



PHP on Windows ガイドライン

【ドラフト版】

～ 第 5 章: ベスト プラクティス

※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 2011/06

日本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願います。

第5章 ベスト プラクティス

ここまでの章で PHP を Windows 上で動かす際の様々な要素を整理してきました。この章ではもう少し細かいレベルで Windows 上で PHP 環境を構築する場合に注目すべき点を整理することにしました。主として以下の記事を翻訳していますが、よりわかりやすく分類と加筆を行っています。

Best Practices for PHP on the Microsoft Web Platform

<http://learn.iis.net/page.aspx/728/best-practices-for-php-on-the-microsoft-web-platform/>

はじめに

本章では Windows Server 2008/R2 上の IIS7 において FastCGI 構成で PHP を動作させる上で注意すべき点、知っておくと役立つ点を具体的に内容によって分類して解説していきます。

IIS や FastCGI 関連の設定は IIS7 の GUI 管理ツールである IIS マネージャーで設定できるほか、Appcmd.exe を使用したコマンド、Windows PowerShell や VBScript で書かれたスクリプトでも設定が可能です。あるいは applicationHost.config ファイル (Apache での httpd.conf 相当) の直接編集でも設定することができます。

この章に登場する IIS7 拡張モジュールに関する参照ページは最新のもの参照するために本家の IIS.NET ページを指すことにします。日本語の解説について本家よりも少し遅れている箇所もありますが、以下のページにあります。

IIS 7 - 拡張機能 ダウンロード ページ

<http://technet.microsoft.com/ja-jp/iis/dd742256>

OS レベルの設定についても IIS 関連の設定の後に実際の手順を交えて解説をします。基本的に不要なサービスを停止する点だけの指摘ですが、合わせて今後の Windows 管理における Windows PowerShell に関する話なども盛り込んでいます。

IIS 関連の設定

それではまずは IIS に直接関連する事項を整理していくことにします。

IIS 6.0 および 5.1 での制限事項

これまでの章で説明してきた FastCGI を使用した PHP 環境の構築は IIS5.1 や IIS6 でも可能で、これらのバージョン用の FastCGI モジュールもマイクロソフトから提供をしております。

FastCGI for IIS (英語)

<http://www.iis.net/download/fastcgi>

しかし、以前のバージョンでは多くの PHP アプリケーションで利用されるであろう URL 書き換えモジュール (URL Rewrite) についてマイクロソフト提供のものが存在しません。このモジュールの主な利用方法は URL を操作する、あるいは 特定のファイルやフォルダーへのアクセス制限することです。以前の IIS では標準では利用できないために、mod_rewrite に似た設定をコード記述なしで行うにはサード パーティ製の IIS 用のモジュール製品を利用するのが早道です。代表的な例は下記のような製品です。

Helicon Tech 社製品 (英語)

<http://www.helicontech.com/>

メモ : 多くのサード パーティ製品がありますが、それぞれの実装になっており、マイクロソフトではソースコードを持っていないですからこれらに直接関係する製品の問い合わせは受け付けることができませんので、製品サポートは切り分けのサポートまでになりますのでご注意ください。

サーバー パフォーマンスの最適化

サーバーのパフォーマンス最適化の大きなポイントの一つは不要なものをとにかく入れないこと、動かさないことです。その観点でいくつかのポイントを取り上げます。

IIS7 のモジュール構造を活かす

以前の IIS はほぼすべての機能が一つの大きなプログラムとなっており、使用しない機能もメモリー上にロードされるケースが多くあったのに比べ、IIS7 ではモジュール化を実現したので、「有効にしない」だけでなく、インストールしないことも容易に構成可能になりました。クライアント OS で

はコントロール パネルから、サーバー OS ではサーバー マネージャーから必要なものだけをインストールするようにしましょう。下記の記事一覧が参考になるはずですよ。

IIS7.0 のインストール

<http://technet.microsoft.com/ja-jp/library/dd647600.aspx>

IIS ログを最小化

必要であれば Web サイト、仮想ディレクトリ、ファイルやフォルダーのログを最低限の出力に設定しましょう。ログの処理がオンになっていると、サーバーの Web サイトに関する詳細をテキストファイルに書き出す処理が継続して行われます。しかし、負荷や帯域のための検討をしている場合には必要でしょう。この場合でも特定期間のみオンにし、不要な時にはオフにする運用を考えましょう。

GUI ツールによる設定方法

ログのオン・オフの切り替えを IIS マネージャーで設定する方法を説明します。

1. [スタート] - [管理ツール] から IIS マネージャーを開きます。

スタート メニューの検索ボックスで inetmgr と入力して Enter キーを押すことでも可能です。

2. 画面左の階層からログ設定を行いたいサーバーあるいはサイトをクリックして選択します。
3. 「ログ記録」 の設定をダブル クリックします。
4. 無効にする場合には、操作ペインの [無効にする] をクリックします。
5. 操作ペインの [適用] をクリックして確定させます。

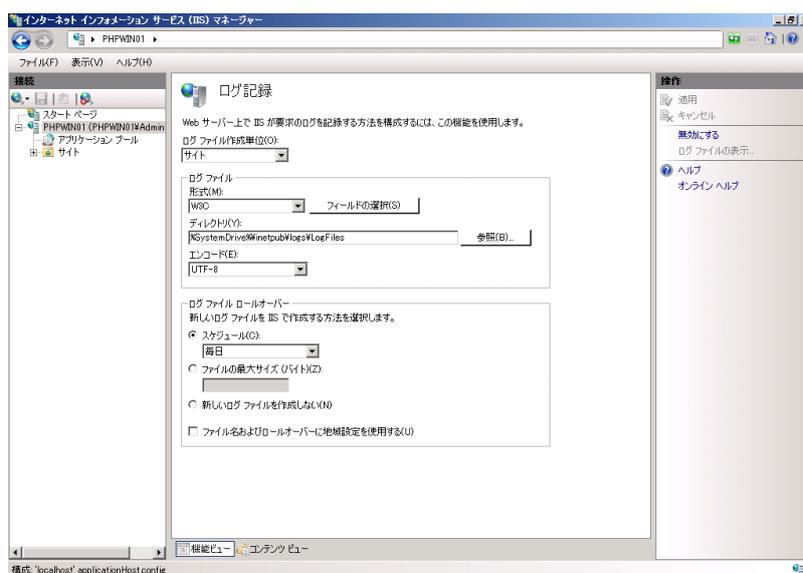


図 1 : ログ記録の設定

コマンドラインからの実行

コマンドラインから実行する場合には `appcmd.exe` コマンドを使用できます。

同様の設定を行うには以下のコマンドを実行します。

```
C:\¥windows¥system32¥inetsrv>appcmd set config /section:httpLogging /dontLog:True
```

PHP 実行環境

これまでの章でも手順などで取り上げたものもありますが、あらためて PHP を IIS 上で動作するように設定する場合の注意点になります。

ノン スレッド セーフ版の PHP を利用

IIS7 の FastCGI モジュールと一緒に利用する際のベスト プラクティスは Windows 用のバイナリーで 5.2.1 以降存在するノン スレッド セーフのモジュールを使うことです。詳細の解説は以前の章を参照ください。

ノン スレッド セーフ版のランタイムはスレッド管理処理を含まない分、当然パフォーマンスがよいことになります。また、最近のランタイムではより新しい Visual Studio でコンパイルしたのも一緒に提供されていますので、VC9 と書いてある方を使いましょう。

PHP 実行プロセスのリサイクル設定

IIS7 で PHP を設定する場合には PHP ネイティブでのリサイクル タイミングよりも FastCGI による `php-cgi.exe` プロセスのリサイクルが早く起こるように設定しましょう。

