



Sitecore CMS 6.2

Sitecore 動的リンク

Sitecore を使用して URL を構築するための開発者向けガイド

目次

Chapter 1	イントロダクション	3
Chapter 2	Sitecore 動的リンク	4
2.1	Sitecore 動的リンク管理	5
2.1.1	動的リンク設定	5
	Rendering.SiteResolving 設定	6
	LinkItemNotFound 設定	7
2.1.2	コンテンツ アイテムの URL へのアクセス方法	7
2.1.3	メディア アイテムの URL へのアクセス方法	8
2.2	検索エンジン最適化 (SEO) 対応の URL	9
2.3	IIS および ASP.NET URL	10
2.3.1	IIS7 またはそれ以降の統合モードの ASP.NET マネージ パイプライン	10
2.3.2	IIS HTTP 404 ページ	11
2.3.3	IIS ワイルドカード	13
	IIS 5.1 またはそれ以前 (Windows XP)	14
	IIS6 (Windows 2003)	14
	IIS7 またはそれ以降のクラシック マネージ パイプライン モード (Windows Vista、Windows 2008 および Windows 7)	15
2.3.4	URL を再作成する ISAPI フィルター	16

Chapter 1

イントロダクション

この文書では、Sitecore 動的リンク管理を設定する方法について説明します。Sitecore 管理者およびデベロッパーはこの情報を使用して、検索エンジン最適化 (SEO) と他のリンク管理機能を設定および実装することができます。

この文書には次の章があります：

- Chapter 1 – イントロダクション
- Chapter 2 – Sitecore 動的リンク

Chapter 2

Sitecore 動的リンク

この章では、ASP.NET を使用して URL を処理できるよう IIS を設定する方法（設定しない場合、URL は静的なファイルとして管理されます）、Sitecore で URL を動的に生成する方法、Sitecore で利用可能な検索エンジン最適化 (SEO) テクニックについて説明します。

この章には次のセクションがあります：

- Sitecore 動的リンク管理
- 検索エンジン最適化 (SEO) 対応の URL
- IIS および ASP.NET URL

2.1 Sitecore 動的リンク管理

Sitecore URL は、ファイル システム上のファイルではなく、Sitecore データベース内のアイテムに対応します。データ ポイント (アイテム) に対応する URL を使用することにより、ファイルに対応する URL を使用する場合よりも多くのメリットを活用できます。たとえば、データ方式の URL を使用すると、複数のデバイス間でのコンテンツの共有、複数の言語へのコンテンツの翻訳、プレゼンテーション コンポーネントの再利用、更新および変更が容易になります。動的 URL を使用すると、静的な URL ではクエリ文字列パラメーターを使用して渡す必要のあった情報を URL パスに組み込むことができます。たとえば、URL クエリ文字列パラメーター `sc_lang=en` の代わりにパス接頭辞 `/en` を使用してコンテンツ言語を指定することができます。

フィールド値にイメージやリンクを埋め込む場合など、多くの場合、アイテムは他のアイテムを参照します。アイテムはユーザーによって移動されたり名前が変更される場合があるため、Sitecore では、そのパスではなく一意の ID を使用してアイテムを参照します。プレゼンテーション コンポーネントは、`renderField` パイプラインを呼び出すか `FieldRenderer Web` コントロールを使用することにより、各フィールド値内の ID を対応するアイテムのわかりやすい URL に変換します。¹

2.1.1 動的リンク設定

`name sitecore` がある `web.config` 内の `/configuration/sitecore/providers/add` エレメントの次の属性を設定することにより、わかりやすい URL を制御することができます。

- **type:** `type` 属性を適切な .NET クラス シグネチャーに設定することにより、リンク プロバイダーを上書きすることができます。²
- **addAspxExtension:** URL に `.aspx` 拡張子を含めるかどうか (`true` または `false`)。 `addAspxExtension` を `false` に設定する場合、「IIS および ASP.NET URL」のセクションで説明されているように、ASP.NET を使用してすべての要求を処理するよう IIS を設定する必要があります。
- **alwaysIncludeServerUrl:** わかりやすい URL に HTTP プロトコルおよびドメイン (`http://localhost`) を含めるかどうか (`true` または `false`)。
- **encodeNames:** `web.config` 内の `/configuration/sitecore/encodeNameReplacements/replace` エレメントに応じてパスに名前をエンコードするかどうか (`true` または `false`)。
- **languageEmbedding:** URL に言語を含めるかどうか (`always`、`never` または `asNeeded`)。 `languageEmbedding` が `asNeeded` である場合、Sitecore では、受信した HTTP 要求からコンテキスト サイトを確認できない場合、言語を示す Cookie がこの HTTP 要求に含まれない場合、またはリンクされたアイテムの言語がコンテキスト言語とは異なる場合、URL に言語が含まれます。

