

Microsoft®

インターネット Web サーバー

構築ガイドライン

【ドラフト版】

～ 第 6 章 : アプリケーションの実行環境を設定しよう

※本ガイドラインは各章の先行ドラフト版公開を行い、全章の公開後、正式版文書としてまとめを行い、再度公開します。

第 1 版 2010/05

マイクロソフト株式会社

免責事項: このドキュメントの内容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。このドキュメントに記載されている情報 (URL 等のインターネット 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。お客様がこのドキュメントを運用した結果の影響については、お客様が負うものとします。別途記載されていない場合、このドキュメントで例として挙げられている企業、組織、製品、ドメイン名、電子メール アドレス、ロゴ、人物、地名、およびイベントは、架空のものです。それらが、いずれかの実際の企業、組織、製品、ドメイン名、電子メール アドレス、ロゴ、人物、地名、あるいはイベントを指していることはなく、そのように解釈されるべきではありません。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お願いします。

第 6 章 アプリケーションの実行環境を設定しよう

当初、**Web** (World Wide Web) は、HTML で記述されたドキュメントを配信する目的で使用されてきました。やがて **CGI** (Common Gateway Interface) が登場し、クライアントからのリクエストに応じてコンテンツの生成が動的に行えるようになると、その仕組みを利用してアプリケーションが作成されるようになってきました。

これら Web サーバーで動作するアプリケーションを、一般的に Web アプリケーションと呼びます。

Web アプリケーションの仕組みは、時代の経過とともに進化し、当初 CGI と標準出力での組み合わせで行われていた処理は、**ASP** や **Java Servlet** のように サーバプロセスに常駐させたインタプリタで処理を行うものや、ASP.NET のようにネイティブコードにコンパイルされて動作するものなどが出てきました。

IIS 7 では、環境を適切に設定することにより、様々な種類の Web アプリケーションを同一のマシン上で動作させることができます。

Web で使用されるアプリケーションとは？

現在、インターネットで使用されているアプリケーションは、Web サーバー上で動作する **サーバーサイド アプリケーション** と、Web ブラウザー内で動作する **クライアントサイド アプリケーション** に分けることができます。さらには、クライアントサイドで動作するアプリケーションは、Web ブラウザーがサポートしているスクリプト言語で書かれたものと、ブラウザーに追加したランタイム (実行環境) が実行を行うプラグイン式のものに分けることができます。

サーバーサイドのアプリケーション	クライアントサイドのアプリケーション	
ASP.NET、ASP、PHP、Perl などで記述されたもの	スクリプト	JavaScript,VB スクリプト,TCL などで記述されたもの
	プラグイン	Silverlight、Flash、Java アプレット等

表:サーバーサイドとクライアントサイド アプリケーションの種類

クライアントサイドで動作するアプリケーションは、動作に Web サーバーのリソース (資源) を必要としないため、基本的にどのような Web サーバーでホストされても動作することができます。

対してサーバーサイド アプリケーションは、動作する際に Web サーバーのリソースを利用して実行されるため Web サーバーにはアプリケーションを動作させるためのランタイムと、セキュリティやリソースの配分、その他の環境の設定が必要になります。

サーバーサイド アプリケーションは、サーバーの CPU 、メモリをはじめ、サーバーから接続可能なデータベースやストレージ、様々なサービスなども使用可能であるため、大量のデータを扱ったり、複雑な処理を行うのに向いています。また、クライアントサイド アプリケーションの機能が、単一のユーザーに対してのものであるのに対し、サーバーサイド アプリケーションは複数のクライアントに機能を提供することが可能です。

Web 上でアプリケーションを使用したサービスを行うのであれば、サーバーサイド アプリケーションで行う必要があります。

そのため一般的に “Web アプリケーション” とは、サーバーサイド アプリケーションのことを指します。

IIS で実行できる Web アプリケーションの種類

Web アプリケーションが実行に使用するランタイムには様々なものがあります。

IIS は、標準機能として **Active Server Pages** (以降 ASP)、と **ASP.NET** のランタイム(※)および実行環境を提供しています。

(※)アプリケーションを実行させるのに必要なファイルの集まり。

その他にも、ランタイム向けの実行環境として **FastCGI** というモジュールを提供しており、たとえば **PHP** のランタイムと組み合わせることにより PHP で作成されたアプリケーションを安全かつ効率よく IIS 7 上で実行することができます。

また、その他のランタイムでも、ランタイム自身が提供する ISAPI などの IIS 用のインフラストラクチャを使用して、アプリケーションを動作させることができます。

ハンドラー マッピングの考え方

※この項目はアプリケーションが実行される仕組みの説明となりますので、興味の無い方は読み飛ばしていただいて結構です。

IIS 7 では、ハンドラーと呼ばれる機構が、アプリケーションへのリクエストを処理します。たとえば、IIS 7 で PHP アプリケーションを実行できるようにするには、ハンドラーの設定で、「拡張子 *.php がリクエストされたら FastCGI を使用して php-cgi.exe を実行する」というようにマッピングを行います。これをハンドラー マッピングと言います。

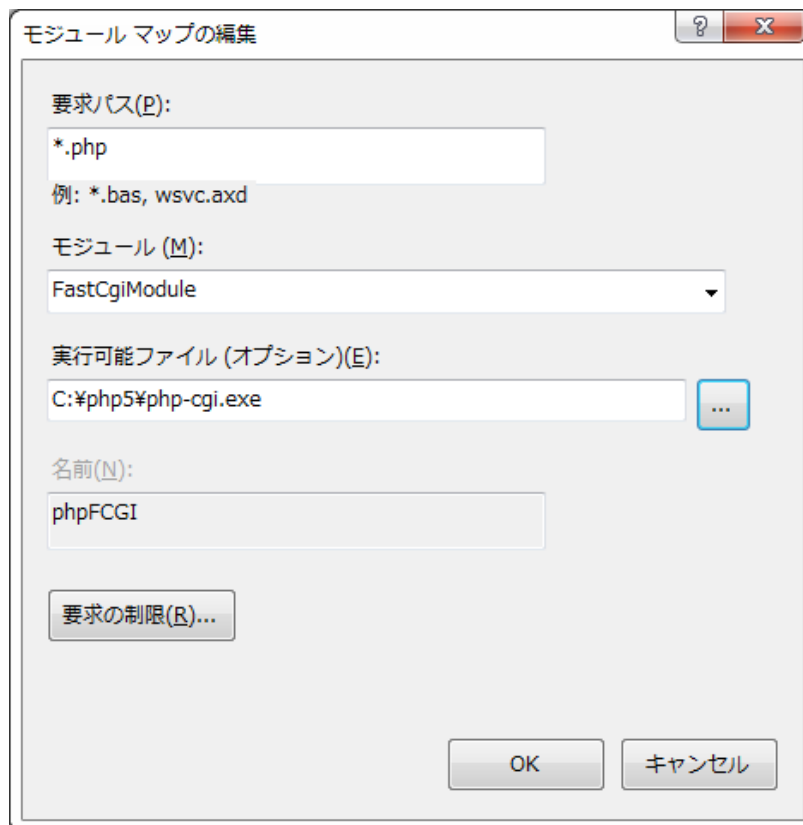


図:[ハンドラー マッピング] 内のモジュールマップの編集画面

IIS に配置されたコンテンツファイルは、ハンドラーを設定することで、サーバー内でコンテンツの解釈と処理が行われるようになります。

ハンドラーの設定されていないコンテンツファイルは、MIME の設定に従い(※)、ファイルがそのままレスポンスされます。

(※) MIME が設定されていない場合は HTTP 404 Not Found. を返します。

IIS での "アプリケーション" とは

IIS マネージャーのツリービューでは、Web サイトの下に、**仮想ディレクトリ** と **アプリケーション** を追加することができます。

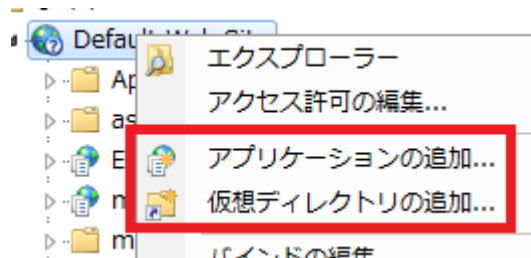
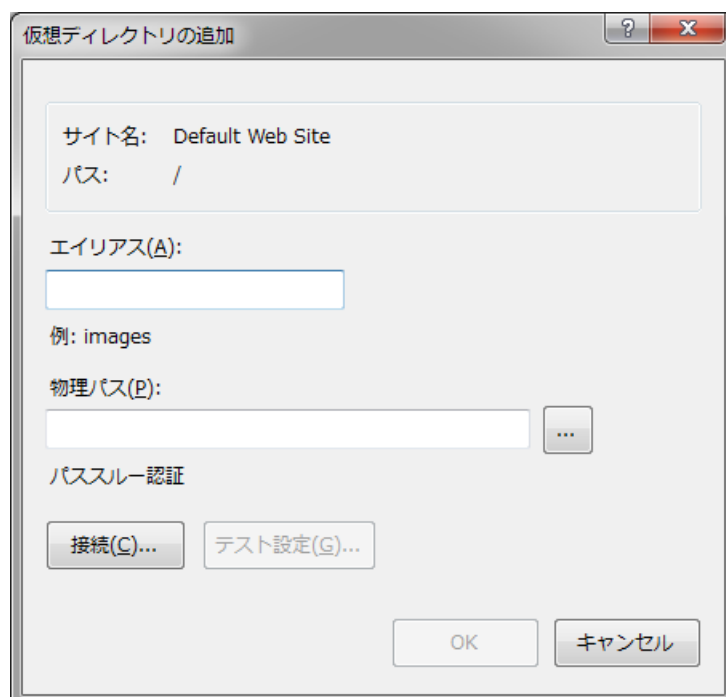
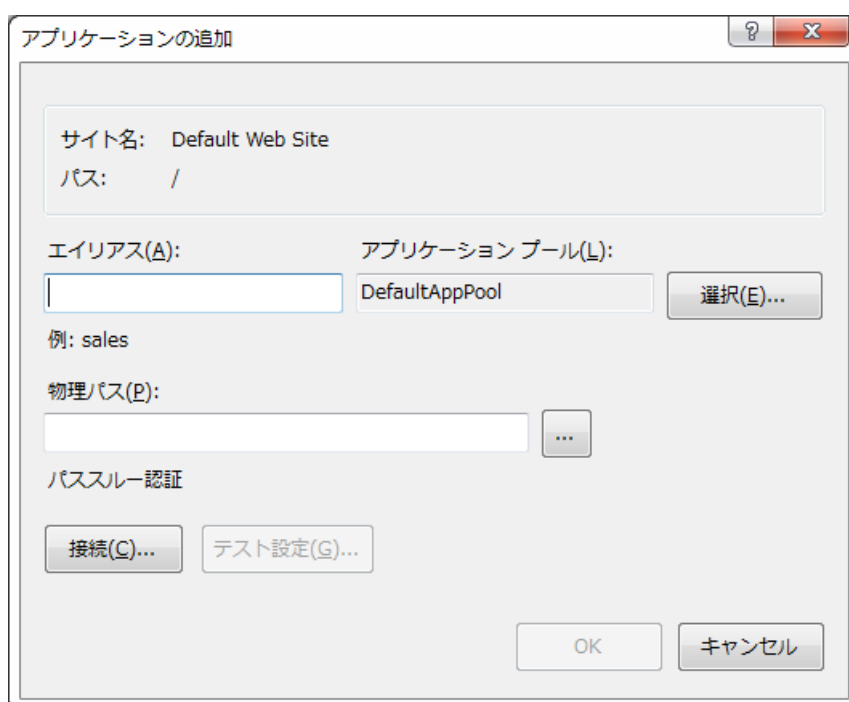


図:Default Web Site で右クリックした際に表示されるメニュー

仮想ディレクトリとアプリケーションはいずれも、物理フォルダーを指定して作成し、フォルダーに配置されたファイルをコンテンツとして HTTP でホストします (Web に公開します)。



図：仮想ディレクトリの追加 ダイアログ ボックス



図：アプリケーションの追加 ダイアログ ボックス

この二つの違いは、仮想ディレクトリがコンテンツのレスポンスに関する設定しか持っていないのに対し、アプリケーションは、サーバーサイドでプログラムを実行するためのアプリケーションプールの設定や、ハンドラー マッピングの設定がされているということです。

つまり、IIS マネージャーのツリービューにおけるアプリケーションとは、サーバーサイドのアプリケーションを実行するための設定がなされた仮想ディレクトリであるといえます。

実際のところ、IIS マネージャーのツリービューで仮想ディレクトリを選択し、マウスの右ボタンをクリックすると、コンテキストメニューに「**アプリケーションへの変換**」というメニューが表示され、同メニューを選択することで、仮想ディレクトリをアプリケーションに変換することができます。

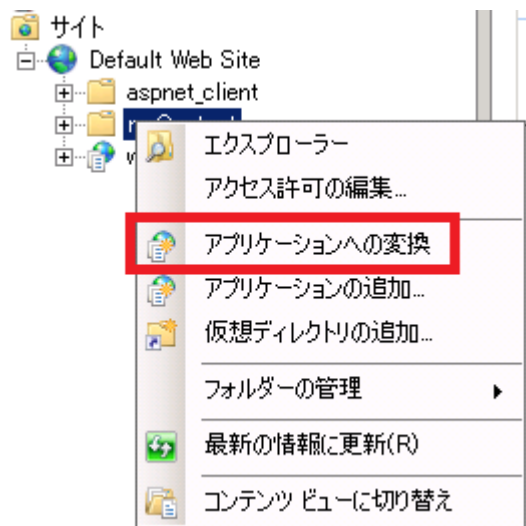


図:仮想ディレクトリ で右クリックした際に表示されるメニュー

逆に、アプリケーションを選択し、マウスの右ボタンをクリックし、コンテキストメニューから「**削除**」を選択すると、アプリケーションを仮想ディレクトリに変更できます。