FastCGI プロセスのリサイクル動作は「インスタンスの `MaxRequests`」(`instanceMaxRequests`) プロパティで構成します。このプロパティでは FastCGI プロセスをリサイクルするまでいくつのリクエストを処理するかを指定することができます。

PHP 自身にも同様のメカニズムがあり、`PHP_FCGI_MAX_REQUESTS` 環境変数で制御されます。`instanceMaxRequests` を `PHP_FCGI_MAX_REQUESTS` 以下に設定することで IIS が直接コントロールしている FastCGI のリサイクルの方が確実に動作するように設定できます。

以下でこの FastCGI 設定を IIS マネージャーと `Appcmd.exe` によるコマンド実行の2つの方法を説明します。

メモ： Windows 7 や Windows Server 2008 R2 から標準で IIS マネージャーに FastCGI 設定アイコンが搭載されるようになりました。この関係で、Windows Vista や Windows Server 2008 ではこのアイコンは存在しません。これを追加するには `Administration Pack` をインストールする必

必要があります。CGI をインストールしている場合だけアイコンが表示されますので環境が準備できているにも関わらず表示されない場合は IIS7 機能のインストール状況を確認ください。

IIS7.0 環境用 Administration Pack

<http://technet.microsoft.com/ja-jp/iis/ee834915>

GUI ツールによる設定方法

インスタンスの MaxRequests を IIS マネージャーで設定する方法を説明します。

1. [スタート] - [管理ツール] から IIS マネージャーを開きます。

スタート メニューの検索ボックスで inetmgr と入力して Enter キーを押すことでも可能です。

2. 画面左の階層からサーバー名をクリックして選択します。
3. [FastCGI の設定] をダブル クリックし、構成したい FastCGI アプリケーションをクリックして選択します。



図 2 : FastCGI アプリケーションの選択

4. 操作ペインから [編集] をクリックします。
5. FastCGI アプリケーション ダイアログ中の「インスタンスの MaxRequests」が 10,000 になっていることを確認します。

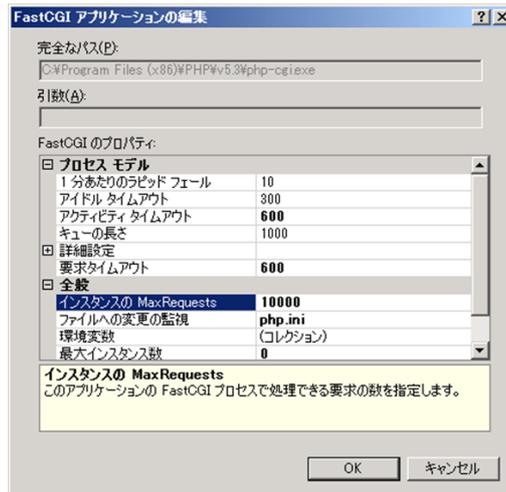


図 3 : インスタンスの MaxRequests の変更

- 「環境変数」を選択し、[参照] ボタン (...ボタン) をクリックします。

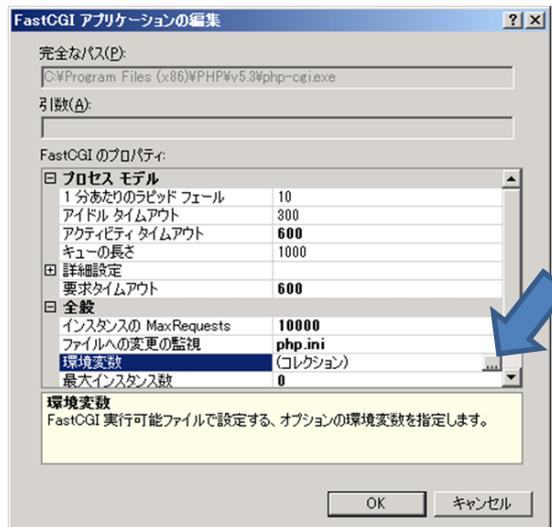


図 4 : FastCGI アプリケーションの環境変数の確認

- PHP_FCGI_MAX_REQUESTS 環境変数が 10,000 になっているのを確認します。

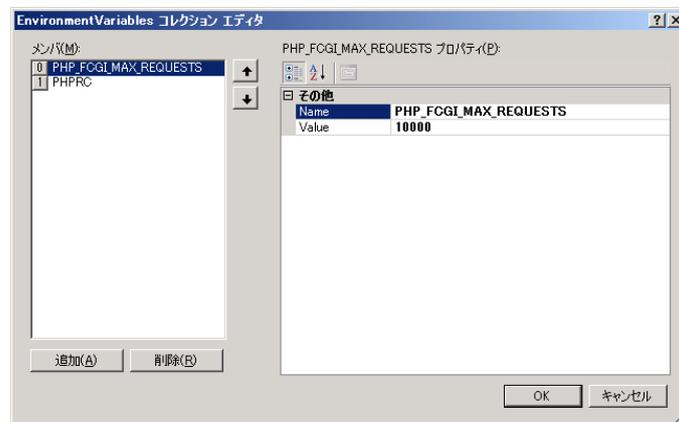


図 5 : PHP_FCGI_MAX_REQUESTS の値を確認

コマンドラインからの実行

コマンドラインから実行する場合には `appcmd.exe` コマンドを使用できます。

同様の設定を行うには以下のコマンドを実行します。

```
C:¥>%windir%¥system32¥inetsrv¥appcmd set config
-section:system.webServer/fastCgi
/[fullPath='c:¥{php_folder}¥php-cgi.exe'].instanceMaxRequests:10000

C:¥>%windir%¥system32¥inetsrv¥appcmd.exe set config
-section:system.webServer/fastCgi
/+"[fullPath='C:¥{php_folder}¥php-
cgi.exe'].environmentVariables.[name='PHP_FCGI_MAX_REQUESTS'
```

フォーム認証の活用

この機能はアプリケーションに加えて、イメージなどの静的なコンテンツもセキュアにすることを可能にします。多くの要求を処理する公開された Web サーバーのサイトやアプリケーションにとっても有効に機能します。この認証モードはクライアントから行われる新規登録や認証を OS や Web サーバーによって提供されているメカニズムを利用するのではなく、アプリケーションのレベルで実施できます。データの登録場所としては既定で組み込まれているメンバーシップ データベースを利用するか、カスタムのメンバーシップ プロバイダーを構築することでほかのデータソースも利用できるようになります。

これまではログオン画面などのアプリケーション保護をご自身で実装しているケースは非常に多いといわれていますが、イメージやメディア コンテンツは公に公開されたまま、保護されていない状態になっているサイトが調べてみると結構多いということが話題になったことがあります。よりセキュアな同一のインターネット ベースの認証スキームを静的コンテンツにも提供するのはカスタムなハンドラーを通じてホストするのではなければなかなか実装が難しかったこともこの一因になっています。パフォーマンス面で不利になる、あるいは標準で利用できる HTTP 圧縮機能や範囲指定のリクエスト処理に対するサポートが使えなくなるなど、ハンドラーを開発する方法論も完全なソリューションではありませんでした。

IIS 7 を使用すれば、フォーム認証および URL 承認ルールをサイト全体に対して一度だけ設定することで、ASP.NET (.aspx)、PHP (.php)、HTML (.html、.htm) およびメディア コンテンツに至るまで認証されていないアクセスでは参照できないように設定されますので安心できます。

UI を使用してフォーム認証を有効にすることも、コマンドラインから Appcmd.exe を実行しても、あるいは構成ファイル (.config) を直接編集する方法、WMI を利用したスクリプトを利用する方法のいずれでも設定が可能です。