¹ `RenderField Web` コントロールの詳細については、

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Component%20Reference.aspx> で『プレゼンテーション コンポーネント リファレンス』を参照してください。

² カスタム リンク プロバイダーの例については、<http://trac.sitecore.net/LinkProvider> を参照してください。

メモ

languageEmbedding が asNeeded である場合、Sitecore では、受信した HTTP 要求からコンテキスト サイトを確認できない場合、言語を示す Cookie がこの HTTP 要求に含まれない場合、またはリンクされたアイテムの言語がコンテキスト言語とは異なる場合、URL に言語が含まれます。

- **languageLocation:** URL パスの第 1 ステップとして言語を指定するか、sc_lang URL クエリ文字列パラメーターを使用するか (filePath または queryString)。
- **useDisplayName:** URL の構築時にアイテム表示名とアイテム名のどちらを使用するか (true または false)。useDisplayName 属性が true である場合、コンテンツ アイテムが同じでも言語が異なると URL が異なる可能性があります。

重要

Web アナリティクス ソリューションおよび検索エンジンを含む、既存の URL に依存する他のアプリケーションに対する設定の変更の影響を考慮してください。

メモ

Sitecore ユーザー インターフェイスでは、リッチ テキスト エディター (RTE) フィールドの RAW 値など、他の URL フォーマットが使用される場合があります。プレゼンテーション コンストラクトは、マークアップを Web クライアントに送信する前にこのような値をわかりやすい URL に変換します。

ヒント

リンク管理を設定する前に、「検索エンジン最適化 (SEO) 対応の URL」のセクションで説明されている検索エンジン最適化のテクニックおよび考慮事項を参照してください。

メモ

Rendering.SiteResolving 設定の影響の詳細については、「Rendering.SiteResolving 設定」のセクションを参照してください。

Rendering.SiteResolving 設定

name Rendering.SiteResolving がある web.config 内の /configuration/sitecore/settings/setting エレメントは、リンク マネージャーがマネージド Web サイト定義から URL に組み込むホスト名を決定するかどうかを制御します。

リンク マネージャーは、リンクされたアイテムごとにマネージド Web サイトを決定します。マネージド Web サイトは、リンクされたアイテムまたはその先祖の 1 つである開始アイテムを指定する最初の規定サイト定義です。

Sitecore の単一インスタンスが複数のマネージド Web サイトをサポートします。デフォルトでは、管理者は web.config 内の /configuration/sitecore/sites エレメントを使用してマネージド Web サイトを設定します。その他のプロパティのうち、各 /configuration/sitecore/sites/site エレメントは、そのマネージド Web サイトのホーム ページを表す開始アイテムを指定します。デフォルト設定では、website という名前のマネージド Web サイトの開始アイテムは /sitecore/content/home です。

```
<site name="website" virtualFolder="/" physicalFolder="/"
  rootPath="/sitecore/content" startItem="/home" ...
```

Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2010 Sitecore. All rights reserved.

Rendering.SiteResolving 設定が false であり、リンク マネージャーがリンクされたアイテムに関連付けられたマネージド Web サイトを決定できないか、リンクされたアイテムに関連付けられたこのマネージド Web サイトがコンテキスト サイトである場合、リンク マネージャーは現在の HTTP 要求内のホスト名を使用するか、リンク マネージャーの alwaysIncludeServerUrl 属性の値に応じてホスト名またはプロトコルを含めません。リンク マネージャーの alwaysIncludeServerUrl 属性の詳細については、「動的リンク設定」のセクションを参照してください。

Rendering.SiteResolving 設定が true であり、動的リンク マネージャーがリンクされたアイテムの論理 Web サイトを決定でき、このサイトがコンテキスト サイトではなく、このサイトの targetHostName 属性に値がある場合、リンク マネージャーは targetHostName 属性を使用します。targetHostName 属性に値がなく、hostName 属性に値があり、この値にアスタリスク文字 ("*") またはパイプ文字 ("|") が含まれない場合、リンク マネージャーは hostName 属性を使用します。