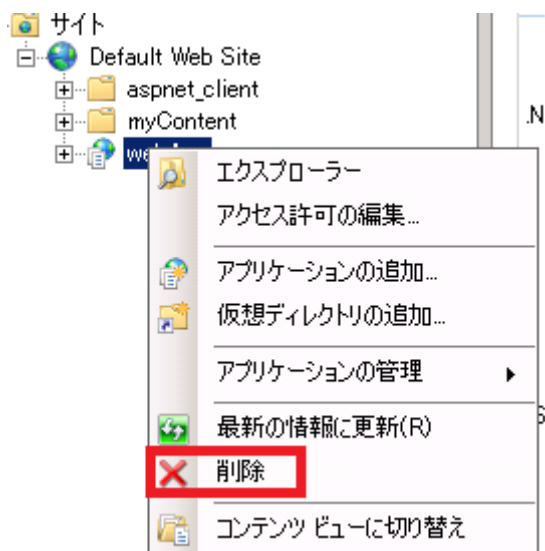


図:アプリケーション で右クリックした際に表示されるメニュー

アプリケーションの実行環境

IIS 7 では **ASP.NET**、**ASP** をはじめ、**PHP** などのアプリケーション用に **FastCGI** という実行環境を標準で提供しています。

ここでは ASP.NET と ASP そして FastCGI を使用した PHP アプリケーションの実行環境をセットアップする手順を紹介します。

ASP.NET

ASP.NET は **.NET Framework** 上に構築された Web アプリケーションを開発するためのフレームワークであり、アプリケーションを実行するためのランタイムでもあります。ASP.NET を使用すると HTML で記述された UI をもつ一般的な Web アプリケーションはもちろん、UI を持たない **SOAP** (Simple Object Access Protocol) や **REST** (Representational State Transfer) といった HTTP 上のプロトコルを使用してホストされるサービス アプリケーションなども開発することができます。また、IIS 7 からは ASP.NET を使用して IIS 自身の機能を拡張することもできるようになっています。

ASP.NET ランタイムで動作するアプリケーションのことを一般的に **ASP.NET アプリケーション** と呼びます。

ASP.NET アプリケーションの実行には .NET Framework が必要となりますが、Windows Server 2003 以降の Windows OS には、OS リリース時における最新のバージョンの .NET Framework を含んでいるため、基本的には追加でインストールを行う必要はありません。それ以降にリリースされる、さらに新しいバージョンの .NET Framework については、インストールして**追加**することができるようになっています。.NET Framework は、新しいバージョンをインストールしても古いバージョンを上書きすることではなく、**side-by-side** (併存) で実行することができるようになっています。そのため IIS では アプリケーションが想定している任意のバージョンの .NET Framework を指定して実行させることができます。

ASP.NET アプリケーションの実行環境は IIS 7 が標準で提供する最新の Web アプリケーション実行環境です。

環境の設定方法

IIS 7 で ASP.NET アプリケーションを実行可能とするには、IIS の追加機能をセットアップする必要があります。

以下に Windows Server 2008 R2 と、Windows 7 でのセットアップ手順を示します。

なお、セットアップ先は、IIS 7 がインストールされ、正常に動作している環境を前提としています。

Windows Server 2008 R2 の場合

1. [スタート] をクリックし、[管理ツール] をポイントして、[サーバー マネージャー] をクリックします。

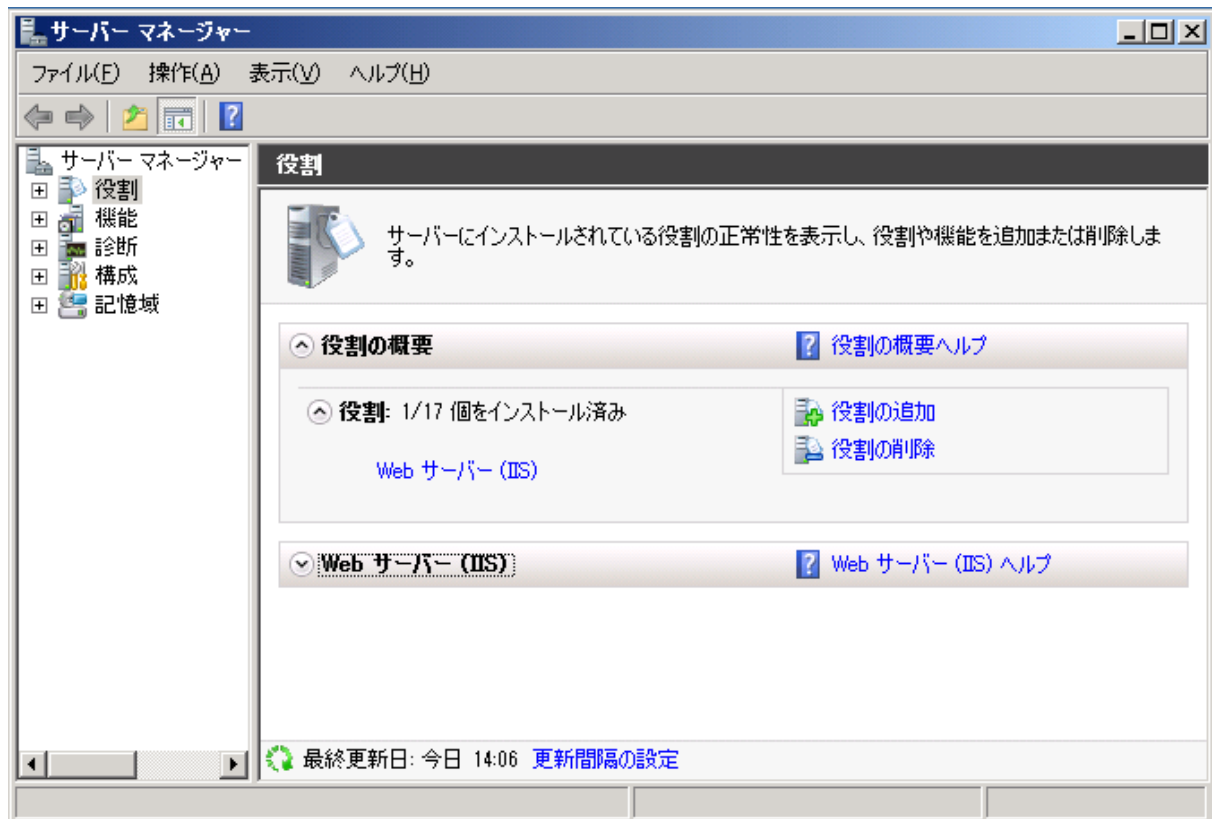


図:サーバー マネージャー

2. [Web サーバー (IIS)] を展開し [役割サービスの追加] をクリックします。

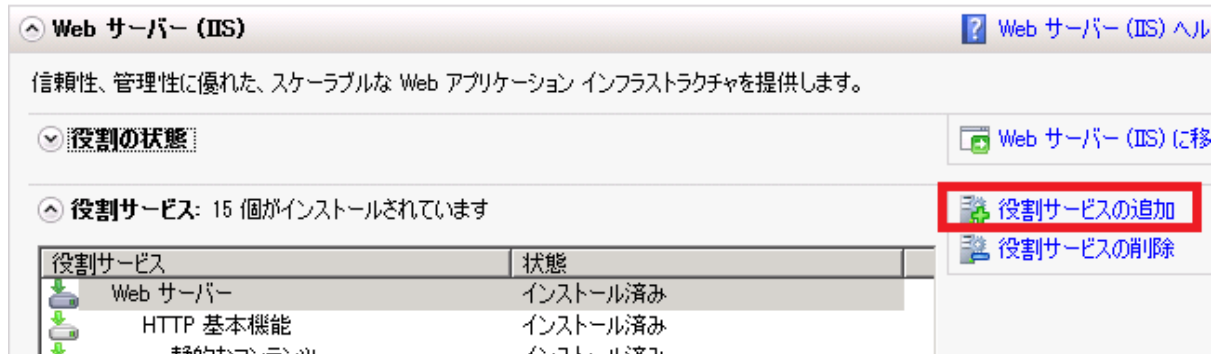


図:役割サービスの追加 メニュー

3. [役割の追加] ウィザードが起動し、[役割サービスの選択]画面が表示されるので、以下の追加役割サービスを選択し、[次へ] ボタンをクリックします。(※)

- ・ ASP.NET
- ・ .NET 拡張機能
- ・ ISAPI
- ・ ISAPI 拡張機能

(※) [ASP.NET] を選択すると以下のダイアログ ボックスが表示されるので [必要な役割サービスを追加] ボタンをクリックしてください。その際にその他の必要な役割も自動で選択されます。

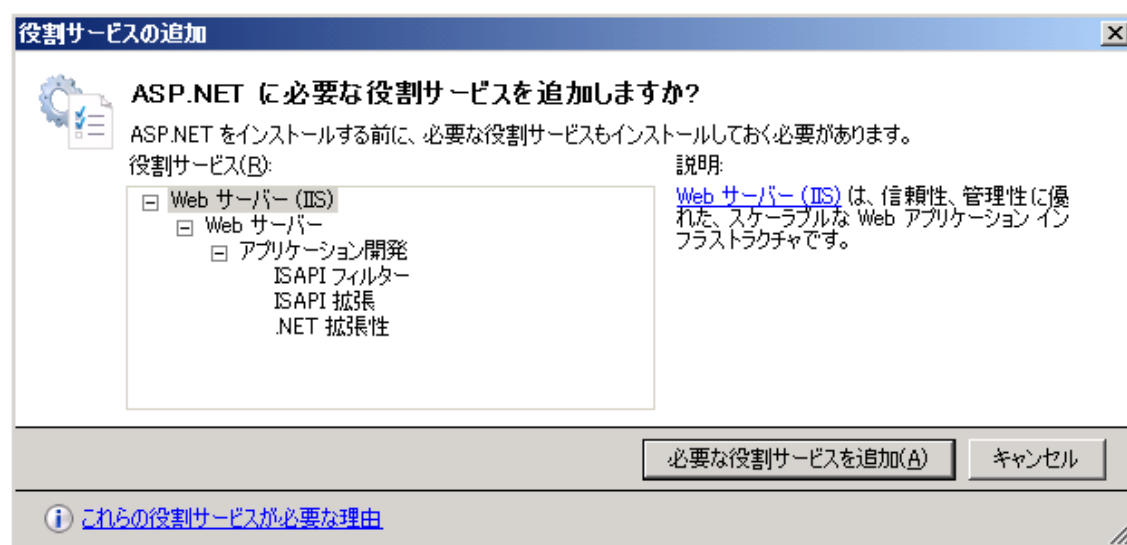


図:役割サービスの追加 確認ダイアログ ボックス

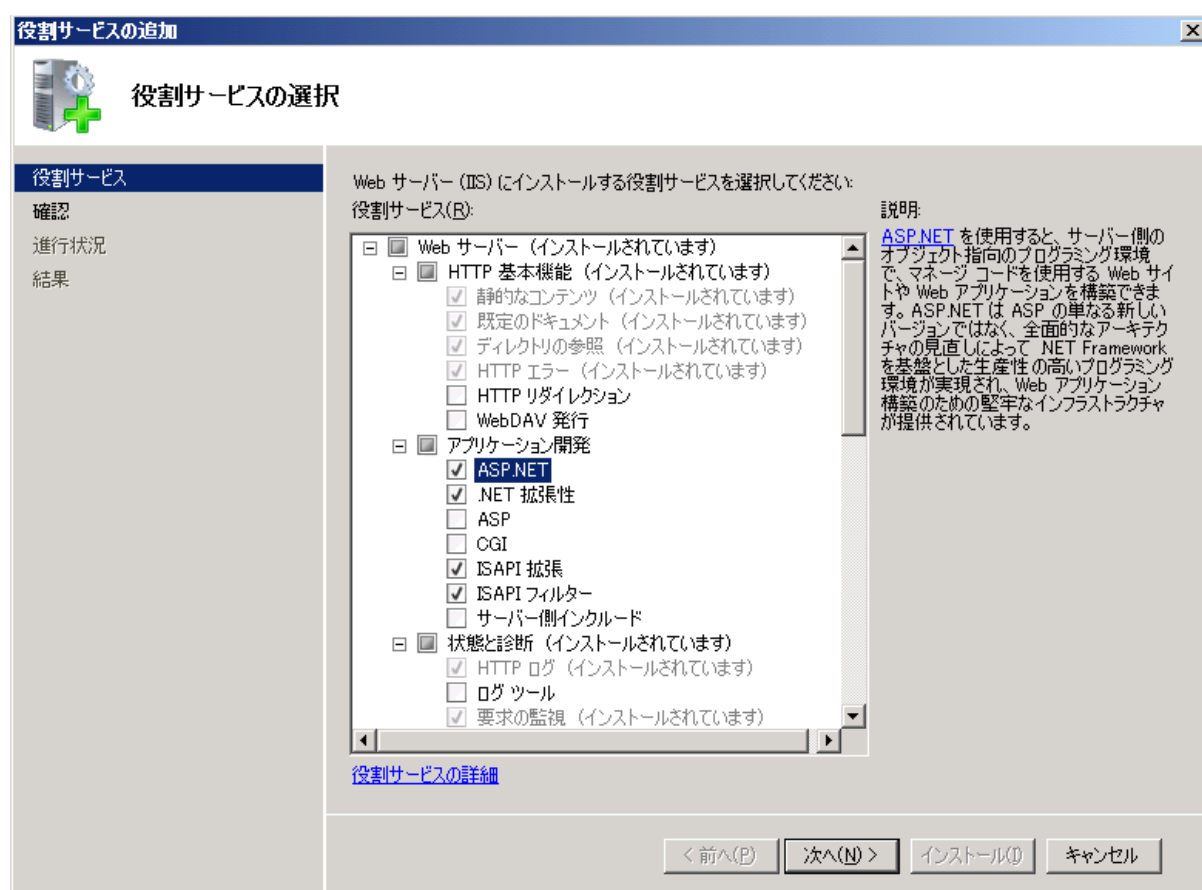


図:サービスの選択 ダイアログ ボックス

4. [インストール オプションの確認] 画面が表示されるので [インストール] ボタンをクリックしてインストールを実行します。
5. [インストールの結果] ページで、Web サーバー (IIS) の役割と必要な役割サービスのインストールが正常に完了したことを確認してから [閉じる] ボタンをクリックします。