GUI ツールによる設定方法

フォーム認証 を IIS マネージャーで設定する方法を説明します。

1. [スタート] - [管理ツール] から IIS マネージャーを開きます。

スタート メニューの検索ボックスで inetmgr と入力して Enter キーを押すことでも可能です。

2. 画面左の階層からサーバー名をクリックして選択します。
3. [認証] をダブル クリックします。

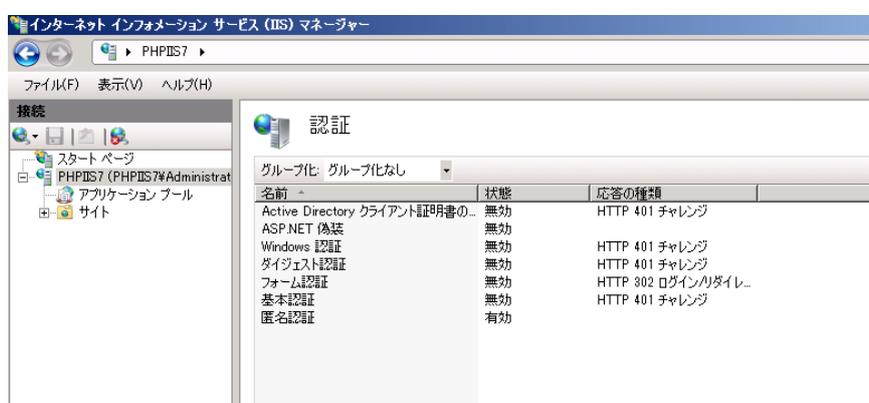


図 6：フォーム認証の設定

4. 一覧から [フォーム認証] をクリックして選択します。
5. 操作ペインから [有効にする] をクリックします。

コマンドラインからの実行

コマンドラインから実行する場合には appcmd.exe コマンドを使用できます。

同様の設定を行うには以下のコマンドを実行します。

```
C:\¥Windows¥System32¥inetsrv>appcmd set config /commit:WEBSITE /section:system.web/authentication /mode: None | Windows | Passport | Forms
```

フォーム認証についての詳細は下記のサイトなどを参照ください。

方法：簡単なフォーム認証を実装する

[http://msdn.microsoft.com/ja-jp/library/xdt4thhy\(VS.80\).aspx](http://msdn.microsoft.com/ja-jp/library/xdt4thhy(VS.80).aspx)

HTTP キープ アライブの利用

HTTP キープ アライブ ヘッダーはクライアント、サーバー間の接続を複数のリクエスト間で共有することにより、Web サーバーのパフォーマンスを改善します。継続して利用される接続はクライアントが Web ページのコンテンツを取得する際のリクエストする時（通常 複数接続を要する）パフォーマンスを改善します。これはサーバーがより早くコンテンツを返送できるためです。キープ アライブが利用されていない場合には、サーバーはリクエスト毎に接続を行う必要があります。IIS7 では既定で HTTP キープ アライブ ヘッダーは有効になっています。

GUI ツールによる設定・確認方法

HTTP キープ アライブ を IIS マネージャーで設定する方法を説明します。

1. [スタート] - [管理ツール] から IIS マネージャーを開きます。

スタート メニューの検索ボックスで inetmgr と入力して Enter キーを押すことで実行することも可能です。

2. 画面左の階層からサーバー名をクリックして選択します。
3. 機能ビューで [HTTP 応答 ヘッダー] をダブル クリックします。
4. 操作ペインから [共通ヘッダーの設定] をクリックします。

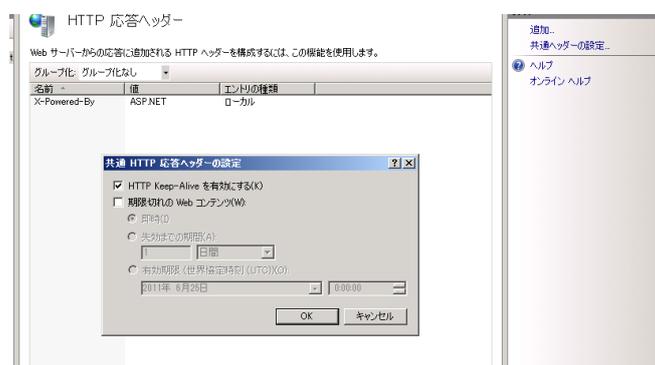


図 7 : HTTP キープ アライブ ヘッダーの設定

コマンドラインからの実行

コマンドラインから実行する場合には appcmd.exe コマンドを使用できます。

同様の設定を行うには以下のコマンドを実行します。

```
C:¥Windows¥System32¥inetsrv>appcmd set config /section:httpProtocol /allowKeepAlive:true | false
```

ネットワーク帯域の調整

メディア コンテンツを提供するサイトが増大するにつれ、多くのビジネス シーンではネットワーク帯域を確保するためのコストがどんどん高額になってきています。加えて、実はメディア コンテンツを提供するための帯域というのは多くの割合で無駄に利用されています。クライアントで実は利用されないデータも含めて通信を行ってしまうためにこのようなことが起こります。ビデオ サイトをブラウズしているときにビデオの一部だけを見て次のビデオへ移動する、あるいは別のページを見ろという操作は実は非常に多く行われていることです。しかしながら、プログレッシブ ダウンロード方式を使用して Web サーバーからビデオを配信している場合には、実際に視聴されているよりも遥かに多くのデータが再生のために必要になります。最終的に通信しつつも、実際には全く使われないこのようなデータが実際に存在するのです。

たとえば、もしあなたがホストしているビデオが平均 5 秒間しか見られないという想定で、しかしバッファのために 30 秒分のビデオ データを毎回送信しているとすると、非常に粗い計算で言うと帯域の **80%**を無駄にしていることとなります。

ビットレート調整モジュール

IIS7 では ビットレート調整モジュールという拡張機能が利用可能です。このモジュールは自動的によくあるビデオ タイプのエンコード レートを検知します。バッファによる遅延を起こさないようにするためにどれくらいクライアントに初期段階でデータを送付するかを制御することができます。また、コンテンツの配信をどのくらいのエンコード レートの割合で送付したいかも指定することができます。これ以外にも多くのオプションがあり、最大帯域幅や同時接続数も制御できるほか、プログラムから自動化することも可能になっています。

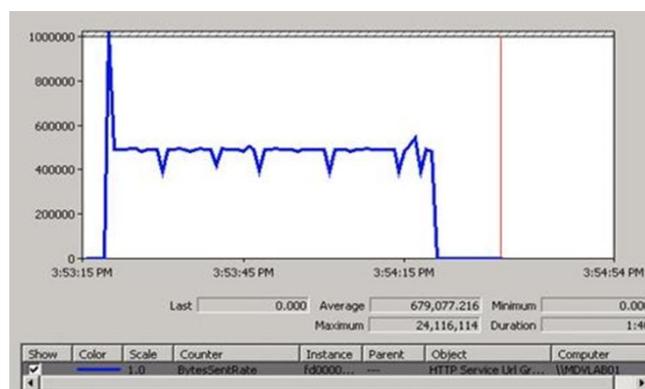


図 8 : ビットレート調整モジュールによる帯域幅調整

サイトへの接続数を制限する

過負荷を避けるために各 Web サイトに対する同時接続数を制限することも可能で、この方法で安定したパフォーマンスを提供することも一つの選択肢です。なお IIS の同時実行数上限の既定値はリソースの限界と同じなので一定数ではありません。