それ以外の場合、URL にホスト名またはプロトコルは含まれません。

注意

Sitecore.Links.LinkManager.GetItemUrl() メソッドは、Rendering.SiteResolving 設定を考慮しません。³

重要

属性はすべて大文字と小文字を区別します。hostName 属性には大文字の N が含まれます。

LinkItemNotFound 設定

name LinkItemNotFoundUrl がある web.config 内の /configuration/sitecore/settings/setting エレメントの value 属性は、リンクによって参照されるアイテムが存在しないときのリンク先のアイテムを制御します。

2.1.2 コンテンツ アイテムの URL へのアクセス方法

Sitecore.Links.LinkManager.GetItemUrl() メソッドを使用して、コンテンツ アイテムの URL にアクセスすることができます。⁴ たとえば、コンテキスト アイテムの URL にアクセスする方法は、次のとおりです：

```
Sitecore.Data.Items.Item item = Sitecore.Context.Item;
Sitecore.Links.UrlOptions urlOptions =
    (Sitecore.Links.UrlOptions) Sitecore.Links.UrlOptions.DefaultOptions.Clone();
urlOptions.SiteResolving = Sitecore.Configuration.Settings.Rendering.SiteResolving;
string url = Sitecore.Links.LinkManager.GetItemUrl(item, urlOptions);
```

³ Rendering.SiteResolving 設定を常に適用するカスタム リンク プロバイダーの例については、<http://trac.sitecore.net/LinkProvider> を参照してください。

⁴ Sitecore.Links.LinkManager.GetItemUrl() メソッドを使用してコンテンツ アイテムの URL にアクセスする例については、<http://trac.sitecore.net/Library/> で説明されている Sitecore.Sharedsource.Data.Items.Item.GetUrlExtension クラスを参照してください。

2.1.3 メディア アイテムの URL へのアクセス方法

`Sitecore.Resources.Media.MediaManager.GetMediaUrl()` メソッドを使用して、メディア アイテムの URL にアクセスすることができます。たとえば、Master データベース内のメディア アイテム `/Sitecore/Media Library/Images/Sample` の URL にアクセスする方法は、次のとおりです：

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item sampleItem = master.GetItem(
    "/sitecore/media library/images/sample");
Sitecore.Data.Items.Item sampleMedia = new Sitecore.Data.Items.MediaItem(sampleItem);
string url = Sitecore.StringUtil.EnsurePrefix(
    '/',
    Sitecore.Resources.Media.MediaManager.GetMediaUrl(sampleMedia));
```

注意

Sitecore では、メディア URL に先頭のスラッシュ文字 ("/") は自動的に含まれません。これにより、メディア アイテムの URL は相対 URL となり、IIS はチルダ文字 ("~") のためにこれをドキュメント ルートとします。インフォメーション アーキテクチャが非常に深いソリューションの場合、相対メディア URL が、Web クライアントまたは Web サーバーによって課される制限を超える可能性があります。この場合、前の例に示される `Sitecore.StringUtil.EnsurePrefix()` メソッドを使用して、メディア URL に先頭のスラッシュ文字が含まれるようにしてください。⁵

メモ

メディア URL にはプロバイダーがありません。

`Sitecore.Resources.Media.MediaUrlOptions` クラスを使用して、メディア アイテムの URL 読み取り時にメディア オプションを指定することができます。たとえば、Master データベース内の `/Sitecore/Media Library/Images/Sample` メディア アイテムのサムネールの URL を読み取る方法は、次のとおりです：

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item sampleItem = master.GetItem(
    "/sitecore/media library/images/sample");
Sitecore.Data.Items.MediaItem sampleMedia =
    new Sitecore.Data.Items.MediaItem(sampleItem);
Sitecore.Resources.Media.MediaUrlOptions mediaOptions =
    new Sitecore.Resources.Media.MediaUrlOptions();
mediaOptions.Thumbnail = true;
string url = Sitecore.StringUtil.EnsurePrefix('/',
    Sitecore.Resources.Media.MediaManager.GetMediaUrl(sampleMedia, mediaOptions));
```