Windows 7 の場合

1. [スタート] ボタンをクリックし、[コントロール パネル] をクリックします。
2. [コントロール パネル] 内の [プログラム] リンクをクリックします。

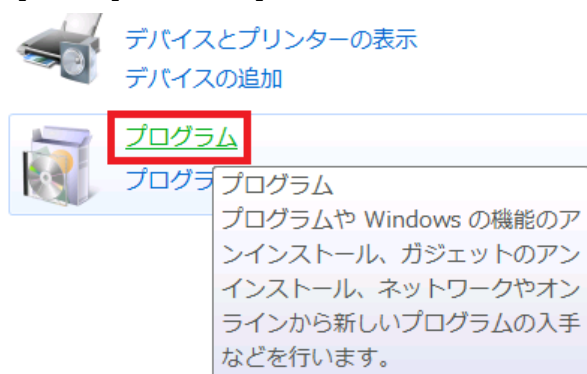


図: コントロール パネルの [プログラム] リンク

3. [Windows の機能の有効化または無効化] リンクをクリックします。



図: [Windows の機能の有効化または無効化] リンク

4. [Windows の機能] ダイアログ ボックスで、[インターネット インフォメーション サービス] のツリーより以下の追加機能を選択します。
 - ASP.NET
 - .NET 拡張機能
 - ISAPI フィルター
 - ISAPI 拡張機能

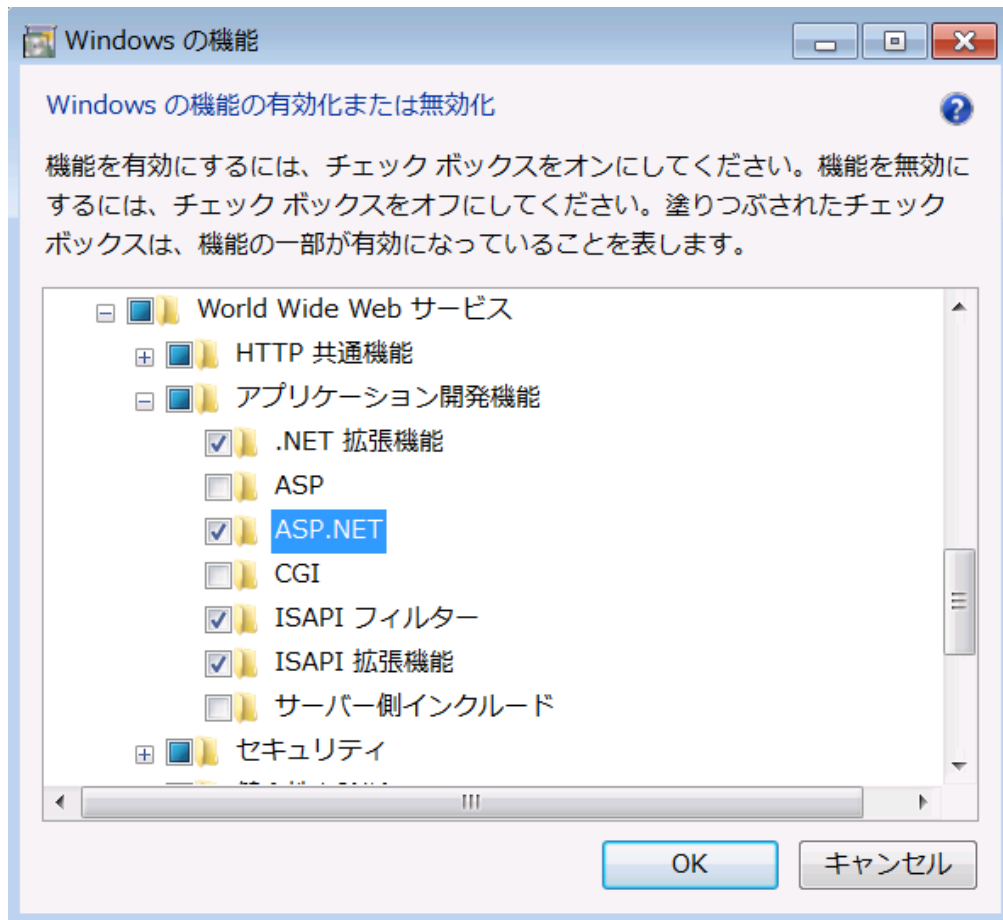


図:Windows の機能 選択ダイアログ ボックス

5. [OK] をクリックするとインストールが行われます。

サンプルの実行

ASP.NET アプリケーションを実行するには、プログラムコードを記述したファイルを**アプリケーション**として公開している仮想ディレクトリに配置する必要があります。

ASP.NET をアプリケーション配置するための具体的な手順は以下のとおりです。

1. エクスプローラーを “管理者として実行” し、C:\inetpub\wwwroot 下に “webApp” という名前でフォルダーを作成します。
2. テキスト エディターに以下のコードを貼り付け、“Hello.aspx” という名前で保存します。

```
<%@ Page Language="C#" %>
<html>
<head runat="server">
    <title></title>
</head>
```

```

<body>
  <form id="form1" runat="server">
    <h3>
      <%= "Hello World." %>
    </h3>
  </form>
</body>
</html>

```

Hello.aspx のソースコード

3. 作成した Hello.aspx ファイルを c:\inetpub\wwwroot\webApp フォルダにコピーします。
4. IIS マネージャーを起動し、画面左のツリービューを展開し、“Default Web Site” 下の “webApp” 仮想ディレクトリを選択します。
5. マウスの右ボタンをクリックし、表示されたコンテキスト メニューより [アプリケーションへの変換] をクリックします。

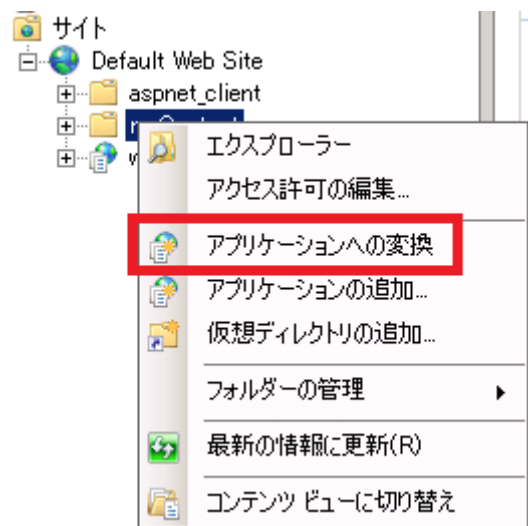


図:アプリケーションへの変換

6. [アプリケーションへの変更] ダイアログ ボックスが表示されるので既定のまま [OK] ボタンをクリックします。

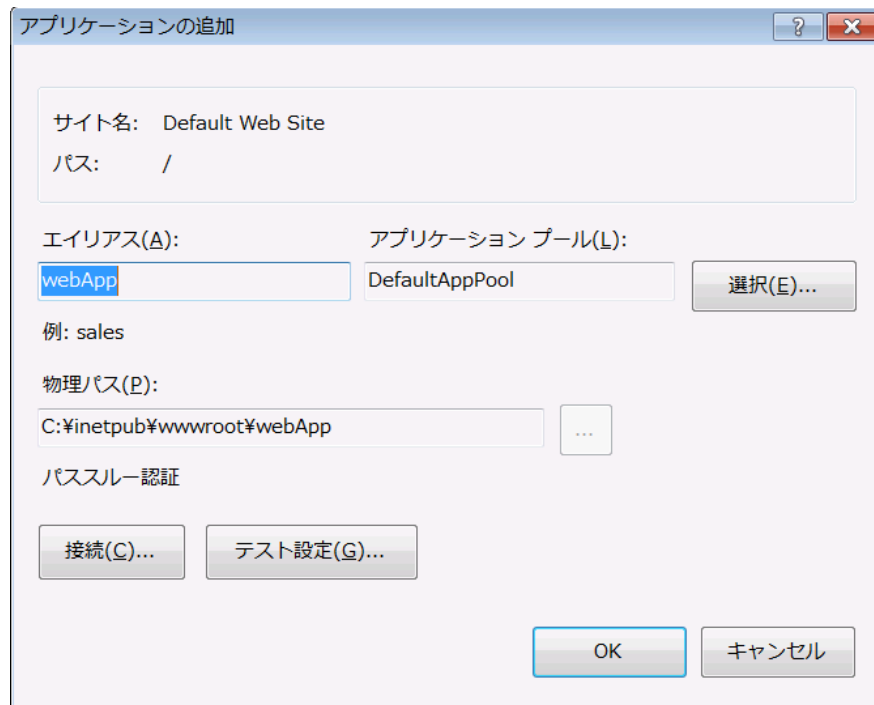


図:アプリケーションの追加 ダイアログ ボックス

7. ブラウザーを起動し、URL ボックスに以下の URL を入力し、[Enter] キーを押下します。

`http://localhost/webapp/hello.aspx`

8. ブラウザーに “Hello World.” と表示されることを確認してください。

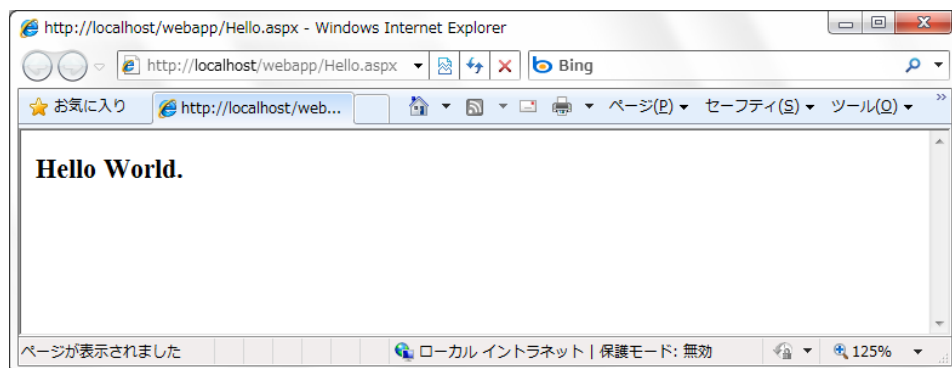


図 : ブラウザーでサンプルを表示

エラーログ

ASP.NET ランタイムは 発生したエラー、警告等を イベント として認識し、その情報を Windows OS のイベント ログとして出力します。ASP.NET のイベント ログ内には 発生時刻、メッセージ、イベント コードのほか、スタックトレース等の問題に関する詳細情報が記録されています。



図:ASP.NET のイベント ログ [全般] タブ

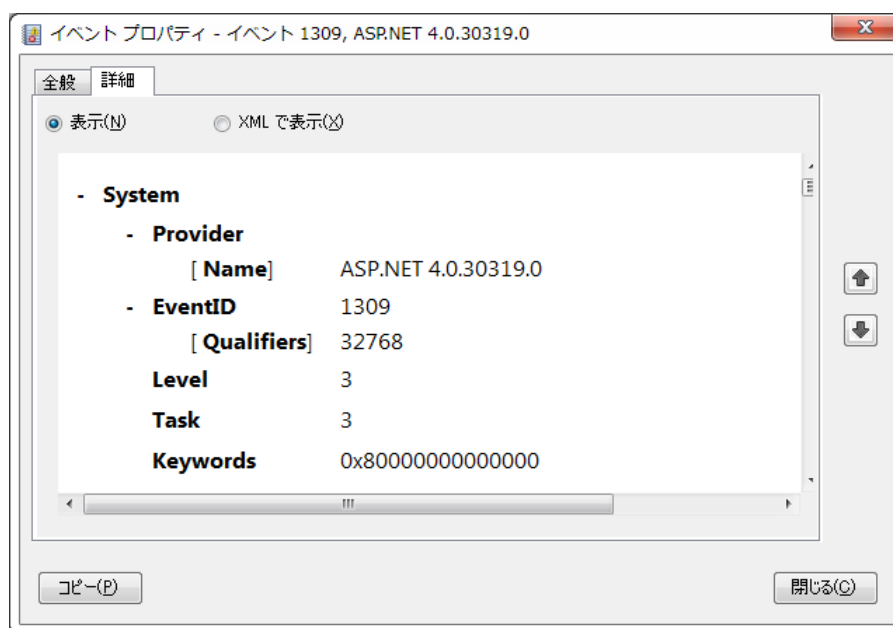


図:ASP.NET のイベント ログ [詳細] タブ

ASP.NET が出力するイベント ログの内容を確認するには、Windows の **管理ツール** 内にある **イベントビューアー** を使用します。具体的な手順は以下のとおりです。