GUI ツールによる設定・確認方法

Web サイトの同時接続数を IIS マネージャーで設定する方法を説明します。

1. [スタート] - [管理ツール] から IIS マネージャーを開きます。

スタートメニューの検索ボックスで inetmgr と入力して Enter キーを押すことで実行することも可能です。

2. 画面左の階層からサーバー名をクリックして選択します。
3. 操作ペインから [制限] をクリックします。
4. [接続数を制限する] チェックボックスにチェックを入れます。
5. 制限したい接続数を空欄に入力します。

メモ：ブラウザの接続は実は平均的に 4 セッション程度同時に使用すると試算してください。

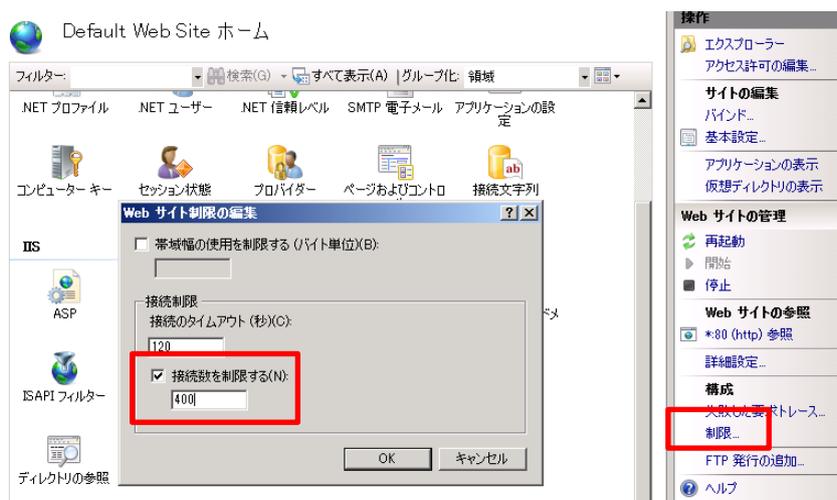


図 9：サーバーに対する同時接続数の制限

6. [OK] をクリックします。

CPU モニタリングを有効にする

非常に多くの CPU 時間を使うワーカプロセスについては CPU モニタリングを使用して監視をしつつ、自動的にワーカプロセスをシャットダウンするようなアクションを設定できます。

GUI ツールによる設定・確認方法

CPU モニタリング を IIS マネージャーで設定する方法を説明します。

1. [スタート] - [管理ツール] から IIS マネージャーを開きます。

スタート メニューの検索ボックスで inetmgr と入力して Enter キーを押すことで実行することも可能です。

2. 画面左の階層から指定したいアプリケーション プールをクリックして選択します。
3. 操作ペインから [詳細設定] をクリックします。
4. [制限] のプロパティ値を任意の割合に設定します。(0は無効)
5. OK をクリックします。

HTTP 圧縮の利用

アプリケーション レスポンスを返すのに必要なネットワーク帯域を削減する最も効果的な方法論の一つは HTTP 圧縮を用いることです。この機能を用いることでレスポンスのデータ量を大幅に削減することができます。HTML のような非常に圧縮効果が高いコンテンツが多い場合には 1/10 まで削減できることもしばしばです。HTTP 圧縮は HTTP 1.1 プロトコルの Content-Encoding ネゴシエーションに基づいているので圧縮をサポートしないクライアント（ブラウザ）が存在する場合でも有効にすることができます。このようなクライアントは圧縮されていないコンテンツを受け取ることになるだけです。

IIS 7 は二種類の HTTP 圧縮 を提供します。

- 静的圧縮
- 動的圧縮

静的な圧縮は静的なコンテンツにリクエストがあった際にディスクに圧縮したコンテンツを保管する仕組みになっているので次のリクエストからは直接この保管されているコンテンツを参照するために圧縮する処理のオーバーヘッドなしに利用が可能です。動的な圧縮はリアルタイムでコンテンツを圧縮しますのでアプリケーションが生成する動的なレスポンスを圧縮できます。IIS7 上で動作するどのようなアプリケーション フレームワークも動的な圧縮を利用でき、これには ASP、ASP.NET だけでなく PHP も含まれます。

動的な圧縮は圧縮処理をリアルタイムに行うのでかなり独占的な CPU オーバーヘッドが発生すると思われるかもしれませんが、そんなことはありません。かなり忙しいサーバーでも動的な圧縮は CPU 稼働率の 5% 未満で済むケースが多いです。このことから多くの Web サイトでこの動的な圧縮は利用されており、どのようなアプリケーション ワークロードにおいても検討するに値します。さらに圧縮のオーバーヘッドを最適化したい場合には圧縮の強度を変更することで圧縮度合と CPU オーバーヘッドのバランスを調整することができます。

UI を使用して HTTP 圧縮を有効にすることも、コマンドラインから `Appcmd.exe` を実行しても、あるいは構成ファイル (`.config`) を直接編集する方法、WMI を利用したスクリプトを利用する方法のいずれでも設定が可能です。

GUI ツールによる設定・確認方法

CPU モニタリング を IIS マネージャーで設定する方法を説明します。

1. [スタート] - [管理ツール] から IIS マネージャーを開きます。

スタート メニューの検索ボックスで `inetmgr` と入力して Enter キーを押すことで実行することも可能です。

2. 画面左の階層から指定したいサイトをクリックして選択します。
3. [圧縮] をダブル クリックします。
4. 有効にしたい圧縮の種類にチェックを入れ、操作ペインから [適用] をクリックします。

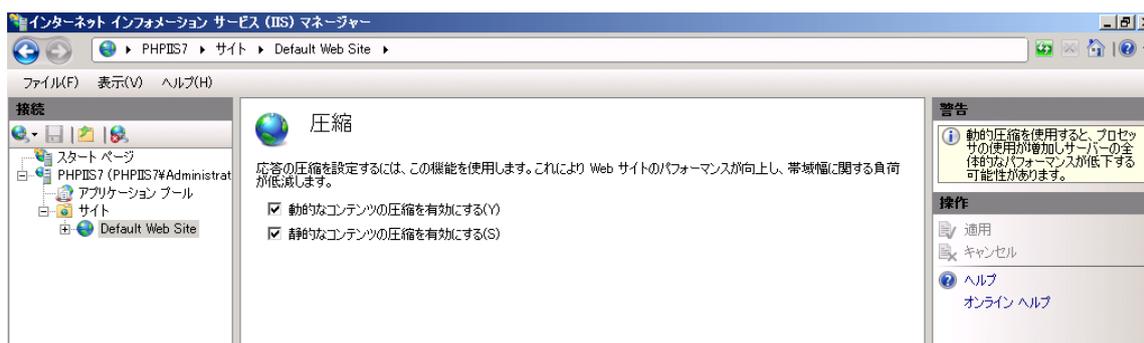


図 10 : HTTP 圧縮の設定

コマンドラインからの実行

コマンドラインから実行する場合には `appcmd.exe` コマンドを使用できます。たとえば、動的な圧縮をコマンドラインで有効にしたい場合には以下のコマンドを実行し、Enter キーを押すだけで設定可能です。

```
C:\¥Windows¥System32¥inetsrv>appcmd set config /section:urlCompression /doDynamicCompression:True
```

既定のドキュメント（インデックス ファイル）設定

IIS7 では以前の IIS とは異なるシステム構成ファイルを採用しており、web.config といいます。このファイルはアプリケーションのどの階層に置くことも可能で、その配下のディレクトリに対して有効な Web サーバー ディレクティブを含みます。