⁵ メディア URL 内の接頭辞を一貫して変換するソリューションについては、<http://trac.sitecore.net/LinkProvider/> を参照してください。

2.2 検索エンジン最適化 (SEO) 対応の URL

次のテクニックを適用するなど、検索エンジン最適化 (SEO) 対応の URL を使用して検索エンジンのランクを上げることができます:⁶

- /humanresources/policies/ など、トピックのカテゴリを示すパス内の単語の階層を使用して、人事方針のカタログを表示します。
- URL クエリ文字列パラメーターは使用しないようにします。
- PDF ファイル用の .pdf などのメディア用は除いて、.aspx や .ashx などの拡張子は使用しないようにします。
- URL の最後は終了スラッシュ文字 ("/") にします。
- 単一コンテンツまたはメディア アイテムに複数の URL を使用しないようにします。

⁶ これらのテクニックの適用例については、<http://trac.sitecore.net/LinkProvider> を参照してください。

2.3 IIS および ASP.NET URL

IIS は、次の 3 つのいずれかの方法で HTTP 要求に応答します：

- IIS は、ディスク上のファイルを処理します。
- IIS は、ディスク上のファイルを処理したり、エラー メッセージによって応答したり、IIS にコントロールを返すことができる ASP.NET などのプロセスを呼び出します。
- IIS は、ファイルのコンテンツをリダイレクトおよび表示したり、ハードコードされたエラー メッセージを表示したりすることにより、エラーを処理します。

デフォルトでは、クラシック ASP.NET パイプラインがある IIS6 と IIS7 では両方とも、ASP.NET のみを使用して、.aspx や .ashx などの特定の拡張子で終わるファイルパスがある要求を処理します。他の拡張子で終わるか拡張子のない要求の場合、IIS は IIS Web サイトのドキュメント ルートまたはサブディレクトリ内のファイルを処理しようとします。

Sitecore は ASP.NET アプリケーションであるため、IIS が ASP.NET を使用して要求を処理しない場合、Sitecore は要求を処理できません。このため、IIS、ASP.NET および Sitecore によって HTTP 404 Page Not Found 状態の処理方法が異なる場合など、明白な不整合が生じる可能性があります。

次のいずれかのテクニックを使用すると、ASP.NET を使用して別の要求を処理するよう IIS を設定することができます：

- IIS7 またはそれ以降の統合モードの ASP.NET マネージ パイプライン
- IIS HTTP 404 ページ
- URL を再作成する ISAPI フィルター
- IIS ワイルドカード

メモ

ASP.NET を使用して別の要求を処理するよう IIS を設定すると、攻撃対象領域が広がったり、マシン リソースの消費量が増えたりする可能性があります。

2.3.1 IIS7 またはそれ以降の統合モードの ASP.NET マネージ パイプライン

IIS7 以降 (Windows Vista、Windows 2008、Windows 7 またはそれ以降) とともに Sitecore 6.2 またはそれ以降を使用する場合、ASP.NET マネージ パイプラインの統合モードを設定することにより、IIS が ASP.NET を使用してすべての要求を処理するようにすることができます。

Web サイトに関連付けられたアプリケーション プールに対して統合モードのマネージ パイプラインを設定する方法:

1. IIS 管理コンソールで、Web サイトを選択して **[基本設定]** をクリックします。**[サイトの編集]** ダイアログが表示されます。
2. **[サイトの編集]** ダイアログで、**[アプリケーション プール]** の値をメモして **[キャンセル]** をクリックします。
3. IIS 管理コンソールで、**[アプリケーション プール]** を選択してから、前のステップでメモしたアプリケーション プールをダブルクリックします。**[アプリケーション プールの編集]** ダイアログが表示されます。
4. **[アプリケーション プールの編集]** ダイアログで、**[マネージ パイプライン モード]** を **[統合]** に設定して **[OK]** をクリックします。

メモ

ASP.NET マネージ パイプラインのクラシック モードを使用する場合は、「IIS7 またはそれ以降のクラシック マネージ パイプライン モード (Windows Vista、Windows 2008 および Windows 7)」のセクションを参照してください。