1. Windows の [スタート] ボタンをクリックし、[コントロール パネル] を選択します。
2. コントロール パネルのウィンドウが表示されるので、ウィンドウ右上にある [表示方法] ドロップダウン リストボックスをクリックし、[大きいアイコン] を選択します。

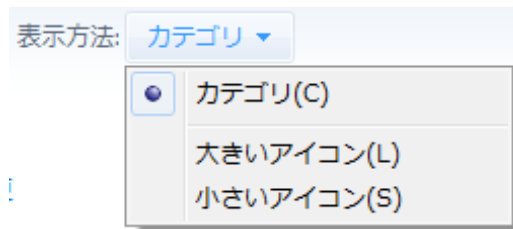


図:コントロール パネルの 表示方法 ドロップダウン リストボックス

3. コントロール パネルのウィンドウから [管理ツール] のアイコンをダブルクリックします。



図:管理ツール アイコン

4. [管理 ツール] のウィンドウが表示されるので、同ウィンドウ内から [イベント ビューアー] のショートカットアイコンをダブルクリックします。



図:イベント ビューアーのショートカットアイコン

5. [イベント ビューアー] のウィンドウが表示されるので、画面左のツリービューを展開し、[Windows ログ] 中の [アプリケーション] を選択します。さらに画面左の [操作] パネルから [現在のログをフィルター...] リンクをクリックします。

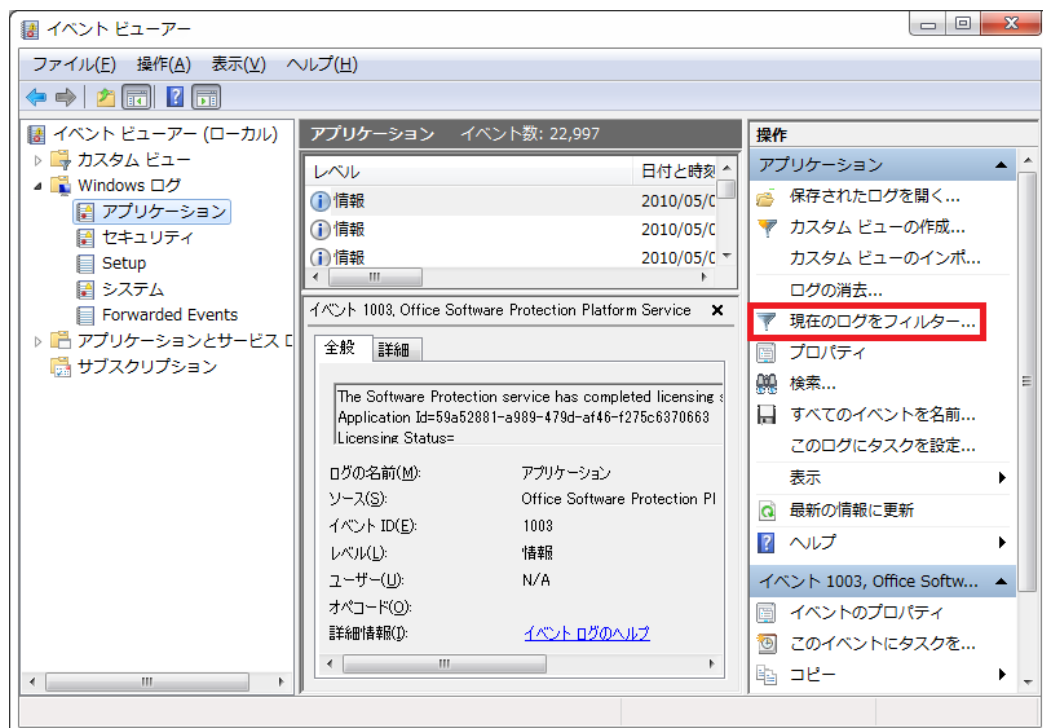


図:イベント ビューアーの画面

6. [現在のログをフィルター] ダイアログ ボックスが表示されるので、[イベント ソース] ドロップダウン リストボックスをクリックし、任意のバージョンの ASP.NET を選択し [OK] ボタンをクリックします。

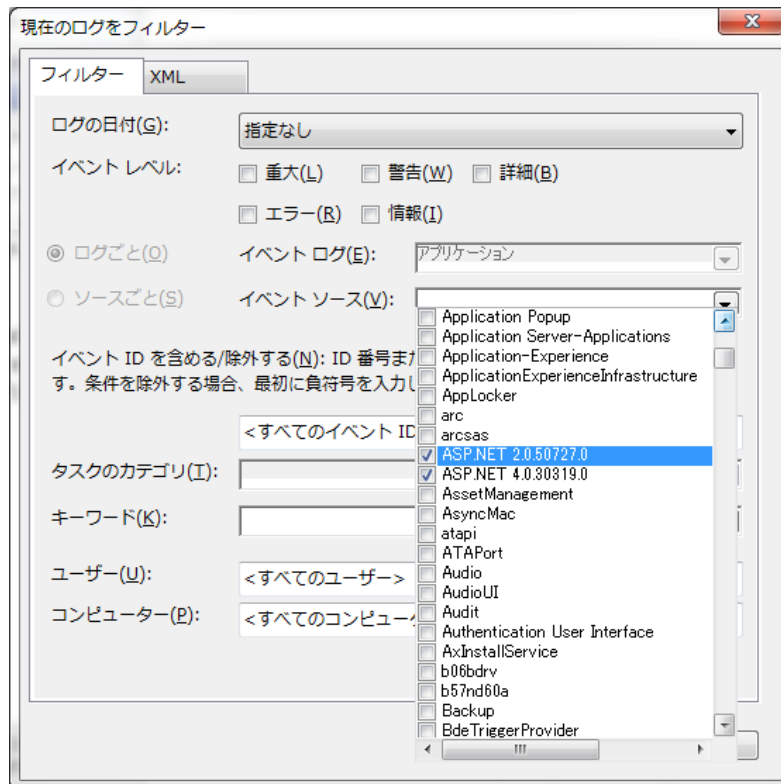


図: イベント ログのフィルター設定画面

7. イベント ログ リストに、ASP.NET に関するイベント ログのみが抽出されて表示されるので、任意のイベント ログをダブルクリックして内容を確認します。

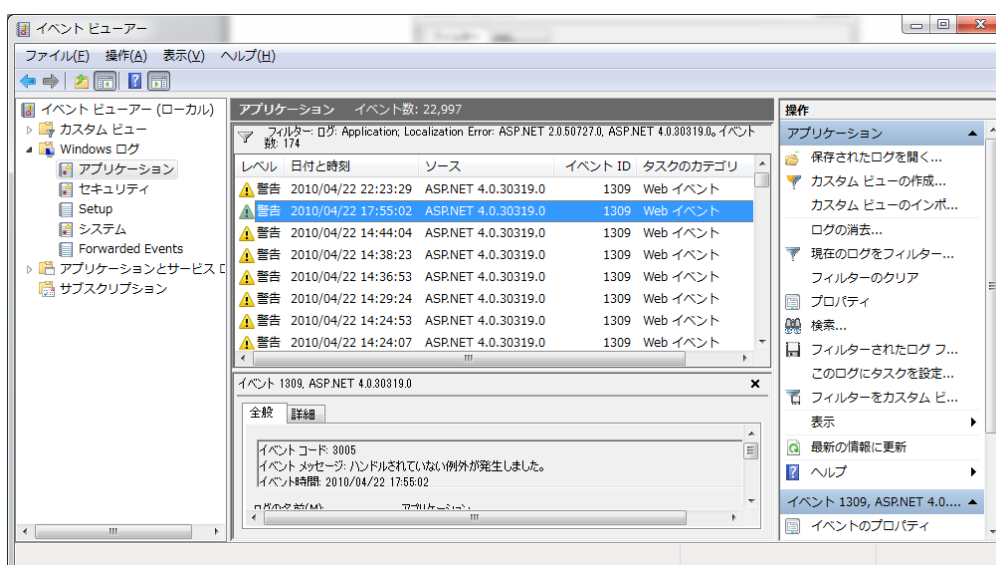


図: イベント ログ リストに ASP.NET のイベント ログが抽出されたところ



図: ASP.NET のイベント ログ

ASP.NET のイベント ログには、人間が読解可能なメッセージ形式でのログが出力されますが、使用している ASP.NET のバージョンによっては、イベント ログ内に “MessageId=(ID 番号)” という正式で情報が出力されている場合があります。

その場合は、以下の技術文書をご参照ください。

文書番号 : 835387

ASP.NET によって生成されるイベント ログのエントリ

<http://support.microsoft.com/kb/835387/ja>

また、ASP.NET 2.0 以降のバージョンでは、ASP.NET の状態監視 (ASP.NET Health Monitoring) 機能を使用して、ASP.NET アプリケーションに対し、独自の状態監視を構成することも可能になっています。

ASP.NET Health Monitoring の概要

[http://msdn.microsoft.com/ja-jp/library/bb398933\(v=VS.90\).aspx](http://msdn.microsoft.com/ja-jp/library/bb398933(v=VS.90).aspx)

ASP

ASP (Active Server Pages) は、IIS に用意されている、Web ページを動的に生成するための仕組みです。ASP の歴史は古く、最初のバージョンは 1996 年に Windows NT 4 用の IIS 3.0 と共に提供されました。以来、2002 年に ASP.NET が登場するまで IIS 上で動作する Web アプリケーションの開発に使用されていました。

現在、IIS 上で動作する Web アプリケーション開発の中心は ASP.NET に移行しており、開発ツールも提供されていませんが、過去に作成され、現在も動作している ASP アプリケーションは非常に多く存在しています。

ASP は IIS の機能の一部として、IIS 7.5 の 32 ビット、64 ビットのどちらでも正式にサポートされています。しかしながら、IIS 7 は ASP が当初実行環境として想定していた IIS 4、IIS 5.x と構造も実装も大きく異なり、また Windows OS 自体も異なるため、以前のバージョンの IIS で動作している ASP アプリケーションを IIS 7 に移行する際には、充分かつ、入念なテストを行っていただく必要があります。ASP アプリケーションの場合、.asp ページそのものの互換性は高いのですが、呼び出しているコンポーネントや部品の新環境への対応状況を必ず確認しましょう。

環境の設定方法

IIS 7 で ASP アプリケーションを実行可能とするには、IIS の追加機能をセットアップする必要があります。

以下に Windows Server 2008 R2 と、Windows 7 でのセットアップ手順を示します。

なお、セットアップ先は、IIS 7 がインストールされ、正常に動作している環境を前提としています。

Windows Server 2008 R2 の場合

1. [スタート] をクリックし、[管理ツール] をポイントして、[サーバー マネージャー] をクリックします。

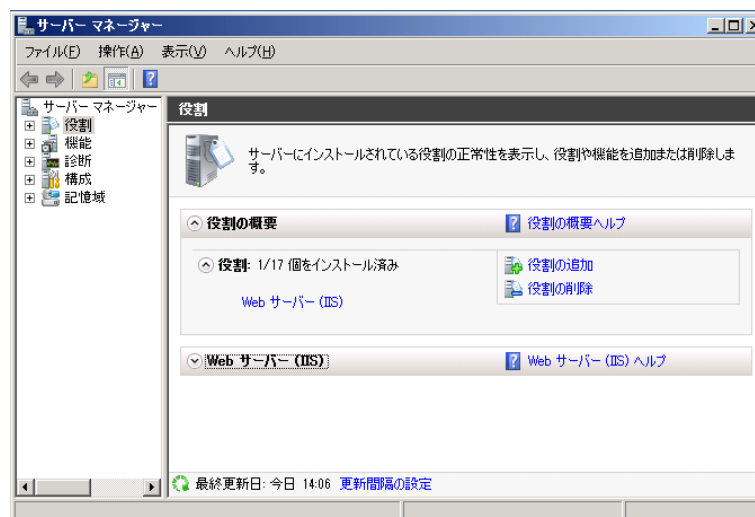


図:サーバー マネージャー

2. [Web サーバー (IIS)] を展開し [役割サービスの追加] をクリックします。

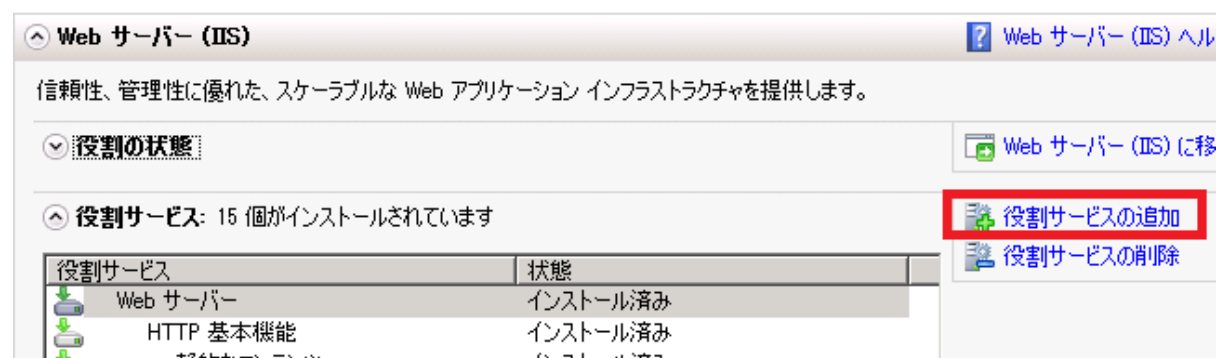


図:役割サービスの追加 メニュー

3. [役割の追加] ウィザードが起動し、[役割サービスの選択]画面が表示されるので、以下の追加役割サービスを選択して [次へ] ボタンをクリックします。(※)

