

【B3】PHPテクニカルセッション



EMBARCADERO
TECHNOLOGIES®

DEVELOPER CAMP

DelphiでPHP

- 拡張モジュールの作成からWAMP/WIMP環境の構築まで -

エンバカデロ・テクノロジーズ エヴァンジェリスト
高橋 智宏

使い慣れた開発環境で、
PHP Extension Moduleを作成して、
WAMP / WIMP 環境を拡張でき、
日本語対応も、デバッグもOKだとしたら、
うれしくないですか？
し・か・も、超超超カンタンに。

- PHP拡張モジュールの作成
- VCL for PHPでカスタムコンポーネントを作成
- WAMP / WIMP 環境の構築



EMBARCADERO
TECHNOLOGIES®

DEVELOPER CAMP

PHP拡張モジュールの作成

- PHP拡張モジュールを新規に作成したことがある人いますか？
- PHP言語から呼び出し可能な関数を実装した、OSネイティブなライブラリ
 - Windowsなら、php_mysql.dll など
 - php.ini に extension=php_mysql.dll と記述してロードする
- 良いところ
 - PHPだけでは実現が難しい処理も実装可能
- あまり良くないところ
 - 一般的に、開発環境の構築がメンドクサイ
 - UNIX系とWindowsとで、構築方法がゼンゼン違う

- Windows向けですが、誰でも超超超カンタンにPHP拡張モジュールを開発できる環境があります！

<用意するもの>

- Delphi for Win32
 - Delphi 5 ~ Delphi 2009
- php4delphi
 - Delphi for Win32 向けのフリーのプラグイン