2.3.2 IIS HTTP 404 ページ

IIS HTTP 404 ページを、.aspx などの ASP.NET 拡張子が含まれる URL (/default.aspx など) に設定することにより、IIS が ASP.NET を使用してファイルに未対応の別の要求を処理するよう設定することができます。⁷ IIS 404 ページの URL の最後にある拡張子が原因で IIS が ASP.NET を使用して要求を処理する場合、IIS は、このファイルが存在するかどうかとは関係なく、ASP.NET を呼び出して要求を処理します。

メモ

web.config に定義されている preprocessRequest パイプライン内の FilterUrlExtensions プロセッサに渡されるパラメーターを更新することにより、Sitecore が特定の拡張子を持つ要求を処理できるようにします。たとえば、.htm および .html 拡張子を持つ要求を Sitecore が処理できるようにする方法は、次のとおりです:

```
<processor
  type="Sitecore.Pipelines.PreprocessRequest.FilterUrlExtensions, Sitecore.Kernel">
  <param desc="Allowed extensions (comma separated)">aspx, ashx, asmx, htm,
html</param>
  ...
```

任意の拡張子を持つ要求を Sitecore が処理できるようにする方法は、次のとおりです:

```
<processor
  type="Sitecore.Pipelines.PreprocessRequest.FilterUrlExtensions, Sitecore.Kernel">
  <param desc="Allowed extensions (comma separated)">*</param>
  <param desc="Blocked extensions (comma separated)"> </param>
```

⁷ Sitecore を使用した HTTP 404 Page Not Found 状態の処理の詳細については、<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Handling%20HTTP%20404.aspx> にある『HTTP 404 への対処』を参照してください。

メモ

Sitecore は、コンテンツ アイテムを確認しようとする前に URL から拡張子を削除します。/Sitecore/Content/Home/Item が存在する場合、URL 内のパスが /item.aspx、/item、/item/ または /item.html のいずれであっても関係なく、別の拡張子があるか拡張子がない要求を IIS が処理するよう設定すると、Sitecore はこのアイテムをこの要求のコンテキスト アイテムとして設定します。

Windows XP または Windows 2003 で IIS HTTP 404 ページを設定する方法:

1. Windows デスクトップで、Windows の [スタート] ボタンをクリックして [ファイル名を指定して実行] をクリックします。Windows の [ファイル名を指定して実行] ダイアログが表示されます。
2. Windows の [ファイル名を指定して実行] ダイアログで、inetmgr と入力して [OK] をクリックします。IIS 管理コンソールが表示されます。
3. IIS 管理コンソールで、マシンを右クリックしてすべての Web サイトに変更を適用します。または、マシン名と [Web サイト] の両方を展開し、Web サイトを右クリックして個々の Web サイトに変更を適用します。次に、[プロパティ] をクリックします。[Web サイトのプロパティ] ダイアログが表示されます。
4. [Web サイトのプロパティ] ダイアログで、[カスタム エラー] タブをクリックします。404 エラーごとに、[編集] をクリックしてから、[メッセージの種類] を URL、[URL] を /default.aspx に設定します。

Windows Vista または Windows 2008 で IIS HTTP 404 ページを設定する方法:

注意

一部のバージョンの Windows Vista 上で IIS 管理コンソールを介して変更を行うと、web.config からテキスト値が削除される可能性があります。⁸

メモ

Windows Vista で HTTP 404 ページを使用するには、**Internet Information Services/World Wide Web Servers/Common HTTP Features/HTTP Errors** をインストールします。Windows 2008 で HTTP 404 ページを使用するには、**Web Server/Common HTTP Features/HTTP Errors** をインストールします。Windows 7 で HTTP 404 ページを使用するには、**World Wide Web Services/Common HTTP Features/HTTP Errors** をインストールします。

1. Windows デスクトップで、Windows の [スタート] ボタンをクリックします。Windows のスタートが表示されます。
2. Windows の [スタート] メニュー上のテキスト フィールドに inetmgr と入力し、Enter キーを押します。IIS 管理コンソールが表示されます。

⁸ web.config の破損の原因となる可能性がある IIS の不具合の詳細については、<http://sdn.sitecore.net/Products/Sitecore%20V5/Sitecore%20CMS%205,-d-,3/Installation/Installing%20Sitecore%20on%20Vista.aspx> を参照してください。