- ASP
- ISAPI 拡張機能

(※) [ASP] を選択すると以下のダイアログ ボックスが表示されるので [必要な役割サービスを追加] ボタンをクリックしてください。その際にその他の必要な役割も自動で選択されます。

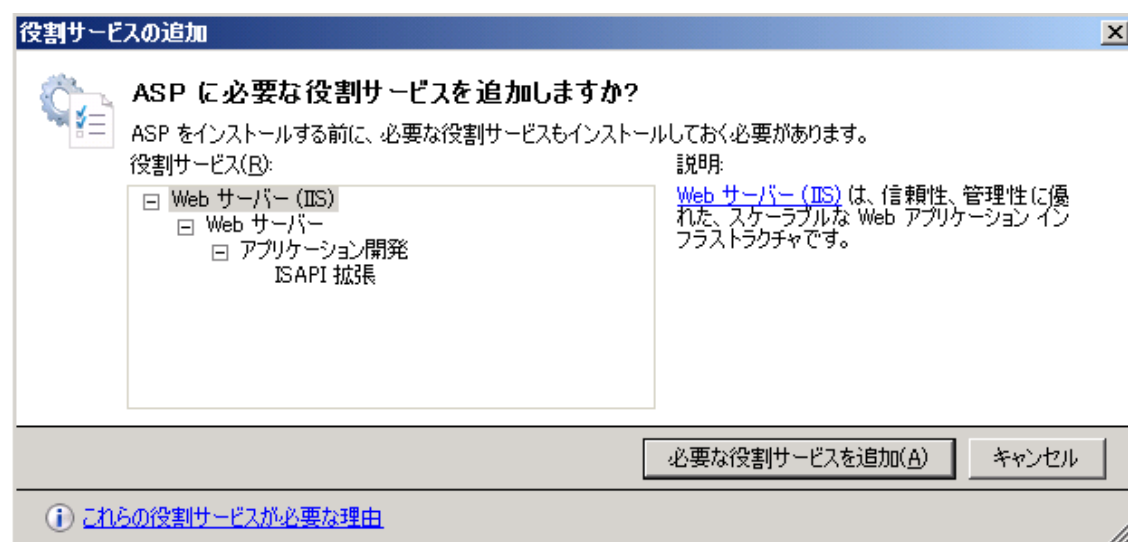


図:役割サービスの追加 確認ダイアログ ボックス

3. [Windows の機能の有効化または無効化] リンクをクリックします。



図: [Windows の機能の有効化または無効化] リンク

4. [Windows の機能] ダイアログ ボックスで、[インターネット インフォメーション サービス] のツリーより以下の追加機能を選択します。(※)

- ASP
- ISAPI 拡張機能

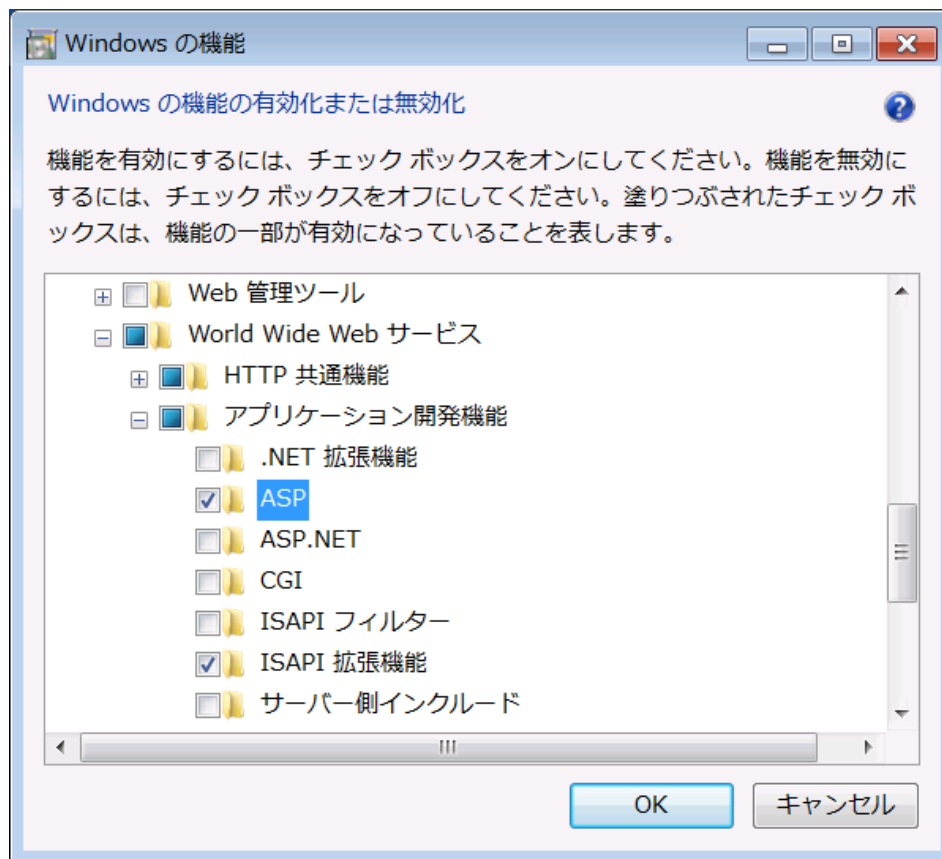


図: Windows の機能 選択ダイアログボックス

(※) [ASP] を選択すると自動的に [ISAPI 拡張機能] も選択されます。

5. [OK] ボタンをクリックするとインストールが行われます。

サンプルの実行

ASP.NET アプリケーションを実行するには、プログラムコードを記述したファイルを**アプリケーション**として公開している仮想ディレクトリに配置する必要があります。

ASP.NET をアプリケーション配置するための具体的な手順は以下のとおりです。

1. エクスプローラーを “管理者として実行” し、C:\inetpub\wwwroot 下に “webApp” という名前でフォルダーを作成します。
2. テキスト エディターに以下のコードを貼り付け、“Hello.asp” という名前で保存します。

```
<html>
<head>
  <title></title>
</head>
<body>
  <form>
    <h3>
      <%= "Hello World." %>
    </h3>
  </form>
</body>
</html>
```

Hello.asp のソースコード

3. 作成した Hello.asp ファイルを c:\inetpub\wwwroot\webApp フォルダーにコピーします。
4. IIS マネージャーを起動し、画面左のツリービューを展開し、“Default Web Site” 下の “webApp” 仮想ディレクトリを選択します。
5. マウスの右ボタンをクリックし、表示されたコンテキスト メニューより [アプリケーションへの変換] をクリックします。

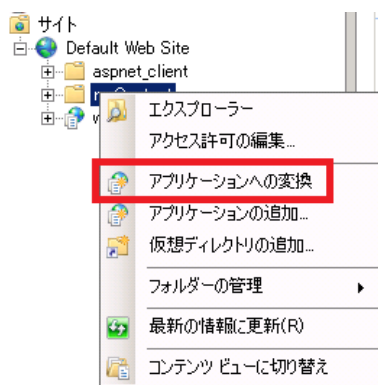


図:アプリケーションへの変換

6. [アプリケーションへの変更] ダイアログボックスが表示されるので既定のまま [OK] ボタンをクリックします。

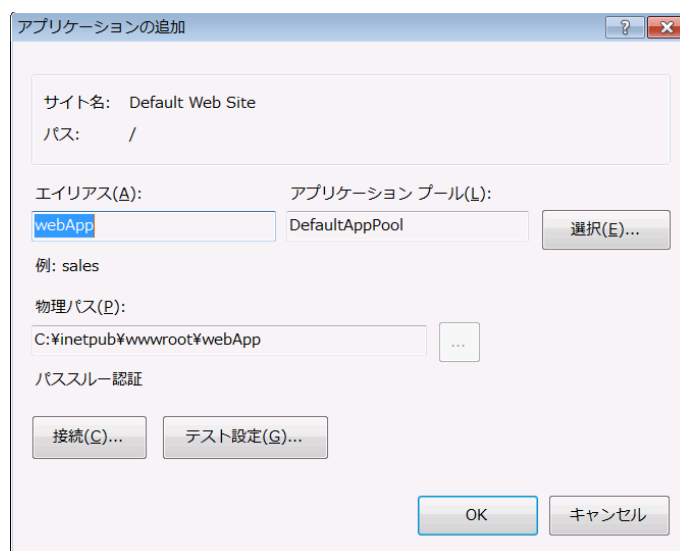


図:アプリケーションの追加 ダイアログボックス

7. ブラウザーを起動し、URL ボックスに以下の URL を入力し、[Enter] キーを押下します。

`http://localhost/webapp/hello.asp`

8. ブラウザーに “Hello World.” と表示されることを確認してください。

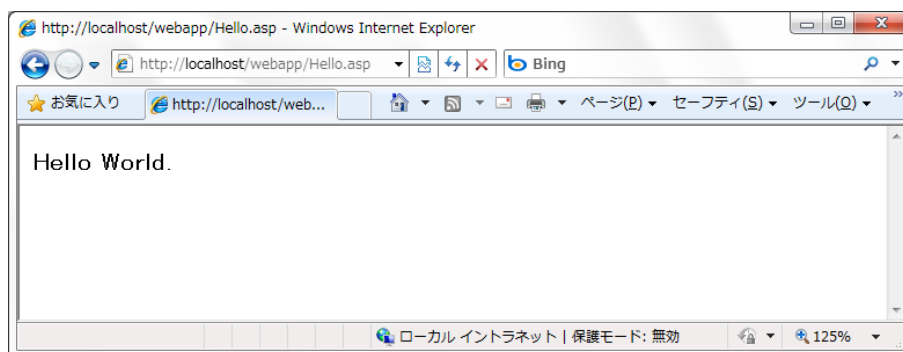


図 : ブラウザーでサンプルを表示

エラーログ

ASP ランタイムは、発生したエラー、警告等を イベント として認識し、その情報を Windows OS のイベントログとして出力します。

ASP.NET が出力するイベントログの内容を確認するには、Windows の **管理ツール** 内にある **イベントビューアー** を使用します。具体的な手順は以下のとおりです。

1. Windows の [スタート] ボタンをクリックし、[コントロールパネル] を選択します。

2. コントロール パネルのウィンドウが表示されるので、ウィンドウ右上にある [表示方法] ドロップダウン リストボックスをクリックし、[大きいアイコン] を選択します。

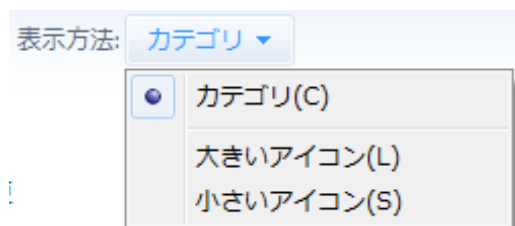


図:コントロール パネルの 表示方法 ドロップダウン リストボックス

3. コントロール パネルのウィンドウから [管理ツール] のアイコンをダブルクリックします。



図:管理ツール アイコン

4. [管理 ツール] のウィンドウが表示されるので、同ウィンドウ内から [イベント ビューアー] のショートカット アイコンをダブルクリックします。

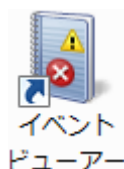


図:イベント ビューアーのショートカット アイコン

5. [イベント ビューアー] のウィンドウが表示されるので、画面左のツリービューを展開し、[Windows ログ] 中の [アプリケーション] を選択します。さらに画面左の [操作] パネルから [現在のログをフィルター] リンクをクリックします。

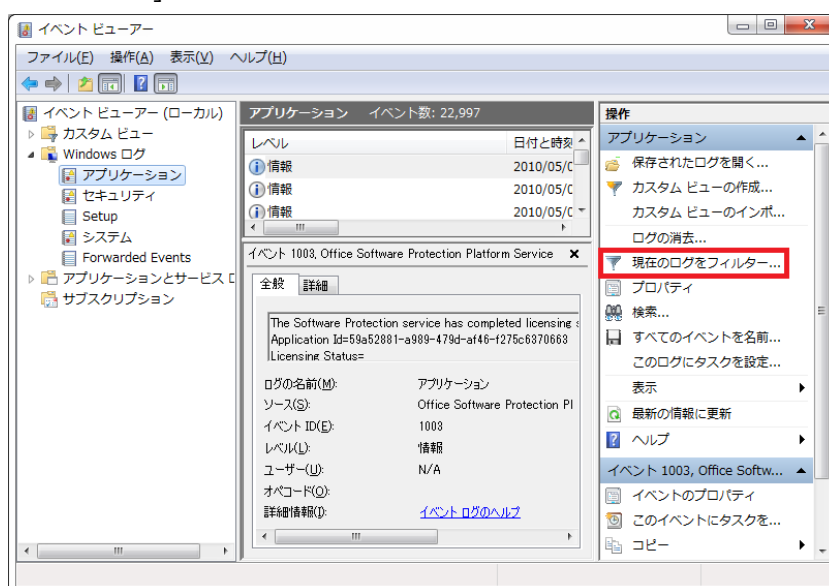


図:イベント ビューアーの画面

6. [現在のログをフィルター] ダイアログ ボックスが表示されるので、[イベント ソース] ドロップダウン リストボックスのリスト中から “Active Server Pages” を選択し [OK] ボタンをクリックします。