PHP の多くのアプリケーションでは .htaccess という名前のファイルを配布していますが、これと非常に似た役割を web.config は果たします。同様に httpd.conf に類似して存在するのが applicationHost.config というシステム構成ファイルです。

web.config ファイルを利用してアプリケーション ディレクトリのインデックス ファイルの設定をしたい場合には以下のような構成を記述することで実現します。

```
<configuration>
  <system.webServer>
    <defaultDocument>
      <files>
        <remove value="index.php" />
        <add value="index.php" />
      </files>
    </defaultDocument>
  </system.webServer>
</configuration>
```

この例で登場する remove のエントリーは必要です。すでに index.php のエントリーがある場合にはこの指定によりそれを削除し、新しいエントリーを一覧の一番上に配置します。エントリーが存在しなかった場合には remove は無視されます。

コラム：構成ファイルの設定がわかったけど手順は？

IIS7 の文献を読んでいるとこういうケースが散見されるので困る人も多いと思いますので少しかだけアドバイスのつもりでこのコラムを書きます。IIS7 の各種設定の根幹はほとんどの場合 .config の拡張子のつくファイルに記載があります。ほとんどと書いたのは今なおレジストリを選択せざるを得なかった設定が散見されるためです。

直前の既定のドキュメントの例を見てみましょう。コマンドでこれを実行したい、よく設定作業をする人は確実に思うはずですが、ここで便利なのが、構成エディターというツールです。IIS7.5では標準機能になっており、IIS7.0はAdministration Packで入手できます。以下、実際の手順を書きます。

1. [スタート] - [管理ツール] から IIS マネージャーを開きます。
2. 画面左の階層から指定したいサイトをクリックして選択します。
3. [構成エディター] をダブル クリックします。
4. 上部の 場所 ドロップダウンから記載したい構成ファイルを選択します。
5. セクション ドロップダウンから今回の例で言えば、system.webServer - defaultDocument を展開しながら開きます。
6. Count=n となっている欄の右にある参照ボタンをクリックします。
7. コレクション エディター内の操作ペインで [追加] をクリックします。
8. プロパティの入力欄で index.php と入力して Enter キーを押します。
9. 上部一覧に表示されますのでエディターを閉じて戻ります。
10. ここで操作ペインにある [スクリプトの生成] をクリックします。

ここまでの操作で C#、JavaScript、Appcmd.exe の3種類に対して今行った画面操作を自動化する方法を教えてくださいますのでかなり便利です。

さて、この方法論には若干残念な点があります。それは設計された時期に Windows PowerShell がまだ出荷されていなかったこと、Appcmd.exe の制限でコレクションの一覧をクリアする機能が無い点です。でも Appcmd.exe のパラメーターの表現方法を知らなくてもこのツールで実際に構成を設定するに加え、スクリプトのヒントももらえるのはかなり有効なはずですが。

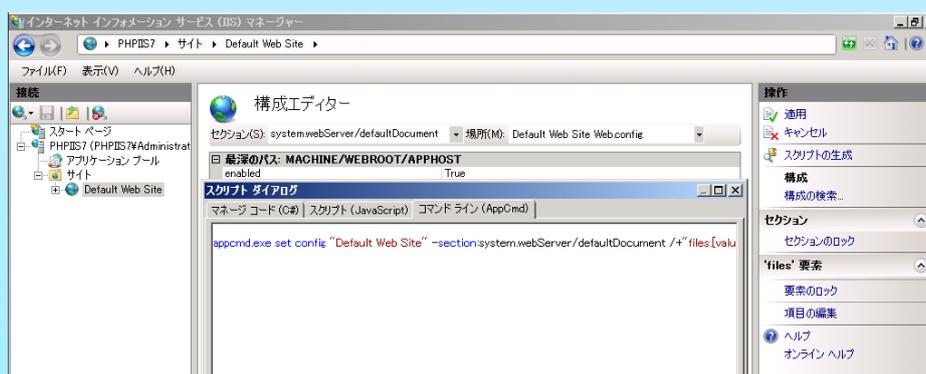


図 11：構成エディターのスクリプト生成機能

もちろん管理者実行したエディターで .config ファイルのセクションを直接いじることも可能なので、ほかの Web サーバーの経験が多い方はそちらの方が便利に思うかもしれません。

要求フィルター機能の活用

IIS7 は要求フィルターモジュールを利用して特定の条件に基づいてブラウザから指定したファイルやディレクトリのアクセスを制限することができます。URL フィルター モジュールはさまざまな URL へのアクセスをセキュアに制限するための強力なエンジンです。

要求フィルターを使用する例として、特定のファイルの種類たとえばログファイルへのアクセスを制限する例を見てみましょう。

```
<configuration>
  <system.webServer>
    <security>
      <requestFiltering>
        <denyUrlSequences>
          <add sequence=".inc"/>
          <add sequence=".log"/>
        </denyUrlSequences>
      </requestFiltering>
    </security>
  </system.webServer>
</configuration>
```

IIS 拡張モジュールの利用

IIS7 には標準機能以外に数多くの拡張モジュールがマイクロソフトから提供されていますので一度どんなものがあるか調べておくといよいでしょう。ここでは URL 書き換えモジュールの利用例を見ておきましょう。

URL 書き換えモジュールの利用

IIS7 には URL 書き換えモジュールがあります。このモジュールでは正規表現を用いたリライトルールを定義可能です。リライト ルールは手動で構成することも可能ですし、IIS マネージャーを利用して構成することも可能です。URL 書き換えモジュールは mod_rewrite のルールをインポートする機能がありますので、完全に変換可能ではないものの、web.config への変換作業を助けてくれます。

よくあるリライト ルールのサンプルを下記に記載します。この例は example.com にアクセスしてきたすべてのリクエストを <http://www.example.com> にリダイレクトするものです。

```
<configuration>
<system.webServer>
  <rules>
    <rule name="Imported Rule 1" stopProcessing="true">
      <match url="^(.*)$" ignoreCase="false" />
      <conditions>
        <add input="{HTTP_HOST}" pattern="^example¥.com$" />
      </conditions>
      <action type="Redirect" redirectType="Permanent" url="http://www.example.com/{R:1}" />
    </rule>
  </rules>
</system.webServer>
</configuration>
```

IIS の管理

ここまでのいくつかの設定は IIS マネージャーという GUI ツールを使う方法論、Appcmd.exe という IIS 開発チームが標準で用意したコマンドを利用していました。実は IIS7 は数多くのスクリプトを行うためのインタフェースを備えており、以下では Windows Server 2008 から登場した Windows PowerShell を使う方法について説明します。

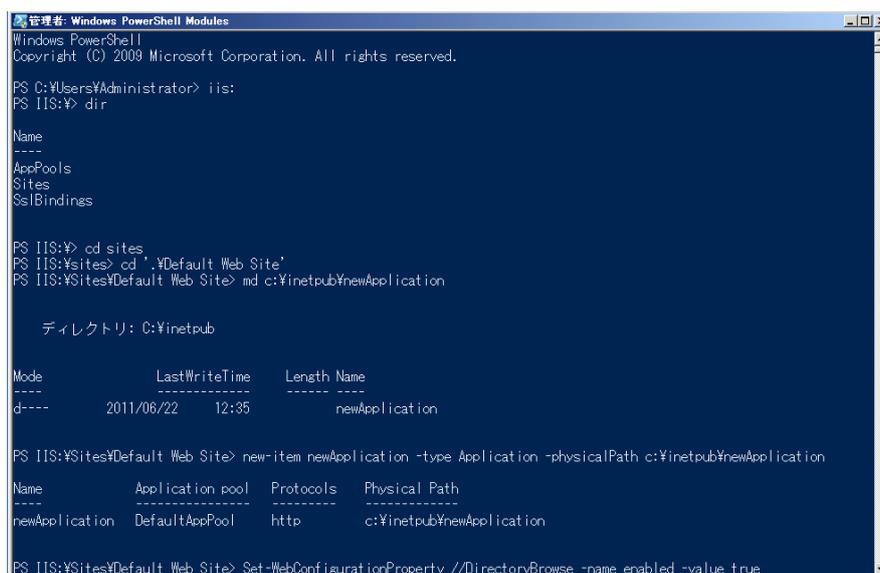
メモ：今までの Windows でスクリプト管理と言えば VBScript を用いた WMI 操作だったわけですが、今後の Windows では益々 Windows PowerShell が主役になっていきますのでこのあたりにご興味がある方はぜひ今後は Windows PowerShell を使うようにしましょう。