3. IIS 管理コンソールの **[接続]** ツリーで、マシンを選択してすべての Web サイトに変更を適用します。または、マシンと **[サイト]** を展開し、個々の Web サイトをクリックして個々の Web サイトに変更を適用します。
4. IIS 管理コンソールの **[IIS]** の下で **[エラー ページ]** をダブルクリックします。エラー コードのリストが表示されます。
5. エラー コードのリストで、**[404]** エントリをダブルクリックします。**[カスタム エラー ページの編集]** ダイアログが表示されます。
6. **[カスタム エラー ページの編集]** ダイアログで、**[URL の実行]** または **[このサイトで URL を実行]** を選択し、**[パス]** または **[URL]** として /default.aspx と入力し、**[OK]** をクリックします。
7. IIS 管理コンソールで、**[操作]** 内で **[機能設定の編集]** をクリックします。**[エラー ページ設定の編集]** ダイアログが表示されます。
8. **[エラー ページ設定の編集]** ダイアログで **[カスタム エラー ページ]** を選択します。

2.3.3 IIS ワイルドカード

統合モードを除いて、IIS 設定では、.aspx、.ashx および .asmx などの特定の拡張子を、ASP.NET を実装する ISAPI フィルターにマッピングします。ワイルドカード拡張子を設定することにより、IIS が ASP.NET ISAPI フィルターを使用してすべての要求を処理するよう設定することができます。⁹

重要

IIS が ASP.NET を使用して別の要求を処理するよう設定すると、パフォーマンスおよびセキュリティに影響する可能性があります。

重要

メディア アイテムを含む Sitecore アイテムに対して使用するファイル名拡張子ごとに、このファイルが存在することを必要とする IIS 管理コンソール内のオプションを無効にする必要がある場合があります。

重要

ファイルと Sitecore アイテムの両方に対して使用するファイル名拡張子ごとに web.config 内の StaticFileHandler を設定する必要がある場合があります。たとえば、.htm 拡張子を持つ要求を Sitecore アイテムまたは静的ファイルとして処理する方法は、次のとおりです：

```
<httpHandlers>
  <add verb="GET,HEAD" path="*.htm" type="System.Web.StaticFileHandler, System.Web,
    Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
  ...
```

⁹ ワイルドカード拡張子の詳細については、<http://professionalaspnet.com/archive/2007/07/27/Configure-IIS-for-Wildcard-Extensions-in-ASP.NET.aspx> を参照してください。

IIS 5.1 またはそれ以前 (Windows XP)

IIS でワイルドカード拡張子を設定する方法 (Windows XP):

1. IIS 管理コンソールで、Web サイトを右クリックして **[プロパティ]** をクリックします。**[Web サイトのプロパティ]** ダイアログが表示されます。
2. **[Web サイトのプロパティ]** ダイアログで、**[ホーム ディレクトリ]** タブをクリックして **[構成]** をクリックします。**[アプリケーションの構成]** ダイアログが表示されます。
3. **[アプリケーションの構成]** ダイアログで、.aspx エントリを選択して **[編集]** をクリックします。**[アプリケーションの拡張子マッピングの追加/編集]** ダイアログが表示されます。
4. **[アプリケーションの拡張子マッピングの追加/編集]** ダイアログで、**[実行可能ファイル]** を Windows のクリップボードにコピーして **[キャンセル]** をクリックします。
5. **[アプリケーションの構成]** ダイアログで **[追加]** をクリックします。**[アプリケーションの拡張子マッピングの追加/編集]** ダイアログが表示されます。
6. **[アプリケーションの拡張子マッピングの追加/編集]** ダイアログで、Windows のクリップボードの値を **[実行可能ファイル]** に貼り付けます。
7. **[アプリケーションの拡張子マッピングの追加/編集]** ダイアログで、ドット文字の後ろにアスタリスク文字を付けて (".*") **[拡張子]** に入力します。
8. **[アプリケーションの拡張子マッピングの追加/編集]** ダイアログで、**[制限]** を選択して GET, POST と入力します。
9. **[アプリケーションの拡張子マッピングの追加/編集]** ダイアログで、**[ファイルの存在を確認する]** をオフにして **[OK]** をクリックします。

IIS6 (Windows 2003)

IIS6 でワイルドカード拡張子を設定する方法 (Windows 2003):