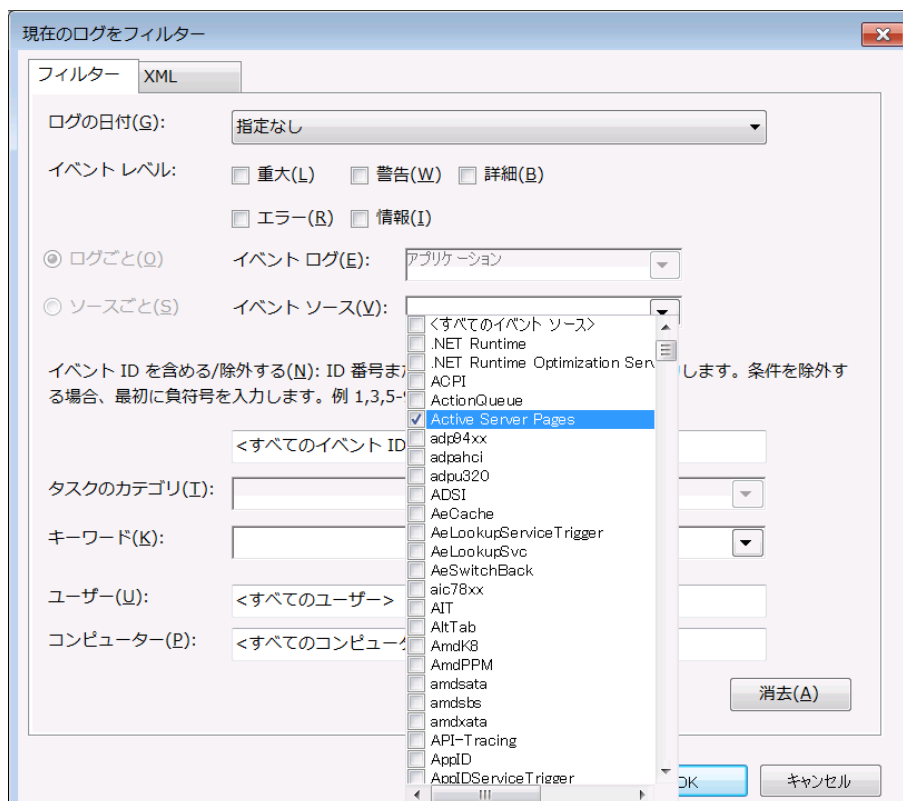


図: イベントログのフィルター設定画面

イベント ログ リストに、ASP に関するイベント ログのみが抽出されるので、任意のイベント ログをダブルクリックすると内容を確認することができます。

ASP の実行環境についてはインストール直後、ほとんどの機能が有効になっていないので以前の IIS を利用なさってきた方は戸惑うかもしれません。エラーの行番号表示を含む詳細表示などをオンにするには IIS マネージャー で ASP アイコンをダブルクリックします。

デバッグ プロパティ	
COM コンポーネント例外をキャッチする	True
On End 関数を匿名で実行する	True
エラーを NT ログに記録する	False
エラー要求のログ記録を有効にする	True
クライアント側のデバッグを有効にする	False
サーバー側のデバッグを有効にする	False
スクリプト エラー メッセージ	An error occurred on the s
ブラウザーへのエラー送信	False
行番号の計算	True

図 : ASP 環境のプロパティよりデバッグ設定

これ以外の詳細な設定については下記の TechNet IIS TechCenter の記事を参照ください。

IIS 7.0 および IIS 7.5 上で Classic ASP アプリケーションを実行する

<http://technet.microsoft.com/ja-jp/library/ee155445.aspx>

PHP

PHP (Hypertext Preprocessor) は、オープンソース コミュニティによって開発、保守されている、Web ページを動的に生成するための言語、およびその言語で記述されたプログラムを動作されるためのランタイムを指します。

PHP 言語で記述された Web アプリケーションのことを一般的に **PHP アプリケーション**と呼びます。

PHP は、インターネットでもっとも多く使用されている Web アプリケーション開発言語であり、また、様々なコミュニティから数多くのオープンソース アプリケーションが提供されていることも魅力のひとつとなっています。

IIS 7 は FastCGI モジュールを使用することで、PHP アプリケーションを安全で快適にホストすることができます。

FastCGI と PHP ランタイムと組み合わせることで、PHP で記述された豊富なアプリケーションのプラットフォームとして IIS 7 を使用することができます。

環境の設定方法

IIS 7 で PHP をホストするには以下の 2 つが必要となります。

1. FastCGI モジュール
2. PHP ランタイム

FastCGI は、あらかじめ IIS 7 の機能として同梱されているため、別途モジュールを入手する必要はなく **サーバー マネージャー** もしくは **Windows の機能の有効化または無効化** から IIS の機能追加画面で [CGI] を選択することで従来の CGI モジュールとともにインストールされます。

PHP ランタイムは PHP コミュニティページ等から任意のバージョンのランタイムを入手して使用することができます。PHP のランタイムには、スレッドとしての動作を考慮したタイプと、スレッドとしての動作を考慮していない Non Threading タイプがあり、後者のほうがスレッドの同期等に関する処理がないため高速に動作します。FastCGI 上での PHP はプロセスモデルでの動作となるため、Non Threading タイプの PHP ランタイムを使用することができます。

このあたりの原理については下記の記事を参照ください。

無償冊子 IT 交差点 4 号 PHP on IIS 記事 (Silverlight ビューアー) p.16-17

<http://www.microsoft.com/japan/powerpro/magazine/viewer/vol4/>

Windows でインターネット Web サーバー 第 2 回 Windows でも PHP

http://www.atmarkit.co.jp/fwin2k/verification/web200802/web200802_01.html

FastCGI モジュールのインストールと、PHP ランタイムの配置、およびそれらを組み合わせて PHP アプリケーションを動作されるための具体的な手順は以下の通りです。

FastCGI のインストール

FastCGI は、IIS 7 のセットアップ画面で [CGI] の機能を追加することでインストールされます。手順については、Windows Server 2008 R2 と Windows 7 では若干の違いがあります。

Windows Server 2008 R2 の場合

1. [スタート] ボタンをクリックし、[管理ツール] - [サーバー マネージャー] を選択してサーバー マネージャーを起動します。



図:サーバー マネージャー

2. 画面左のツリービューで [役割] を選択し、画面右の [Web サーバー (IIS)] パネルで [役割サービスの追加] リンクをクリックします。

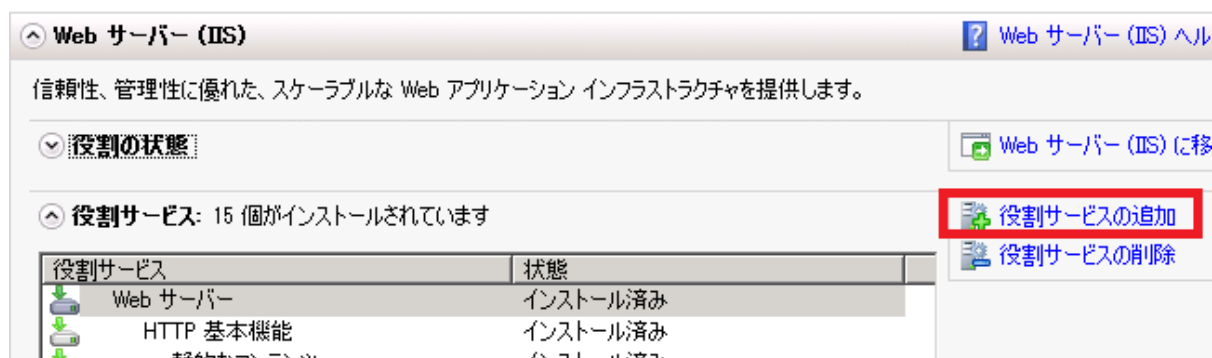


図:役割サービスの追加 メニュー

3. [役割サービスの追加] 画面が表示されるので、[役割サービス] リスト内の “CGI” チェックボックスにチェックを入れ、[次へ] ボタンをクリックします。

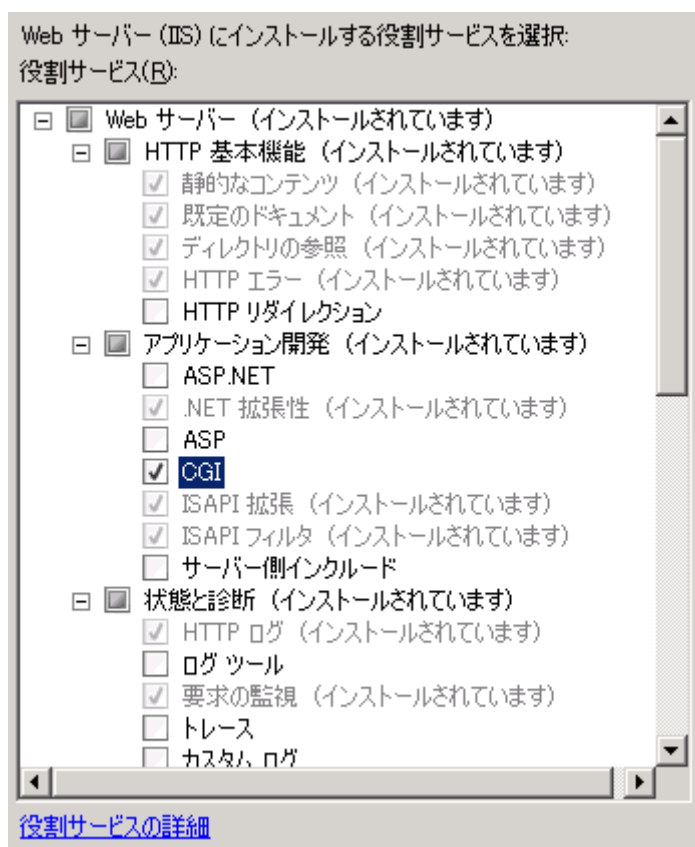


図:役割サービスの追加画面

4. [インストール オプションの確認] 画面が表示されるので [インストール] ボタンをクリックします。

5. インストールの進行状況が表示され、その後、インストールの完了画面が表示されるので [閉じる] ボタンをクリックします。

以上で Windows Server 2008 R2 上の IIS 7 への CGI の役割追加は完了です。

Windows 7 の場合

1. [スタート] ボタンをクリックし、[コントロール パネル] をクリックします。
2. [コントロール パネル] 内の [プログラム] リンクをクリックします。

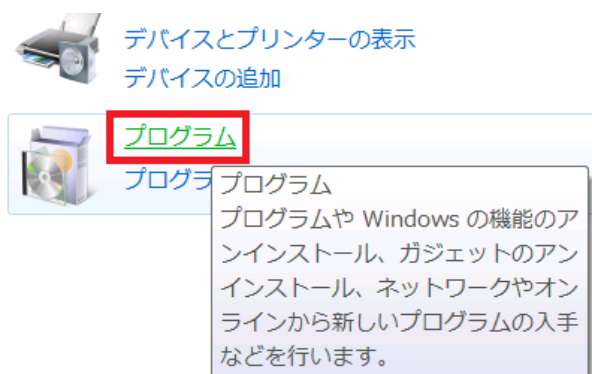


図:コントロール パネルの [プログラム] リンク

3. [Windows の機能の有効化または無効化] リンクをクリックします。



図: [Windows の機能の有効化または無効化] リンク

4. [Windows の機能] ダイアログ ボックスで、[インターネット インフォメーション サービス] のツリーより [CGI] を選択します。

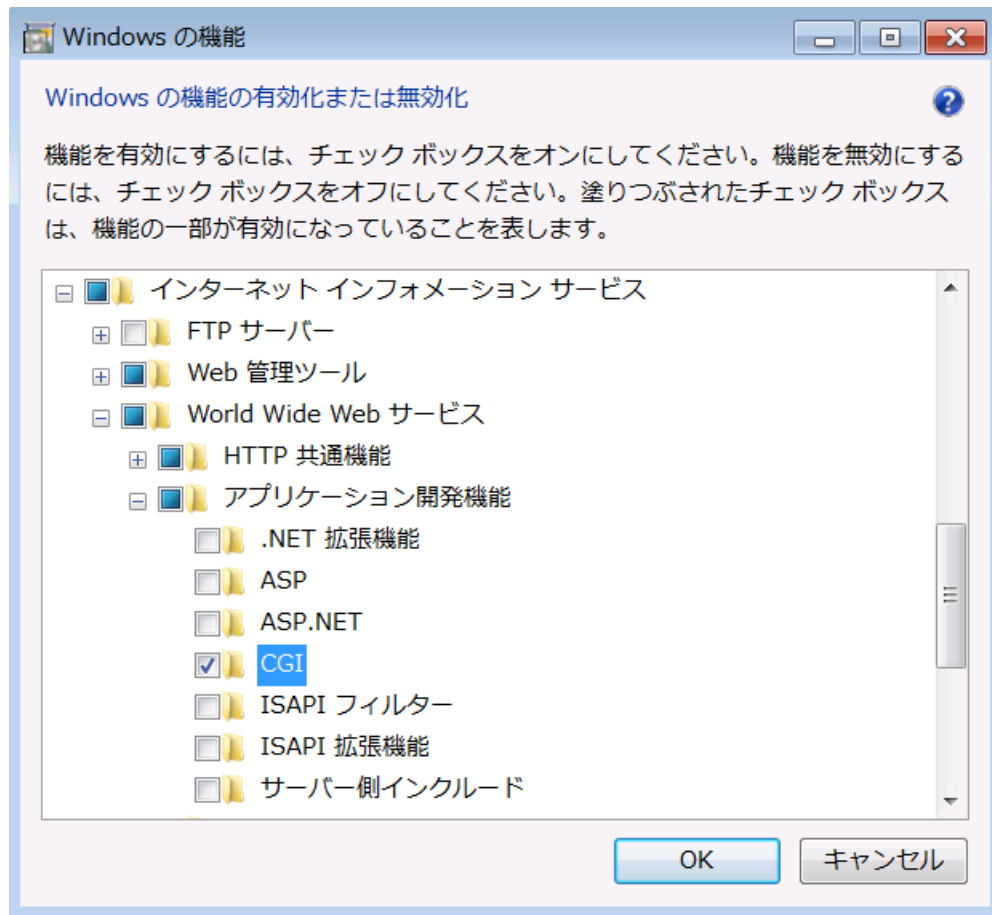


図:Windows の機能 選択ダイアログ ボックス

5. [OK] をクリックするとインストールが行われます。

PHP ランタイムの入手と配置

PHP ランタイムの入手と配置作業については、Windows Server 2008 R2 、 Windows 7 とともに共通です。

1. PHP のサイト (<http://windows.php.net>) より PHP のランタイムをダウンロードし、以下のフォルダーに内容を解凍します。※フォルダーが存在しない場合は作成してください。

C:\php5

2. 同フォルダーに対し、**インターネット ゲスト アカウント**(IIS_IUSRS) に対して読み取りの権限を付与します。

以上で PHP ランタイムの配置は完了です。

なお、ホストするアプリケーションによっては **php.ini** ファイルの作成、設定などが必要になる場合があります。その場合はアプリケーション添付のドキュメントに従い php.ini ファイルへの作業を行ってください。