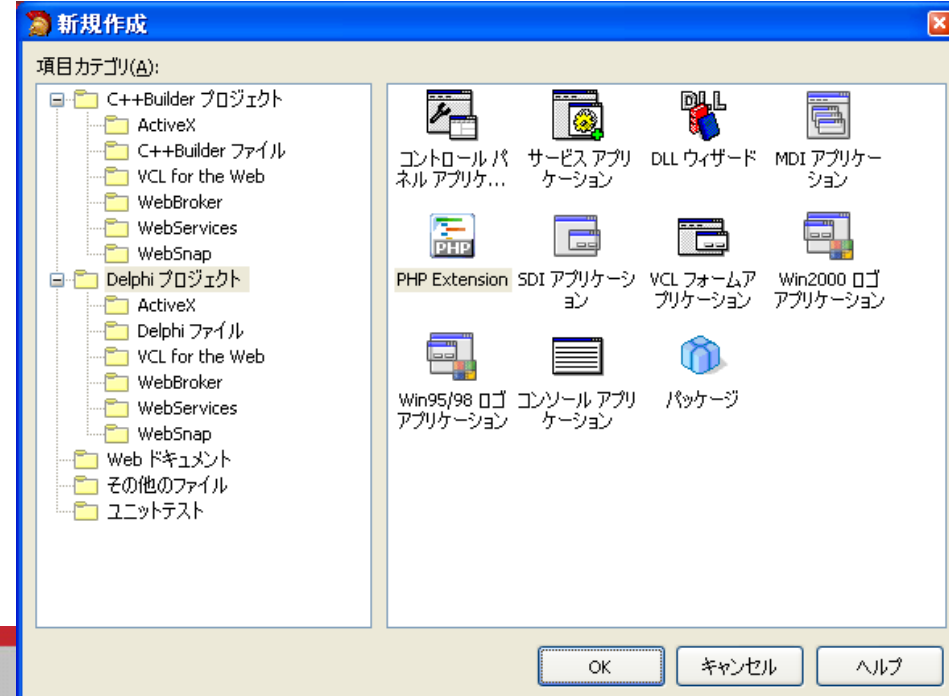
<PHP環境>

- Delphi for PHP - PHPアプリケーションの開発・デバッグ
- WAMP / WIMP - 配布環境

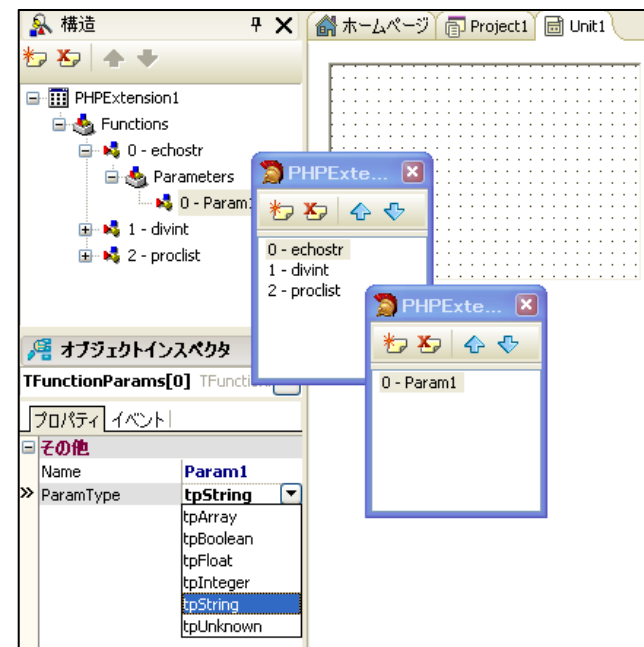
- Delphi for Win32 をインストール
 - 例: Delphi 2007
- php4delphiをダウンロードし、展開
 - <http://cc.embarcadero.com/Item/26233>
 - 例: "C:¥CodeGear¥RAD Studio¥5.0¥php4delphi"
- php4delphiをIDEに統合
 - プロジェクトを開く
 - 例: "C:¥CodeGear¥RAD Studio¥5.0¥php4delphi¥php4Delphi2007group.groupproj"
 - php4DelphiR2007.bpl, php4DelphiD2007.bpl をビルド
 - php4DelphiD2007.bpl をインストール

これだけ!!

- プロジェクトの新規作成
 - [ファイル]-[新規作成]-[その他]-[Delphi プロジェクト]-[PHP Extension]
 - [プロジェクトオプション]-[ディレクトリ/条件-検索パス]を設定
 - 例: "C:¥CodeGear¥RAD Studio¥5.0¥php4delphi"
- 関数の定義と実装を行う...
- ビルド
 - xxxx.dll (例: Project1.dll)を生成
 - PHP環境に配布



- echostr関数
 - 動作 → 文字列を受け取り、その文字列を返す
- TPHPFunctionを1つ追加
 - TPHPExtensionクラスのFunctionsプロパティ
 - FunctionNameプロパティ → echostr
- TFunctionParamを1つ追加
 - TPHPFunctionクラスのParametersプロパティ
 - ParamTypeプロパティ → tpString
- OnExecuteイベントを追加



```
procedure TPHPExtension1.PHPExtension1Functions0Execute(Sender: TObject;  
    Parameters: TFunctionParams; var ReturnValue: Variant;  
    ZendVar: TZendVariable; TSRMLS_DC: Pointer);  
  
var  
    str: AnsiString;  
begin  
    str := AnsiString(Parameters.Items[0].Value);  
    ReturnValue := str;  
end;
```

これだけ!!

● Delphi for PHP

- Project1.dll を “<D4PHP>%php%ext” にコピー
- “<D4PHP>%php%php.ini.template” に extension=Project1.dll を追加
- 実行時に php.ini が更新される
 - [Windows2000, WindowsXP]
 - C:%Documents and Settings%ユーザ名%Application Data%CodeGear%Delphi for PHP%2.0%php%php.ini
 - [Vista]
 - C:%Users%ユーザ名%AppData%Roaming%CodeGear%Delphi for PHP%2.0%php%php.ini

● コマンドライン

- php.exeコマンドの -cオプションで php.ini のフォルダを指定
- 例
 - php.exe -c "C:%Users%codegaer%AppData%Roaming%CodeGear%Delphi for PHP%2.0%php" c:%tmp%test.php

- php.exeを実行して、DLL内のブレークポイントで止める
 - 日本語文字列は？
- プロジェクトオプションを設定
 - [リンカ]-[リモートデバッグシンボルを含める] → ON
 - [デバッガ]-[ホストアプリケーション] → <D4PHP>%php%php.exe
 - [デバッガ]-[パラメータ] →
-c "C:%Users%codegaer%AppData%Roaming%CodeGear%Delphi for PHP%2.0%php" c:%tmp%test.php
 - [デバッガ]-[ソースパス] → .pas のある場所
- ビルド
 - DLL(Project1.dll)とデバッグシンボル(Project1.rsm)を
“<D4PHP>%php%ext”にコピー