1. IIS 管理コンソールで、Web サイトを右クリックして **[プロパティ]** をクリックします。**[Web サイトのプロパティ]** ダイアログが表示されます。
2. **[Web サイトのプロパティ]** ダイアログで、**[ホーム ディレクトリ]** タブをクリックして **[構成]** をクリックします。**[アプリケーションの構成]** ダイアログが表示されます。
3. **[アプリケーションの構成]** ダイアログで、**[マッピング]** タブをクリックして .aspx エントリを選択して **[編集]** をクリックします。**[アプリケーションの拡張子マッピングの追加/編集]** ダイアログが表示されます。

4. [アプリケーションの拡張子マッピングの追加/編集] ダイアログで、[実行可能ファイル] を Windows のクリップボードにコピーして [キャンセル] をクリックします。
5. [アプリケーションの構成] ダイアログで [追加] をクリックします。[アプリケーションの拡張子マッピングの追加/編集] ダイアログが表示されます。
6. [アプリケーションの拡張子マッピングの追加/編集] ダイアログで、Windows のクリップボードの値を [実行可能ファイル] に貼り付けます。
7. [アプリケーションの拡張子マッピングの追加/編集] ダイアログで、[拡張子] にアスタリスク文字 ("*") を入力します。
8. [アプリケーションの拡張子マッピングの追加/編集] ダイアログで、[制限] を選択して GET, POST と入力します。
9. [アプリケーションの拡張子マッピングの追加/編集] ダイアログで、[ファイルの存在を確認する] を無効にして [OK] をクリックします。

IIS7 またはそれ以降のクラシック マネージ パイプライン モード (Windows Vista、Windows 2008 および Windows 7)

クラシック マネージ パイプライン モードのアプリケーション プールを使用して IIS7 またはそれ以降でワイルドカード拡張子処理を設定する方法:

重要

IIS ワイルドカードを Microsoft Windows Vista で使用するには、**World Wide Web Services/Application Development Features/ISAPI Extensions** をインストールします。IIS ワイルドカードを Microsoft Windows 2008 で使用するには、**Web Server/Application Development/ISAPI Extensions** および **Web Server/Application Development/ISAPI Filters** をインストールします。IIS ワイルドカードを Windows 7 で使用するには、**Internet Information Services/World Wide Web Services/Application Development Features/ISAPI Extensions** および **Internet Information Services/World Wide Web Services/Application Development Features/ISAPI Filters** をインストールします。

1. IIS 管理コンソールで、Web サイトをクリックし、[ハンドラ マッピング] をダブルクリックし、.aspx エントリを選択し、[編集] をクリックします。[スクリプト マップの編集] ダイアログが表示されます。
2. [スクリプト マップの編集] ダイアログで、[実行可能ファイル] を Windows のクリップボードにコピーして [キャンセル] をクリックします。
3. [スクリプト マップの追加] をクリックします。[スクリプト マップの追加] ダイアログが表示されます。
4. [スクリプト マップの追加] ダイアログで、[要求パス] にアスタリスク文字 ("*") を入力します。
5. [スクリプト マップの追加] ダイアログで、Windows のクリップボードの値を [実行可能ファイル] に貼り付けます。

6. **[スクリプト マップの追加]** ダイアログで、Wildcard ASP.NET ISAPI などのハンドラ マッピングの名前を **[名前]** フィールドに入力します。
7. **[スクリプト マップの追加]** ダイアログで **[要求の制限]** をクリックします。**[要求の制限]** ダイアログが表示されます。
8. **[要求の制限]** ダイアログで、**[マップ]** タブをクリックし、**[要求のマップ先が次の場合のみハンドラを呼び出す]** を無効にします。
9. **[要求の制限]** ダイアログで、**[動詞]** タブをクリックし、**[次の動詞のうちの 1 つ]** を選択し、GET, POST と入力し、**[OK]** をクリックします。

2.3.4 URL を再作成する ISAPI フィルター

IIS が URL の処理方法を決定する前に URL を再作成する ISAPI フィルターを実装することにより、IIS が ASP.NET を使用して別の要求を処理するよう設定することができます。¹⁰

メモ

この文書では、ISAPI フィルターを使用した URL の再作成については説明しません。

¹⁰ ISAPI フィルターを使用した URL の再作成の詳細については、<http://sdn.sitecore.net/Scrapbook/Friendlier%20Marketing%20URLs.aspx> を参照してください。