PHP アプリケーションの実行環境の設定

1. エクスプローラーを [管理者として実行] で起動し、パス C:\inetpub\wwwroot の直下に “php” という名前でフォルダーを作成します。

作成するフォルダーのパス C:\inetpub\wwwroot\php

2. [スタート] ボタンをクリックし、[管理ツール] - [インターネット インフォメーション サービス (IIS) マネージャー] を選択して起動します。
3. 画面左のツリービューでコンピューター名をクリックし、[(コンピューター)] - [サイト] - [Default Web Site] を展開し、仮想ディレクトリ php を選択します。
4. 画面中央の パネルが [機能ビュー] となっていることを確認し、同パネルの [ハンドラー マッピング] のアイコンをダブルクリックします。

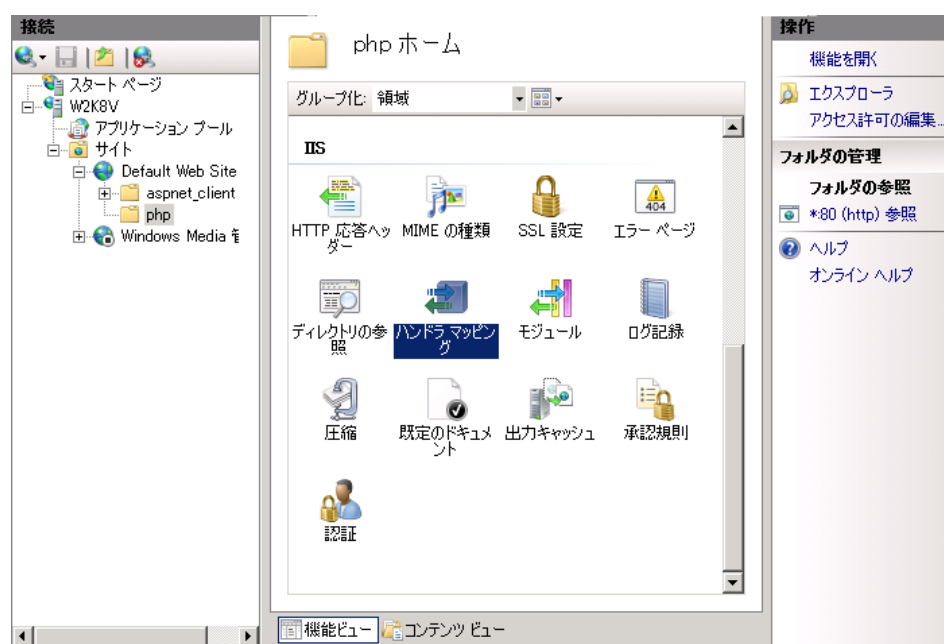


図:IIS マネージャー [機能ビュー]

5. [ハンドラー マッピング] の画面が表示されるので、画面右の [操作] パネルより [モジュール マップの追加...] リンクをクリックします。

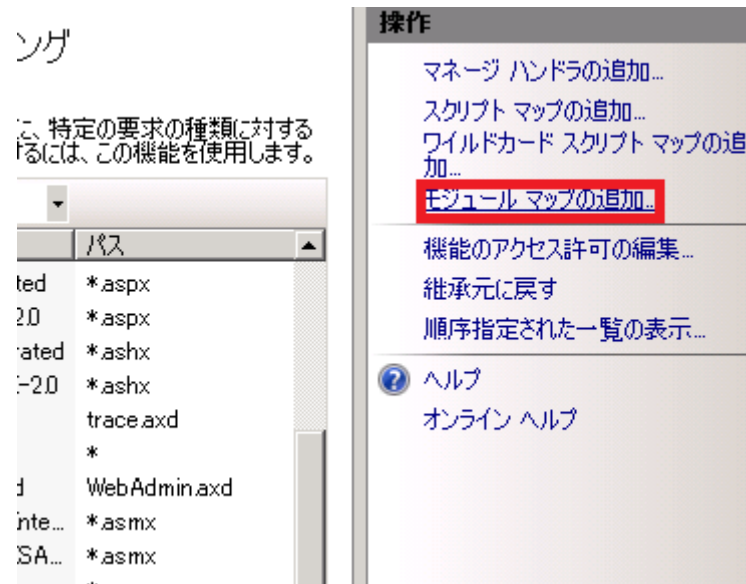


図: ハンドラー マッピング [モジュール マップの追加] メニュー

6. [モジュール マップの追加] ダイアログ ボックスが表示されるので、各項目を以下のように設定し [OK] ボタンをクリック

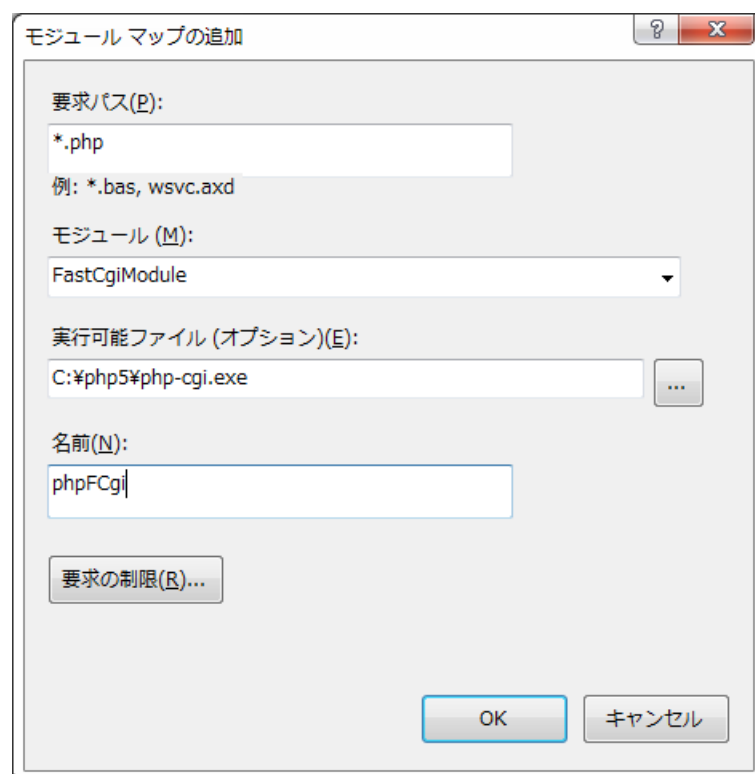


図: モジュール マップの追加ダイアログ ボックス

以上で、IIS 7 で PHP アプリケーションを動作させるための設定は完了です。作成したフォルダーに有効な php アプリケーションを配置してホストすることができるようになります。

サンプルの実行

IIS 7 で PHP アプリケーションを動作させるには FastCGI モジュールと PHP ランタイムを使用
して正しくハンドラー マッピングがされた仮想ディレクトリにアプリケーションとして有効な

*.php ファイルを配置するだけです。

PHP アプリケーションが実行可能であるかどうかは、以下の手順で確認することができます。

1. テキストエディタを [管理者として実行] し、以下のコードを貼り付けて info.php という名前で保存します。

```
<?php phpinfo(); ?>
```

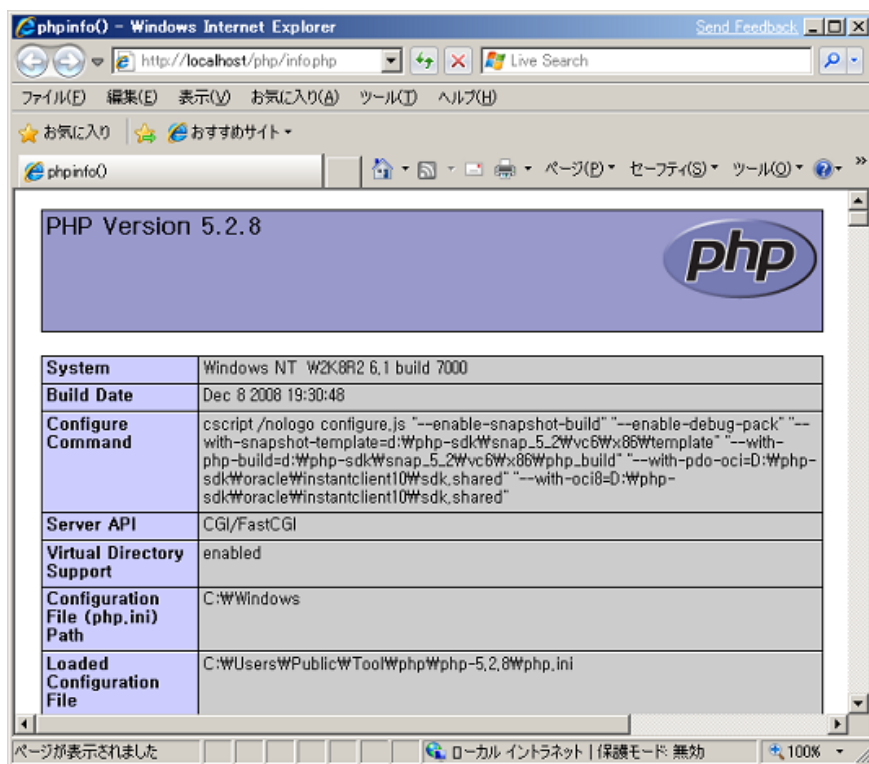
2. 作成した info.php ファイルを前出の手順でハンドラー マッピングの設定を行った以下のフォルダーにコピーします。

C:\inetpub\wwwroot\php

3. Web ブラウザーを起動し、以下の URL にアクセスします。

<http://localhost/php/info.php>

以下のような PHP ランタイムの情報がブラウザーに表示されることを確認してください。



図：phpinfo の表示を確認

エラーログ

PHP アプリケーションのエラーログを取得するには、php.ini ファイルの error_log ディレクティブで、エラーを出力するためのファイルを指定しておく必要があります。

例) PHP アプリケーションのエラーログを、パス C:\webApplogs 下に php_error.log と名前のファイルに出力するように指定

```
log_errors = On  
error_log = "C:\WebApplogs\php_error.log"
```

また、FastCGI ハンドラーや PHP ランタイムのエラーに関しては、エラーの情報をイベント ログとして出力する場合もありますので、適宜、イベント ビューアーを使用してイベント ログを確認してください。その場合、“PHP”、“FastCGI” をキーワードとして、イベント ログ リストを検索すると、目的のイベント ログを探しやすくなります。

その他 設定周りに関する情報は？

PHP 5.3 から複数のバージョンの PHP を動かす設定が非常に簡単になっていますが、このような設定関係の情報は以下を参照ください。

IIS での PHP アプリケーションの実行

<http://technet.microsoft.com/ja-jp/library/ee155446.aspx>

Windows Maniax サイト

<http://www.windows-maniac.com/>

Multiple PHP versions on the same IIS server (英語)

<http://ruslany.net/2009/12/multiple-php-versions-on-the-same-iis-server/>

SSI を使えるようにするには

SSI (Server Side Includes) は、Web ページに、動的にコンテンツを挿入するための仕組みの一つです。

SSI は HTML に埋め込まれた SSI のコマンドをサーバー側で処理し、その結果をクライアントに返すことで、動的にページ内の一部コンテンツを更新します。

SSI が実行可能なコマンドは、あくまでも IIS 上の SSI モジュールが提供しているコマンドのみであり、ASP.NET や PHP といった Web アプリケーションのように、開発者が定義した独自のロジックを実行することはできません。

しかしながら、複数のコマンドを組み合わせ、簡単なロジックを実装することが可能です。また、Web アプリケーションを使用して同様の機能作成するよりも、少ない工数で実装することが可能です。

SSI の主な用途は以下の通りです。

- ・ コンテンツの更新日時、サイズの表示
- ・ 複数のページに渡り共通したヘッダーやフッターの表示
- ・ 外部コマンドの実行結果の表示
- ・ カウンターなど

