |

* コードブロックの後のセミコロン

Internet Explorer 9では、セミコロン（;）の処理がより厳密になりました。

そのため、構文の途中でセミコロン（;）があるとエラーとなります。

推奨されていませんが、セミコロン（;）の省略は可能です。

＜対応策＞

・ドキュメントモードをQuirksモードにする。

・セミコロン（;）の位置を見直して、スクリプトを修正する。

|  |  |  |  |
| --- | --- | --- | --- |
| No | スクリプト | IE6 | IE9 |
| 1 | var s = “”;  …略  if( s != null ){  alert( “ s = “ + s );  } // セミコロンなし  else {  alert( “ s is null.” );  } // セミコロンなし | ○ | ○ |
| 2 | var s = “”;  …略  if( s != null ){  alert( “ s = “ + s );  } // セミコロンなし  else {  alert( “ s is null.” );  }; // セミコロンあり | ○ | ○ |
| 3 | var s = “”;  …略  if( s != null ){  alert( “ s = “ + s );  }; // セミコロンあり  else {  alert( “ s is null.” );  }; // セミコロンあり | ○ | エラー  else の行でエラーになります。 |
| 4 | var s = “”;  …略  if( s != null ){  alert( “ s = “ + s );  }; // セミコロンあり  else {  alert( “ s is null.” );  } // セミコロンなし | ○ | エラー  else の行でエラーになります。 |

* 変数に代入された関数

Internet Explorer 9では、関数を変数に代入した場合の関数の呼び出し方が変更になりました。

＜対応策＞

　・ドキュメントモードをQuirksモードにする。

　・変数に代入した関数を呼び出す場合、callやbindなどを用いて実行する。

|  |  |  |  |
| --- | --- | --- | --- |
| No | スクリプト | IE6 | IE9 |
| 1 | var f = document.write;  f( “Internet Explorer 9”); | ○ | エラーオブジェクトの呼び出しが無効です。 |
| 2 | var f = document.write;  f.call( document, “Internet Explorer 9”); | ○ | ○ |

### DOMイベントの変更 標準準拠

JavaScriptイベントがW3C仕様に準拠しました。そのため、いくつかのイベントハンドラの実装が変更されています。

* バブルアップ

Internet Explorer 9では、イベントのバブルアップが行われるようになりました。

イベント発生元の上位のタグに、依然と異なるイベントが発生する場合には、発生元のタグで上位のタグにイベントハンドラを渡さないようにしてください。

＜対応策＞

　・ドキュメントモードをQuirksモードにする。

　・上位のタグにイベントハンドラを渡したくない場合、発生元のタグでそのイベントに対して、

“cancelBubble=true;”と追加する。

|  |  |
| --- | --- |
| IE6の基本動作 | IE9の基本動作 |
| width  200  height  200  最小値：250に変更されます。 | height  200  width 250  イベントが発生しても伝播しない  <input>  <form> |

以下のような場合、selectBoxで項目を変更した場合、IE6ではdoChange()は実行されませんが、IE9では、doChange()が実行されます。

<script type="text/javascript">

function doChange() {

…略

}

</script>

<form>

<P onChange="doChange()">

<select>

<option>項目を選択してください</option>

<option>りんご</option>

<option>いちご</option>

<option>すいか</option>

</select>

</P>

</form>

以下は、例外です。どの「click!」の上でマウスをクリックしてもdoClicked()が実行されます。

（IE６、9ともに同じ動作になります。）

<script type="text/javascript">

function doClicked() {

…略

}

</script>

<form>

<P onclick=”doClicked()” >

click! <B>click!<.B> click!

</P>

</form>

バブルアップを止めたい場合は、“cancelBubble=true;”と記述します。

<form>

<P onclick=”doClicked()” >

click! <B onclick=“window.event.cancelBubble=true;”>click!<.B> click!