Windows PowerShell での IIS7 管理

IIS7 PowerShell プロバイダーは Windows PowerShell スナップインになっており、IIS7 構成データの管理、実行状態データの管理を可能にします。Web サイト、アプリケーション プール、Web アプリケーション、仮想ディレクトリへの容易なアクセスが可能です。また今までの IIS では困難であった 実行中のリクエスト情報、実行中のワーカプロセス、プロセス内で稼働している .NET アプリケーション ドメインへのアクセスも可能になっているために相当パワフルな管理機能になっています。

Windows PowerShell コマンド ライン シェルおよびスクリプト言語はサーバー管理者に今までとは比較できない制御範囲と生産性をもたらします。管理者を意識した新しいスクリプト言語、130以上の新しい標準コマンド、コマンドを覚えやすくするルールに基づいた命名規則や記述方法 などを通じてサーバー管理者は今までにない容易なシステム管理と自動化の促進が可能です。Windows PowerShell は .NET 共通言語ランタイム (CLR) および .NET Framework に基づいており、.NET のオブジェクトのやりとりが可能です。

下図の例では、PowerShell プロバイダーを用いて新しい IIS アプリケーションを作成する様子を表示したのになっています。



```
管理権: Windows PowerShell Modules
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> iis:
PS IIS:\> dir

Name
----
AppPools
Sites
SslBindings

PS IIS:\> cd sites
PS IIS:\Sites> cd '..\Default Web Site'
PS IIS:\Sites\Default Web Site> md c:\inetpub\newApplication

ディレクトリ: C:\inetpub

Mode                LastWriteTime         Length Name
----                -
d-----           2011/06/22   12:35         newApplication

PS IIS:\Sites\Default Web Site> new-item newApplication -type Application -physicalPath c:\inetpub\newApplication

Name                Application pool  Protocols  Physical Path
-----
newApplication      DefaultAppPool   http       c:\inetpub\newApplication

PS IIS:\Sites\Default Web Site> Set-WebConfigurationProperty //DirectoryBrowse -name enabled -value true
```

図 12 : Windows PowerShell を使用したアプリケーションの作成

IIS7 PowerShell プロバイダーを使用して実施できることの例を下記に挙げます。

- アプリケーション プール、Web サイト、Web アプリケーション、仮想ディレクトリの作成
- アプリケーション プール、Web サイト、Web アプリケーション、仮想ディレクトリの単純な構成プロパティの変更
- 複雑な構成設定の追加と変更
- 実行状態データのクエリー (Web サイトの状態、アプリケーション プールの状態、現在実行されているリクエストなど)
- ほかのスナップインや機能との統合によるさらに複雑な構成タスクやスクリプト実行
- 構成設定の検索や発見

たとえば、PowerShell プロバイダーを利用して新しいアプリケーション プールを作成することができます。この処理に必要なパラメーターはプールの名前だけです。

```
PS IIS:¥> new-item AppPools¥DemoAppPool
```

Name	State	Applications
----	-----	-----
DemoAppPool	Started	

この IIS7 PowerShell プロバイダーについては以下の記事を参照ください。

IIS 7.0 PowerShell スナップインを使用した IIS の管理

<http://technet.microsoft.com/ja-jp/library/ee175763.aspx>

IIS7 の拡張

IIS7 のアーキテクチャは上から下まで拡張できることを念頭に設計されています。結果として、標準で搭載されているビルトインの機能を要件に合う独自実装に部品交換することがかなりの自由度で実現可能です。このことから IIS 開発チーム自身も公開された拡張方法で機能を実装しています。このような構造的な特徴・設計は IIS のシステムを通じて加味されており、モジュール構造な Web サーバーそのもの、新構成システム (applicationHost.config, web.config) そして GUI 管理ツールである IIS マネージャーも拡張可能となっています。

実装しようとしている Web システムには IIS の標準機能に含まれていないものが必要になることがあります。あるいは当該のアプリケーションは標準の機能では柔軟さが足りない要件がある状況に遭遇することもあるでしょう。IIS7 の全機能は公開されている拡張可能な API で構築されているために、ニーズに最適なお自身で開発したカスタム実装に変更することも容易にできます。

IIS マネージャーの拡張

標準で IIS に付属している他のコンポーネント同様に、開発者は IIS マネージャーを拡張して独自の機能の UI を IIS マネージャーと統合した形で提供することができます。IIS マネージャー拡張を利用することで、IIS マネージャーが想定している様々な要素（例：リモート管理が容易、HTTP 通信で管理）を活用することができます。

IIS マネージャーは標準でリモート管理をする機能を備えています。IIS マネージャーとサーバー間で HTTPS 通信を行うこの機能は、IIS マネージャーを利用するユーザーを独自管理する機構も持

っており、その管理者情報を IIS の構成システム、Windows のアカウント管理、Active Directory、.NET プロバイダーを使用したデータソースに保管することも可能になります。

リモート管理をサポートするほかのツールと違い、IIS7 の IIS マネージャーのリモート管理アーキテクチャはいくつかの優位性を持っています。サーバーの管理者権限を持たないユーザーに対してサイトやアプリケーション単位で管理権限を委任することができます。次に、IIS マネージャーのリモート管理は DCOM ではなく HTTPS を通信メカニズムに利用していますのでファイアウォールが介在するような遠隔管理がしやすいです。これらの機能を総合すると、IIS Web サイトのリモート管理を委任するようなシナリオ、たとえば共有ホスティングを行う環境では強力です。実際に米国の共有ホスティング サービスの中には契約ユーザーが IIS マネージャーで接続して管理することを許可したサービスも登場しています。

IIS マネージャーは標準機能のほとんどが実際に利用している拡張可能なアーキテクチャを提供しています。リモート管理機能を活かすことができるように、各管理機構は基本的に二つのパーツでできています。IIS マネージャー内の UI 処理を提供するクライアントのコンポーネント、および実際の管理を行うための機能を実装したサーバー側のコンポーネントです。サーバー側のサービスはローカルで管理している際には IIS マネージャー内に読み込まれますが、リモート管理の場合には Web 管理サービス内に呼び込まれる仕組みになっています。後者では IIS マネージャーは必要なサーバー (Web 管理サービス) とクライアント間の通信を引き受けて処理します。

システム構成スキーマの拡張

システム構成スキーマを拡張することで実装しようとしているアプリケーションが便利になる可能性があります。アプリケーションの仮想ディレクトリへの展開を自動化しやすくなるメリットが考えられます。あるいはアプリケーション内の権限を IIS のシステム構成およびサイトや Web アプリケーションと紐づけてアクセス権限を管理するような統合シナリオも考えられます。