これだけ!!

```
procedure TPHPExtension1.PHPExtension1FunctionsOE  
Parameters: TFunctionParams; var ReturnValue: Va  
TSRMLS_DC: Pointer);  
var  
str: AnsiString;  
wstr: WideString;  
50 begin  
str := AnsiString(Parameters.Items[0].Value);  
wstr := UTF8Decode(UTF8String(str));  
53 wstr := '日本語' str;  
end;
```

- divint関数
 - 整数値を2つ受け取り、割り算した商を返す
- TFunctionParamを2つ追加
 - TPHPFunctionクラスのParametersプロパティ
 - ParamType → tpInteger

```
procedure TPHPExtension1.PHPExtension1Functions0Execute(Sender: TObject;  
                Parameters: TFunctionParams; var ReturnValue: Variant;  
                ZendVar: TZendVariable; TSRMLS_DC: Pointer);  
var  
    c: Integer;  
begin  
    c := Integer(Parameters.Items[0].Value) div Integer(Parameters.Items[1].Value);  
    ReturnValue := c;  
end;
```

- ゼロ除算はどうなる？
 - echo divint(10, 2); → 5
 - echo divint(10, 0); → ?

proclist関数

- 配列(array)を返す
 - サーバー内のプロセス名一覧を返してみます
- Booleanパラメータを受け取る
 - True / False → TrueならプロセスIDも追加する
- TFunctionParamを1つ追加
 - TPHPFunctionクラスのParametersプロパティ
 - ParamType → tpBoolean

<注意点>

- 配列を返すには、Zend API を利用する
 - `_array_init`, `add_next_index_string` など...
 - OnExecuteイベントの「`var ReturnValue: Variant;`」は使用しない!!

proclist関数の実装

```
procedure TPHPEExtension1.PHPEExtension1Functions0Execute(Sender: TObject;
    Parameters: TFunctionParams; var ReturnValue: Variant; ZendVar: TZendVariable; TSRMLS_DC: Pointer);
var
    retval: pzval;
    hSnap: THandle;
    proc: TProcessEntry32;
    pid: Boolean;
    data: AnsiString;
begin
    retval := ZendVar.AsZendVariable;
    _array_init(retval, nil, 0);
    pid := Boolean(Parameters.Items[0].Value);
    proc.dwSize := Sizeof(TProcessEntry32);
    hSnap := CreateToolHelp32SnapShot(TH32CS_SNAPPROCESS, 0);
    try
        if Process32First(hSnap, proc) then
            begin
                data := proc.szExeFile;
                if pid then
                    data := data + '(' + IntToStr(proc.th32ProcessID) + ')';
                add_next_index_string(retval, PAnsiChar(data), 1);
                while Process32Next(hSnap, proc) do
                    begin
                        data := proc.szExeFile;
                        if pid then
                            data := data + '(' + IntToStr(proc.th32ProcessID) + ')';
                        add_next_index_string(retval, PAnsiChar(data), 1);
                    end;
                end;
            finally
                CloseHandle(hSnap);
            end;
        end;
    end;
```