</P>

</form>

上記の場合、真ん中の「click!」の上でマウスをクリックしてもdoClicked()は実行されません。

また、functionの中で書けば、バブルアップはそこで止まります。

<script type="text/javascript">

function doClicked() {

…略

event.cancelBubble = true;

}

</script>

### DOM L2/3対応標準準拠

* プロパティ、属性の管理領域の分離

これまでのInternet Explorerのアーキテクチャでは、DOMオブジェクトは独自のプロパティ記憶域を持っていました。このプロパティ記憶域は、属性（HTML内の属性）の記憶域と同じ場所で管理されていました。Internet Explorer 9の新しいアーキテクチャでは、要素の属性の記憶域と要素のプロパティの記憶域が分離して管理されるようになりました。

そのため、プロパティと属性の取得方法が、Internet Explorer 6とInternet Explorer 9では異なります。

＜対応策＞

　・ドキュメントモードをQuirksモードにする。

　・プロパティ、属性を取得する方法を変更する。

＜例＞

<a id=”id” rel=”IE9”>sample</a>

上記のように定義した要素の属性の数を取得した結果を以下に記載します。

var element = document. getElementById("id");

alert( element.attributes.length );

|  |  |  |  |
| --- | --- | --- | --- |
| No | 取得方法 | IE6 | IE9 |
| 1 | element.attributes.length | 147 | 2 |

次に要素のの属性を取得した結果を以下に記載します。

var element = document. getElementById("id");

alert( element.id ); // Object.属性名

alert( element["id"] ); // Object[“属性名”];

alert( element.getAttribute("id") ); // Object.getAtteibute(“属性名”);

|  |  |  |  |
| --- | --- | --- | --- |
| No | 取得方法 | IE6 | IE9 |
| 1 | Object.属性名 | id | id |
| 2 | Object[“属性名”] | id | id |
| 3 | Object.getAtteibute(“属性名”) | Id | id |

ここで、プロパティを追加します。

var element = document. getElementById("id");

element.userProperty = "test"; // プロパティ名：userProperty

再度、属性の数を取得します。

var element = document. getElementById("id");

alert( element.attributes.length );

|  |  |  |  |
| --- | --- | --- | --- |
| No | 取得方法 | IE6 | IE9 |
| 1 | element.attributes.length | 148 | 2 |

IE6では、数が増えていますが、IE9では増えていません。

追加したプロパティの値を取得した結果を以下に記載します。

var element = document. getElementById("id");

alert( element.userProperty ); // Object.属性名

alert( element["userProperty "] ); // Object[“属性名”];

alert( element.getAttribute("userProperty ") ); // Object.getAtteibute(“属性名”);

|  |  |  |  |
| --- | --- | --- | --- |
| No | 取得方法 | IE6 | IE9 |
| 1 | Object.属性名 | test | test |
| 2 | Object[“属性名”] | test | test |
| 3 | Object.getAtteibute(“属性名”) | test | null |

* 空白文字の処理

DOM 内に空白文字が配置されている場合、従来のInternet Explorer とその他のブラウザーでは異なる処理が行われていました。この部分で他のブラウザーとの相互運用性が欠けていたことから Web開発者の負担となっていました。従来のバージョンの Internet Explorer では、ノードの先頭に空白文字があるとそれを無視し（削除または折りたたみ）、DOM 内のテキスト ノードにはその空白文字を配置しませんでした。Internet Explorer 9 では、すべての空白文字は維持され、DOM 内のテキスト ノードに配置されます。この動作は、その他の大半のブラウザーと同じになります。

＜対応策＞

　・ドキュメントモードをQuirksモードにする。

　・スクリプトなどを修正する。

＜例＞

|  |  |  |  |
| --- | --- | --- | --- |
| No. | マークアップ | IE 6の表現 | IE 9の結果 |
| 1 | <div><span>IE 9</span></div> | div  | ->span  |--->”IE 9” | div  | span  |--->”IE 9” |
| 2 | <div>  <span>IE 9</span>  </div> | div  | ->span  |--->”IE 9”  |->”\n“ | div  |->”\n “  | ->span  |--->”IE 9”  |->”\n“ |