スキーマの拡張は実際には簡易な書式の xml 形式ファイルを %SystemRoot%\system32\inetsrv\config\schema ディレクトリに置くところからスタートします。IIS_schema.xml ファイルは IIS をインストールすると既に配置されており、ご自身のスキーマを組み立てる参考になります。CUSTOM_schema.xml のような任意のファイル名で新規作成し、applicationHost.config 内でセクション定義を行う必要があります。これも applicationHost.config ファイルに一行追加するだけです。

システム構成スキーマの拡張に関する詳細な情報は、下記の情報を参照ください。

構成の拡張

<http://technet.microsoft.com/ja-jp/library/ff453978.aspx>

Windows のレベルで考えられる点

ここまでは IIS に直接関係するポイントを書きましたが、この項では Windows OS のレベルで気にかけておきたいポイントを取り上げます。まずは以降の話がわかるようにコラムから入ります。

コラム：サーバー OS とクライアント OS の機能の取り扱い

Windows Server 2003 と Windows XP を比較するとわかりやすいですが、このころから顕著にサーバー OS とクライアント OS の違いが少しずつ増えてきました。このため、最新の Windows Server 2008 R2 ではクライアント OS（Windows Vista や Windows 7）と操作の違いがところどころにあり、考え方も少し違うところがあります。

サーバー OS ではファイル サーバー、Web サーバーなどの役割を担うことが多いです。結果として、Windows Server でも機能やドライバーをインストールする考え方の上位概念として「役割をインストールする」という考え方が搭載されています。ファイル サーバーに必要な A と B と C という機能をインストールすることも引き続きできるのですが、「ファイル サーバーをインストールする」という指示ができるようになっています。

通常 Windows で機能やソフトウェアをインストールしようとするときにはコントロール パネルから追加と削除するのは有名ですが、その単位はあくまでもクライアント OS の考え方であって、役割についての取り扱いは今の Windows Server ではサーバー マネージャーから管理操作を行うこととなります。

もう一つ、Windows にはサービスという考え方があり、管理ツールからサービスのアイコンを使用して操作したことがある方も多いでしょう。このサービスと役割の関係にふれておくと、役割 1 に対してシステム サービスは複数になることがあります。つまり、IIS という Web サーバーを取り上げると依存サービスも含め、7つのサービスが IIS 関連ということになります。

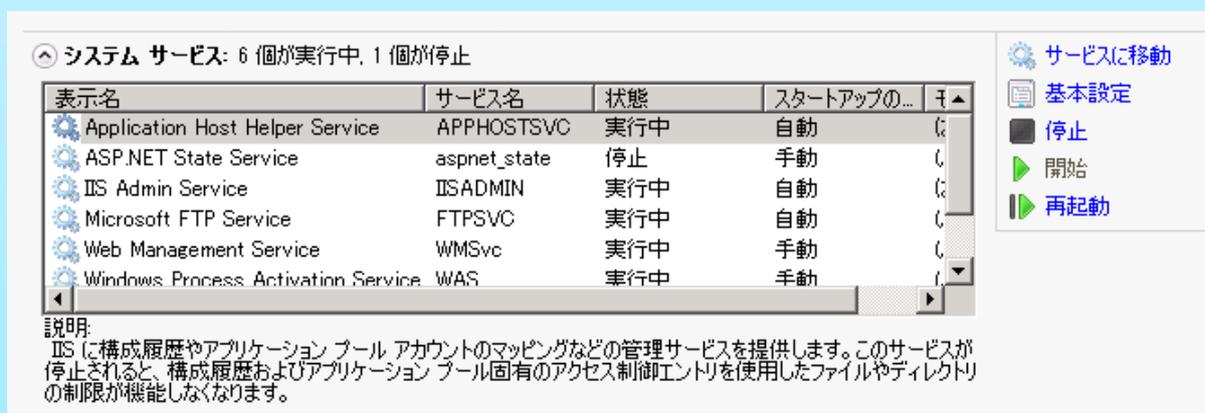


図 13：サーバー マネージャーで見た Web サーバー関連のシステム サービス

さらに補足すると似た言葉で役割サービスというのが登場します。これが少しわかりにくいですが、明らかに役割に関係する機能のことなのですが、コントロール パネルで追加・削除できる Windows 機能と同じものを指しているケースもあるからです。基本的に役割を構成する個別の機能 (=コントロール パネルでも追加・削除可能) を役割サービスと呼ぶと思ってください。

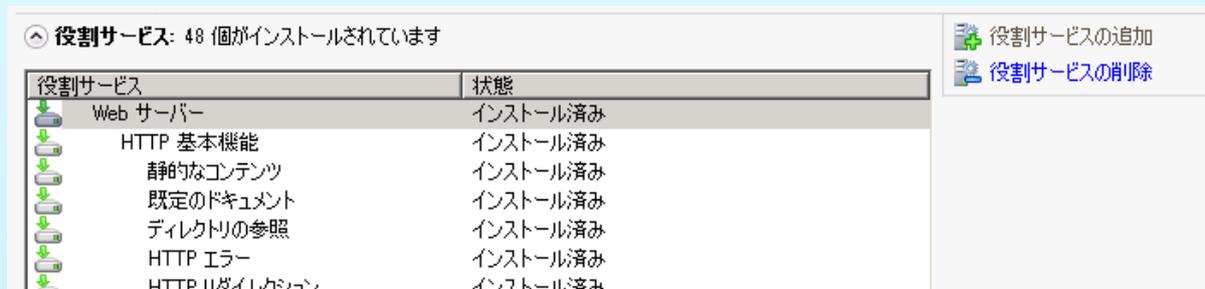


図 14 : サーバー マネージャーで見た Web サーバーの役割サービス

ここまで登場したもの以外に環境周りと言うと、デバイス ドライバーの設定を見るためのデバイス マネージャーなどもあり、ドライバーの編集・削除が可能です。この他にもネットワークの設定など、もっといろいろな設定があります。

総合すると、Windows ってシンプルに見えて裏でやっていることはかなり複雑なことやっているわけです。

Windows Server 2008 R2 の管理ではいくつかのコマンド ライン ツールが登場しますが、上記背景に基づき、それぞれ特徴があります。

Pkgmgr.exe:

コントロール パネルの粒度の細かい機能単位での環境構成を行う
パラメーターを渡すことでインストール作業を行え、このコマンドは Windows 7 でも
Windows Server 2008 R2 でも利用可能です。

ServerManagerCmd.exe:

サーバー マネージャーのコマンド版。Windows PowerShell 登場で将来は使われなくなるツール

Windows PowerShell の Add-WindowsFeature:

今後はこれが最も使われるといわれています。

ここまでの違いを知っておくとほかのガイドラインなども読み進めやすくなるでしょう。

不必要なサービスとプロセスの無効化

Windows Server 2008/R2 において不要な役割、役割サービス、機能がインストールされていないことを確認しましょう。不要なサービスの例としては、以下のようなものです。結構当たり前と思っている方が多いものもありますが、実際の環境では動作しているケースが結構あります。

- システム サービス (管理ツール - サービス)
 - Microsoft Find Fast などのインデックス サービス
 - 不必要なメール関連サービス
- デバイス関連 (デバイス マネージャー)
 - 3次元の描画を行うようなスクリーン セーバー (CPUのリソースを消費)
 - 不必要なマウスや赤外線などのハードウェア ドライバー
- ネットワーク関連 (ネットワークと共有センター)
 - 使用しないネットワーク プロトコル
- アプリケーション (アプリケーションの追加と削除)
 - 使用されていないアンチ ウィルス のスキャナー