IIS 7 の SSI 機能の有効化

IIS 7 で SSI を使用するには、IIS の機能追加画面で サーバー側インクルード を追加します。具体的な手順は以下の通りです。

Windows Server 2008 R2 の場合

1. [スタート] ボタンをクリックし、[管理ツール] - [サーバー マネージャー] を選択してサーバー マネージャーを起動します。



図:サーバー マネージャー

- 画面左のツリービューで [役割] を選択し、画面右の [Web サーバー (IIS)] パネルで [役割サービスの追加] リンクをクリックします。

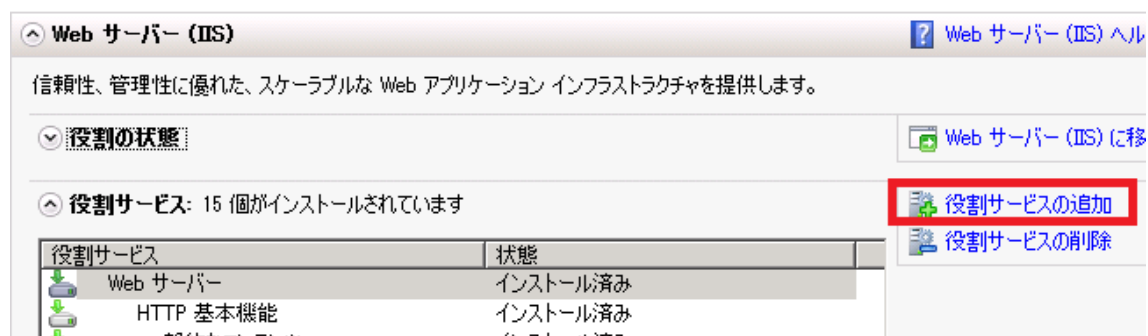


図:役割サービスの追加 メニュー

- [役割サービスの追加] 画面が表示されるので、[役割サービス] リスト内の “サーバー側インクルード” にチェックボックスにチェックを入れ、[次へ] ボタンをクリックします。

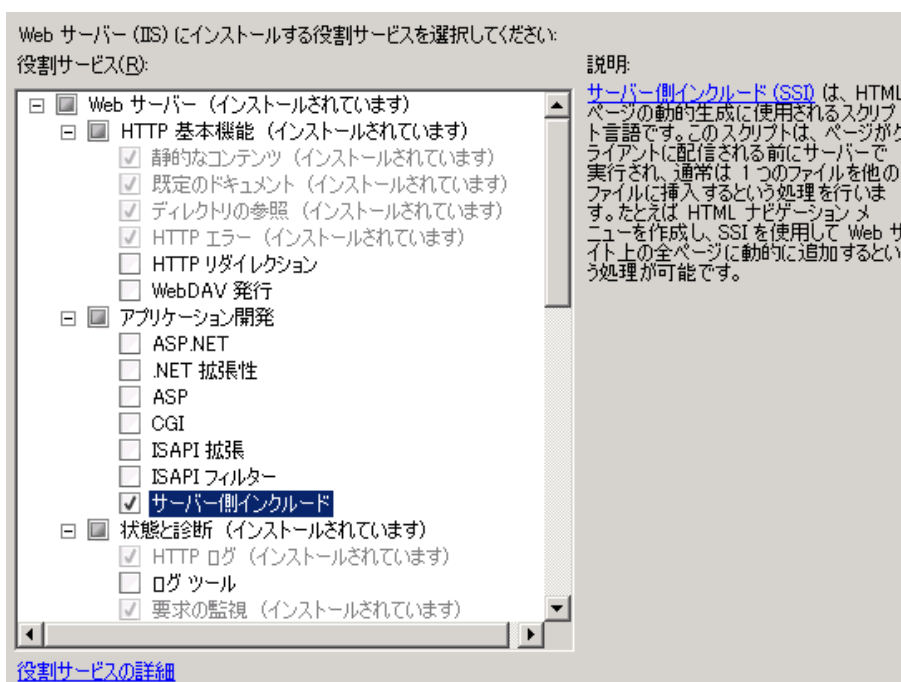


図:役割サービスの追加画面

- [インストール オプションの確認] 画面が表示されるので [インストール] ボタンをクリックします。
- インストールの進行状況が表示され、その後、インストールの完了画面が表示されるので [閉じる] ボタンをクリックします。

以上で、Windows Server 2008 R2 上の IIS 7 への SSI のインストールは完了です。

適切な SSI コマンドを記述した、拡張子 **shtm**, **shtml** のファイルを仮想ディレクトリに配置して実行することができます。

Windows 7 の場合

1. [スタート] ボタンをクリックし、[コントロール パネル] をクリックします。
2. [コントロール パネル] 内の [プログラム] リンクをクリックします。

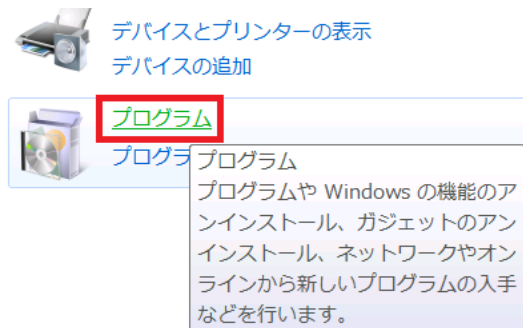


図: コントロール パネルの [プログラム] リンク

3. [Windows の機能の有効化または無効化] リンクをクリックします。



図: [Windows の機能の有効化または無効化] リンク

4. [Windows の機能] ダイアログ ボックスで、[インターネット インフォメーション サービス] のツリーより [サーバー側インクルード] を選択します。

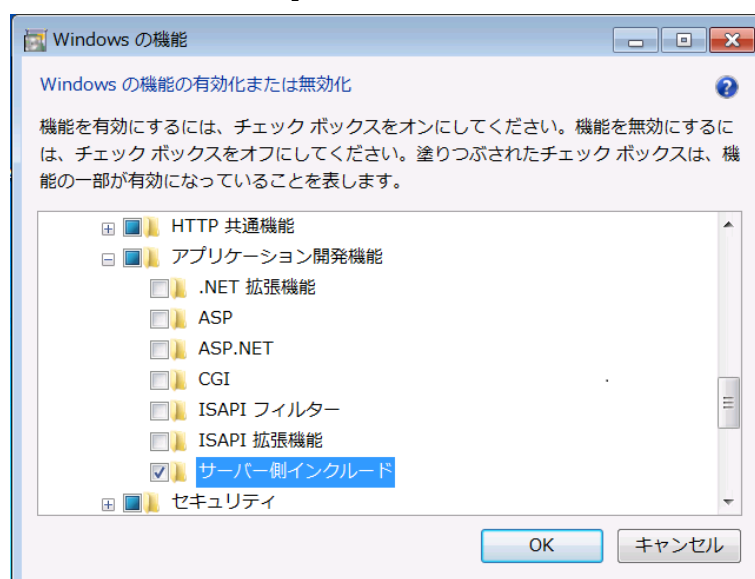


図: サーバー側インクルード の選択画面

5. [OK] をクリックするとインストールが行われます。

以上で、Windows 7 上の IIS 7 への SSI のインストールは完了です。

適切な SSI コマンドを記述した、拡張子 **shtm**, **shtml** のファイルを仮想ディレクトリに配置して実行することができます。

SSI コマンドの実行

Web サーバーで SSI コマンドを実行させるには、以下のように HTML にコメントとして SSI のコマンドを埋め込みます。

```
<html>
<head></head>
<body>
<!--#config timefmt="%y 年%m 月%d 日の%H 時%M 分(%p)" -->
現在の日時は、<!--#echo var="DATE_LOCAL" -->です。
</body>
</html>
```

サンプル：現在の日時を SSI を使用して表示

この SSI のコマンドを記述している部分を**ディレクティブ**と呼びます。ディレクティブの内容は、Web ブラウザーではコメント扱いとなりますので、SSI をサポートしていないサーバーに SSI コンテンツファイル配置しても、ディレクティブの内容が表示されてしまうということはありません。(ただし、HTML ソースにはそのまま表示されます)

SSI のコマンドを含めた HTML を IIS 7 で処理させるには、ファイルの拡張子を shtm もしくは shtml として、仮想ディレクトリに配置するだけです。

サーバー側インクルードの機能がインストールされた IIS 7 では、*.shtm、*.shtml のファイルに対し、SSI のハンドラーが動作するように構成されています。

SSINC-shtm	*.shtm	有効	ファイル	ServerSideIncludeModule	継承
SSINC-shtml	*.shtml	有効	ファイル	ServerSideIncludeModule	継承

図:IIS 7 ハンドラー マッピングの SSI の設定

たとえば、以下の サンプル HTML を test.shtml という名前で任意の仮想ディレクトリに配置し、Web ブラウザーからアクセスすると、ブラウザー上に User Agent の情報が表示されます。

```
<html>
<head></head>
```

```
<body>
使用しているブラウザの情報<br>
<!--#echo var="HTTP_USER_AGENT" -->
</body>
</html>
```

サンプル：ブラウザの User-Agent の情報を表示

IIS 7 の SSI に用意されているコマンド

IIS 7 に用意されている SSI ハンドラーは一般的な SSI コマンドをサポートしていますが、他の Web サーバーで提供されている SSI と処理内容が異なる可能性もありますので (※)、移行の際は十分な検証を行ってください。

コマンド	処理内容
echo	変数の内容を出力
If,else	条件判断を行う
fsize	指定したファイルのサイズを取得
flastmod	指定したファイルの最終更新日時を取得
include	他のファイルの内容を差し込む
exec	他のプログラムを実行し、出力結果を差し込む
set	変数の設定を行う
config	SSI コマンドのさまざまな設定を行う

表：IIS 7 の SSI モジュールがサポートしている標準的なコマンド

(※) たとえば Apache 1.2 以上が提供している printenv コマンドは提供されていません。また、IIS 7 では、exec コマンドに cmd パラメータを利用することはできなくなっています。

コラム： IIS 7 での Web アプリケーションの開発環境

IIS 7 が標準でサポートしている Web アプリケーションの実行環境は、ASP と ASP.NET ですが、ASP の実行環境は過去に作成された Web アプリケーションを保守する目的で残されています。そのため ASP.NET のリリース以降、ASP の機能拡張は行われず、専用の開発環境は提供されていません。

現在の IIS での Web アプリケーション開発は ASP.NET で行うことが多く行われています。

IIS は、ASP.NET のアプリケーションサーバーとしても設計されているため、ASP.NET を使用して Web アプリケーションを開発することで、IIS の提供する機能を余すところなく利用することができます。また、IIS 7 からは、ASP.NET との機能統合が進み、ASP.NET を使用して IIS 自身の拡張機能を作成することも可能になっています。

ASP.NET で Web アプリケーションを開発するには

ASP.NET アプリケーションの開発には Microsoft Visual Studio を使用します。Microsoft Visual Studio は量販店などで購入できますが、ホビーユーザー向けに 無償の Express Edition も用意されています。

Microsoft Visual Studio Express

<http://www.microsoft.com/japan/msdn/vstudio/express/>

Microsoft Visual Studio には開発専用の Web サーバーが用意されているため IIS がインストールされていない環境でも ASP.NET アプリケーションを開発することができます。

ASP.NET の実際の開発についても豊富なドキュメント用意されています。

Microsoft Web 開発 ガイドライン ～ ASP.NET プログラミング エssenシャル ～

<http://msdn.microsoft.com/ja-jp/asp.net/ff602016.aspx>

「300 秒でズバリ !!」 & 「10 行でズバリ !!」 シリーズ ～ ASP.NET 編 ～

<http://msdn.microsoft.com/ja-jp/events/4218e2ab-ccda-48d6-b2d2-d772ca739aba.aspx>

Code Recipe -.NET 開発サンプル コード集

<http://msdn.microsoft.com/ja-jp/samplecode.recipe.aspx>

Windows 上で PHP アプリケーションを開発するには

Microsoft Expression Web には PHP コードを記述するための IDE が用意されており、インテリセンスや、スクリプトブロックの挿入などの強力な入力支援機能を使用して PHP コードの記述にかかる労力を軽減することができます。

Microsoft Expression Web 3 – PHP サイトの作業

[http://msdn.microsoft.com/ja-jp/library/cc294979\(v=Expression.30\).aspx](http://msdn.microsoft.com/ja-jp/library/cc294979(v=Expression.30).aspx)

テキスト エディターや、Eclipse など使い慣れたツールを使用して開発を行うことも可能です。

このドキュメントでは、Windows Server 2008 R2、Windows 7 に搭載されている IIS 7 で PHP アプリケーションを動作させる方法についてのみ解説していますが、Windows XP (IIS 5.1) や Windows Server 2003 (IIS 6.0) でも別途 FastCGI をインストールすることで PHP アプリケーションを効率よく安全に動作させることができます。

FastCGI 1.5 for IIS 6

<http://technet.microsoft.com/ja-jp/iis/ee839445.aspx>

複数の環境で検証を行うには

マイクロソフトでは、無償で使用可能な仮想化プラットフォームを提供しており、それらを利用して仮想化した OS を複数用意し、テストを行うことができます。また、各仮想プラットフォームは OS の状態を保存するスナップショット機能をサポートしていますので、設定のリセット作業も軽減されます。

Microsoft Virtual PC と Windows Virtual PC (Windows 7 用)

<http://www.microsoft.com/japan/windows/virtual-pc/default.aspx>

Windows Server 2008/R2 付属の Hyper-V

<http://www.microsoft.com/japan/virtualization/default.mspx>

Microsoft Hyper-V Server (Server Core ベースの OS 込み無償版)

<http://www.microsoft.com/japan/servers/hyper-v-server/default.mspx>

Virtual PC 系と Hyper-V のそれぞれの仮想化プラットフォームは、特性的に大きいさまざまな違いがあるので、実際に使用する際は、URL 先のドキュメントを熟読してから作業を行ってください。