※上記２のIE9の結果の２行目は、この場合は“改行”＋“半角スペース”＋“半角スペース”です。

上記のマークアップで以下の処理を行った場合の結果を記載します。

var nodes = document.getElementsByTagName("div");

for( i=0; i<nodes.length; i++ ) {

for( j=0; j<nodes[i].childNodes.length; j++ ) {

alert( nodes[i].childNodes[j].outerText );

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| No. | マークアップ | IE 6の結果 | IE 9の結果 |
| 1 | <div><span>IE 9</span></div> | IE 9 | IE 9 |
| 2 | <div>  <span>IE 9</span>  </div> | IE 9  undefined | undefined  IE 9  undefined |

IE 9で要素だけが必要な場合、Element Traversal APIを使用します。

（IE 6では、使用できません）

var nodes = document.getElementsByTagName("div");

for( i=0; i<nodes.length; i++ ) {

var childEl = nodes[i].firstElementChild;

while( childEl ){

alert( childEl.firstChild.nodeValue );

childEl = childEl.nextElementSibling;

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| No. | マークアップ | IE 6の結果 | IE 9の結果 |
| 1 | <div><span>IE 9</span></div> | 実行不可 | IE 9 |
| 2 | <div>  <span>IE 9</span>  </div> | 実行不可 | IE 9 |

* createElementメソッドの仕様変更

Internet Explorer 9では、createElementメソッド内で“<>”を利用してタグを指定することが出来なくなりました。

＜対応策＞

　・ドキュメントモードをQuirksモードにする。

　・スクリプトなどを修正する。

＜例＞

|  |  |  |  |
| --- | --- | --- | --- |
| No. | script | IE 6の結果 | IE 9の結果 |
| 1 | var element = document.createElement(“<div id=’id’>”); | ○ | DOM Exception: INVALID\_CHARACTER\_ERR |
| 2 | var element = document.createElement('div');  element.id = "id"; | ○ | ○ |

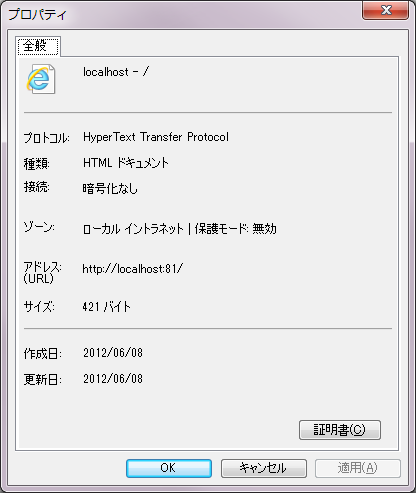
## セキュリティ

### セキュリティ ゾーン

「3.1.1. セキュリティ ゾーン」で説明したとおり、Internet Explorer 9 では Internet Explorer 6 と比較して規定のセキュリティレベルが強化されている。

Internet Explorer が Web サイトにアクセスした際に、どのように振る舞うかはセキュリティ ゾーンで決定される。Internet Explorer 9 においては組織内の Web アプリケーションは「ローカル イントラネット」としてアクセスするように構成しておくこと。

なお、Internet Explorer 9 でアクセスしているサイトのセキュリティ ゾーンを確認するには、＜ファイル＞－＜プロパティ＞ 操作を行ってページのプロパティ情報で確認できる。



* Internet Explorer 8 以前ではステータス バーで確認できたが、Internet Explorer 9 ではステータス バーからセキュリティ ゾーンの情報を確認できない。

### 保護モード

Internet Explorer の新しいセキュリティ機構である保護モードは、Windows 7 に搭載されているユーザー アカウント制御 (UAC) などの機能と連携することで、Internet Explorer を介した悪意のあるソフトウェア の進入を防ぐ機能です。

保護モードで動作する Internet Explorer は通常のプロセスよりも制限されたセキュリティ空間で実行され、Internet Explorer を介したコンピューターの侵害を防止します。

なお、保護モードはセキュリティ ゾーン毎に有効、無効を設定することができます。

既定では「インターネット」ゾーンと「制限付き」サイトにおいて、保護モードが有効になっています。

組織内の Web アプリケーションは「ローカル イントラネット」で動作するため、「ローカル イントラネット」ゾーンとして動作している限りにおいては、保護モードは無効なため、保護モードの影響を考慮する必要はありません。