GUI ツールによる設定・確認方法

それでは Windows のシステム サービスを無効にする例を説明します。

1. [スタート] メニューから [コントロール パネル] を開きます。
2. [管理ツール] をクリックし、[サービス] を実行します。
3. 無効にしたいサービスまでスクロールし、クリックして選択します。
4. 右クリックし、[プロパティ] を開きます。
5. スタートアップの種類の項目で「無効」を選択します。
6. [サービス] を閉じます。

メモ：サービスを停止・無効にする際に Web サーバーが正しく動作していることを確認しましょう。また、依存関係にある別のサービスにも影響があるので注意しましょう。[プロパティ] を開いた際に「依存関係」のタブがあるのでそこをチェックしましょう。

以前のバージョンの Windows ではサーバーの役割を追加あるいは削除する際に依存関係をチェックする機構はそれほど発達していませんでした。Windows Server 2008 以降ではこの点が強化され、サーバーの構成は XML 形式でメモリー内管理されており、サーバー マネージャーもこの情報を

用いて動作するようになっていました。このような構造で動作するようになったことからサーバーの役割や機能の自動インストール・アンインストールを行うこともコマンドやスクリプトで非常に簡単にできるようになっています。

Windows Server 2008 R2 SP1 が登場している現在では最もおすすめの方法は Windows PowerShell の ServerManager コマンドレットに含まれるコマンドを使用することです。

IIS7 の全標準機能を搭載した状態でインストールするには下記のコマンドを Windows PowerShell Modules を起動した状態で実行します。

```
PS >Add-WindowsFeature Web-Server -I
```

※-I = IncludeAllSubFeature

現在インストールされている役割や機能の一覧を把握したい場合には下記を実行します。

```
PS >Get-WindowsFeature
```

Windows PowerShell による OS やほかの役割も含めての一般的な管理手法についての情報は下記をご覧くださいと豊富に情報があります。

Windows PowerShell でのスクリプティング

<http://technet.microsoft.com/ja-jp/scriptcenter/powershell.aspx>

最後に

PHP on Windows ガイドライン（ドラフト版）を最後までお読みいただきましてありがとうございます。基本的にこのガイドは「[インターネット Web サーバー構築ガイドライン](#)」と対になるものなので、IIS のより詳細な話を読みたい方はそちらもぜひご覧ください。

Windows で PHP を利用する際の情報が少ないというご要望にお応えする形でこのガイドラインは執筆して参りましたが、無論 足りない情報がまだまだあると認識しております。今後も皆様のご希望に沿えるような様々な活動、執筆をしていきますのでご要望は積極的に寄せいただきますよう、この場をお借りしてお願い申し上げます。

リンク集

TechNet オンライン

TechNet オンラインはマイクロソフト製品のインフラ系の技術情報を掲載しているサイトです。IIS の情報はまずは Windows Server の下に存在し、いくつかの記事を下記にリンクします。

TechNet IIS TechCenter ～ 日本語の IIS 関連技術ポータル ～

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

IIS のパフォーマンスを最適化する (IIS 7)

[http://technet.microsoft.com/ja-jp/library/cc770381\(WS.10\).aspx](http://technet.microsoft.com/ja-jp/library/cc770381(WS.10).aspx)

IIS 7.0 のパフォーマンス向上機能トップ 10

<http://technet.microsoft.com/ja-jp/magazine/2008.09.iis.aspx>

ログの記録を有効または無効にする (IIS 7)

[http://technet.microsoft.com/ja-jp/library/cc754631\(WS.10\).aspx](http://technet.microsoft.com/ja-jp/library/cc754631(WS.10).aspx)

IIS 7 で HTTP 圧縮を構成する

[http://technet.microsoft.com/ja-jp/library/cc771003\(WS.10\).aspx](http://technet.microsoft.com/ja-jp/library/cc771003(WS.10).aspx)

フォーム認証を有効にする (IIS 7)

[http://technet.microsoft.com/ja-jp/library/cc771077\(WS.10\).aspx](http://technet.microsoft.com/ja-jp/library/cc771077(WS.10).aspx)

MSDN オンライン

MSDN オンラインはマイクロソフト製品の開発者向け技術情報を掲載しているサイトです。

エンドツーエンドの拡張性を持つ Web サーバー ソリューションを構築する

<http://msdn.microsoft.com/ja-jp/magazine/cc164241.aspx>

Windows Server Performance Team Blog

Windows Server の開発チームの中でも特にパフォーマンスを専門としている部隊のブログです。

Tuning Windows Server 2008 for PHP (英語)

<http://blogs.technet.com/b/winserverperformance/archive/2008/07/25/tuning-windows-server-2008-for-php.aspx>

ServerSide

Mike Volodarsky は IIS7.0 開発時の開発チーム主要メンバーでした。現在 Microsoft MVP です。

スライド ショー (英語)

<http://mvolod.com/photos/serverside/slideshowpro.aspx>

ScottGu's Blog

Web に関係する開発ツール製品の開発チームを担当する副社長、スコット ガスリーのブログです。

IIS 7.0 Bit Rate Throttling Module Released (英語)

<http://weblogs.asp.net/scottgu/archive/2008/03/18/iis-7-0-bit-rate-throttling-module-released.aspx>

※この記事そのものの翻訳はないのですが、Scott のブログを翻訳しているサイトがあり、下記です。

<http://www.atmarkit.co.jp/fdotnet/scottgublog/index/index.html>

IIS.NET および 翻訳版 (TechNet)

IIS.NET は IIS 開発チームが中心となって構築したコミュニティ サイトです。IIS 開発チームも多く情報発信をしています。しかし、英語なので特に learn.iis.net 配下の記事に関しては TechNet 日本語サイトにて翻訳記事を配置していますので、ここでは翻訳記事の方をご案内します。

Bit Rate Throttling is now released (英語)

<http://blogs.iis.net/vsood/archive/2008/03/15/bit-rate-throttling-is-now-released.aspx>

IIS マネージャーで IIS 7 用のモジュール プロパティを拡張する方法

<http://technet.microsoft.com/ja-jp/library/ff453977.aspx>

IIS 7 統合パイプラインの利点の活用方法

<http://technet.microsoft.com/ja-jp/library/ff454025.aspx>

要求フィルターの使用方法

<http://technet.microsoft.com/ja-jp/library/dd939071.aspx>

Apache mod_rewrite ルールのインポート

<http://technet.microsoft.com/ja-jp/library/dd939075.aspx>

.htaccess ファイルを web.config に変換する際の注意事項

<http://technet.microsoft.com/ja-jp/library/dd647603.aspx>

Web 技術系のエバンジェリスト ブログ

日本マイクロソフト株式会社のエバンジェリストで Web 技術に関するテクノロジーのブログを書いているもので、PHP のタグのあるものはタグの参照先になっています。

Hiroshi Okunushi's Blog ☆ミ | IIS PHP etc. (執筆者)

<http://blogs.technet.com/b/hirookun/archive/tags/php/>

monoe's blog (執筆者)

<http://blogs.msdn.com/b/osamum/archive/tags/php/>

WebPI の歩き方 by 武田正樹

http://blogs.msdn.com/b/web_/archive/tags/php/

Days with Microsoft Web Platform

<http://blogs.msdn.com/b/tashinmu/>

THE TRUTH IS OUT THERE

<http://blogs.msdn.com/b/chack/>

その他のサイト

Microsoft.com 系のほかの関連サイトです。ご参考まで。

Microsoft /web サイト – 世界共通の Web 技術ポータル

<http://www.microsoft.com/web/>

Microsoft Japan /web サイト - 斬新なデザインがウリ!

<http://www.microsoft.com/japan/web/>

WebMatrix のサイト

<http://www.microsoft.com/japan/web/webmatrix>