### LCIE

Internet Explorer 8 ではブラウザの信頼性、パフォーマンス、およびスケーラビリティを改良する為の1つの方法としてLCIE と呼ばれる機能を導入しています。

LCIE とはタブ ブラウザの仕組みと深く関係しており、ウィンドウのアドレスバーやお気に入りバーが表示される部分を処理するプロセス (フレーム プロセス) とコンテンツが表示される部分を処理するプロセス (タブ プロセス) が分離され動作します。

これにより、タブ ウィンドウの1つがクラッシュしても全体に影響を与えることなく、当該 タブ ウィンドウ内だけに影響が限定されます。

しかし、LCIE の導入により Web アプリケーションの作り方や利用方法によっては、予期しない動作を行うことがあります。 (認証情報が共有されてしまうなど)

LCIE を制御する為には、下記のレジストリを追加する必要があります。

-----------------------------------------------------------------------------------

キー : HKLM (or HKCU) \Software\Microsoft\Internet Explorer\Main

名前 : TabProcGrowth

種類 : REG\_DWORD

値 : 0 、1

0 : LCIE を無効にし、iexplore.exe を1つに制限します。（注 : 本設定をすると、保護モードも無効となります）

1 : 保護モード無効／有効の状態で iexplore.exe （タブ プロセス）をそれぞれ 1 つずつに制限する。

-----------------------------------------------------------------------------------

### DEP/NX

データ実行防止 (DEP) または No-Execute (NX) と呼ばれるセキュリティ技術は、実行不可とマークされたコードをメモリ内で実行できないようにすることで、各種の攻撃を阻止します。

DEP/NX を ASLR (Address Space Layout Randomization) 機能などの他のテクノロジと組み合わせると、バッファー オーバーランのようなメモリ関連の特定の種類の脆弱性を悪用する攻撃者の侵入を防ぐことができます。この保護は、Internet Explorer 9 と読み込まれたアドオンの両方に適用されます。

## ActiveX コントロール

### ActiveX 仕様変更

Internet Explorer 6でMS06-013 QFEを適用した場合、ActiveXの仕様変更の影響を受けます。この仕様変更はInternet Explorer 9においても有効です。

ActiveX 仕様変更の内容は、<APPLET>, <EMBED>, <OBJECT> タグを利用してWebBrowser コントロールをページに配置した場合、コントロールを動作させるためには1クリックする必要があります。

参考資料：

Activating ActiveX Control

<http://msdn.microsoft.com/library/?url=/workshop/author/dhtml/overview/activating_activex.asp>

[MS06-013] Internet Explorer 用の累積的なセキュリティ更新プログラム

<http://support.microsoft.com/kb/912812>

### ActiveX Opt-in

Internet Explorer 9 では、ActiveX コントロールが無効化された状態でインストールされます。このため、ActiveX コントロールを従来の方法でデザインした場合、ユーザーの有効化のためのアクションがあるまで、インタラクティブなコントロールを受け付けません。

### ユーザーによるインストール

Windows 7 で Internet Explorer 8 を実行している場合、標準ユーザーは管理者権限を必要とすることなくActiveX コントロールをインストールできます。この機能向上によって、標準ユーザーにが日常のブラウジングで使用する ActiveX コントロールのインストールを可能にします。これにより、組織は、ユーザー アカウント制御の長所のすべてを容易に実現できます。

ユーザーごとの ActiveX は互換性を念頭に設計されているため、既存のActiveX コントロールのほとんどを書き換えることなくこの機能を使用できます。必要な変更は再パッケージだけです。Web ページがコントロールをインストールしようとすると、 Internet Explorer 6 と同様に、ユーザーに対して情報バーが表示されます。

### サイト構成 ActiveX

ユーザーが ActiveX コントロールを含む Web サイトに移動すると、Internet Explorer 8 によって、コントロールを実行できる場所の判断を含む多数のチェックが実行されます。このチェックは、サイト別 ActiveX と呼ばれ、悪意をもってコントロールの用途が変更されないようにするための保護メカニズムです。コントロールがインストールされていても特定の Web サイトで実行が許可されていない場合は、情報バーが表示され、現在の Web サイト上でコントロールの実行を許可するかどうかの判断がユーザーに求められます。

### ActiveX Installer Service

Windows 7ではActiveX コントロールのインストール時に管理者 (Administrators) 権限が必要になりました。従来はPowerUsers権限でもインストールできましたが、Windows 7ではPowerUsers権限が下位互換性維持の目的でのみ、提供されており基本的な権限は一般のユーザー (Users) 権限と何ら変わりません。Windows 7 では管理者権限を持たない場合には、ActiveX コントロールをインストールすることができなくなっています。

Users 権限しか保有していないユーザーに対して ActiveX コントロールのインストールを許可するために、ActiveX Installer Service が Windows 7 に実装されています。

ActiveX Installer Service はグループ ポリシーを利用して、Users 権限しか保有していないユーザーに対してインストール可能な Web サイトを定義します。

管理者が許可した Web サイトから ActiveX コントロールをインストールする場合、管理者権限が必要なくインストールすることが可能になります。

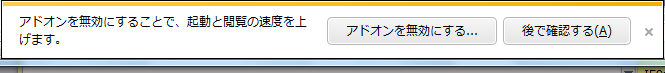
## その他

### アドオンがデフォルトで無効に設定される

Internet Explorer 9 にアドオンなどをインストールした場合、自動的に有効にはならず、有効化するにはユーザーが明示的に有効化する必要があります。

### アドオンのパフォーマンス アドバイザー

Internet Explorer 9 ではアドオンのパフォーマンスを測定しており、過去10回の起動時の平均値が閾値 (既定では 0.2 秒) を超えるとユーザーに通知バーで知らせます。



# まとめ

Internet Explorer 9 は最新の HTML 5 規格に対応し、またセキュリティやパフォーマンスが大幅に向上した最新のWeb Browser です。

Internet Explorer 9 を導入することでパフォーマンスの向上やセキュリティ向上、最新 HTML 5 技術を利用したリッチな Web アプリケーションのフロント エンドとして利用することが可能です。

しかし、本書のテーマであるように組織内で利用されている Internet Explorer 6 向けの Web アプリケーションを Internet Explorer 9 に対応させない限り、Internet Explorer 9 の導入は進みません。

本書では Internet Explorer 6 を利用している組織が互換性の影響を受けずに、Internet Explorer 9 に移行できる方法について紹介しました。

以下に要点をまとめますので、今後の参考にしてください。

1. インベントリ収集  
   Web アプリケーションのインベントリは自動的に作成することが困難です。  
   ヒアリングやユーザーへのアンケートなどを通じて、組織内に存在する Web アプリケーションを一覧化し、テスト計画やスケジュール策定の基礎資料として利用する必要があります。
2. Internet Explorer 9 事前構成  
   Internet Explorer 9 を利用して Web アプリケーションの検証を行う際には、必ずセキュリティ ゾーンや詳細設定、互換表示設定について事前検討を行ってください。  
   事前検討を行うことで、セキュリティや画面描画に関する互換性問題の発生を大幅に低減することができます。
3. 「ローカル イントラネット」ゾーンでの動作  
   組織内の Web サイトや Web アプリケーションは「ローカル イントラネット」として識別されるように構成してください。「ローカル イントラネット」として構成することで、セキュリティ強化に伴う影響や保護モードなどの影響を受けることがなくなります。
4. “イントラネットの互換表示設定” を有効化  
   Internet Explorer 9 に搭載されている互換表示設定を有効に使ってください。  
   “イントラネットの互換表示設定” を有効にすることで、「ローカル イントラネット」サイトにアクセスする際に Internet Explorer 6 と互換性の高い Internet Explorer 7 モードで描画を行います。  
   これにより、Internet Explorer 9 モードと比較して、大幅に互換性問題の発生を低減することが可能です。