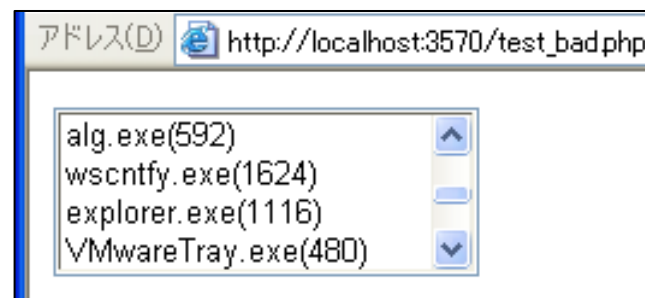


DEVELOPER CAMP

VCL for PHPでカスタムコンポーネントを作成

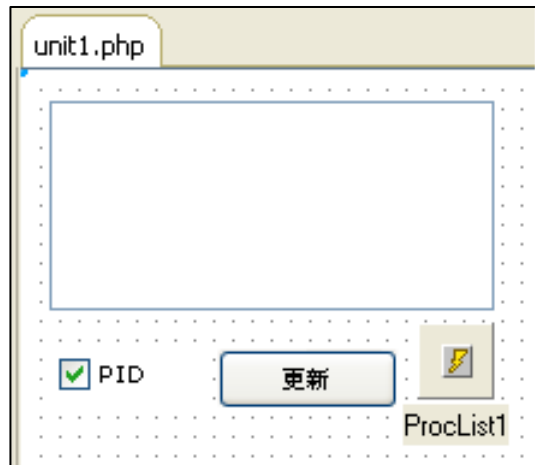
- proclist関数にTrue/Falseを渡して、リストボックスに一覧表示
フツに使うと、リストボックスに関する処理だけで...

```
1 <select name="ListBox1" size="4">
2 <?php
  $pid = $_POST['pid'];
  $list = array();
  if( isset($pid) )
  {
    $list = proclist(true);
  }
  else
  {
    $list = proclist(false);
  }
  $i = 1;
10 foreach ($list as $value) {
  echo "<option value=\"%{$i}%\">{$value}</option>";
  $i++;
}
?>
</select>
```



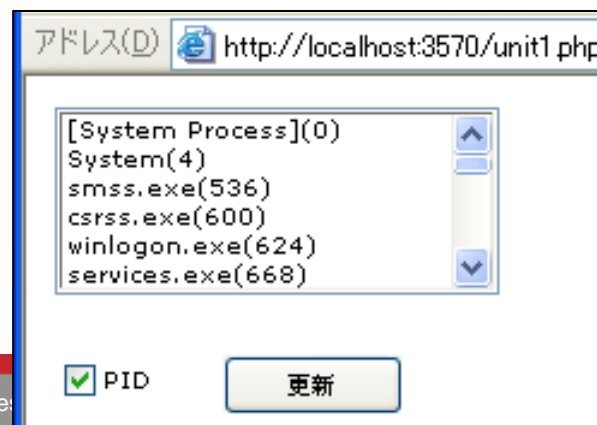
- Delphi for PHP と VCL for PHP を使えば...
 - もっと使い易くて、
 - もっと再利用可能性のある、
 - もっとオブジェクト指向的な、
 - コンポーネント/パッケージが作成できます

- proclistをコンポーネント化すると...
 - 非ビジュアルコンポーネントとして、標準のビジュアルコンポーネントと連携



+

```
public $Proclist1 = null;  
public $CheckBox1 = null;  
public $ListBox1 = null;  
public $Button1 = null;  
function Button1Click($sender, $params)  
{  
    $this->Proclist1->Pid = $this->CheckBox1->Checked;  
    $this->ListBox1->Items = $this->Proclist1->PArray;  
}
```



- メインメニューから[コンポーネント]-[新規コンポーネント]を選択
 - 継承元のクラス → Component
 - クラス名 → ProcList
 - パレットページ名 → Ken
 - パッケージの作成 → ON
- コンポーネント/パッケージは、VCL for PHPの配下に置く
 - コンポーネントの.phpを vcl/ken/proclist.inc.php として保存
 - パッケージの.phpを vcl/ken/ken.package.php として保存
- まずは、ProcListコンポーネントにpublicな読み取り専用プロパティ“PArray”を追加してみよう
 - ```
function readPArray() { return proclist(true); }
```
- コンポーネントをIDEに統合する
  - [コンポーネント]-[パッケージ]-[追加] で ken.package.php を選択

- publishedなプロパティは、IDE上でのフォームデザイン時にも設定可能
    - プロセスIDの表示/非表示を設定するBoolean型プロパティ「Pid」の追加
  - [編集]-[Publishedプロパティの追加]
    - プロパティ名 → Pid
    - デフォルト値 → true
- ```
protected $_pid=true;
function getPid() { return $this->_pid; }
function setPid($value) { $this->_pid=$value; }
function defaultPid() { return true; }
```
- プロパティ"PArray"の実装を変更し、Pidプロパティに連動させる
 - function readPArray() { return proclist(\$this->_pid==true); }
 - PidがBoolean型のプロパティであることをIDEに通知する
 - ken.package.php に registerBooleanProperty('ProcList','Pid'); を追加
 - コンポーネントの変更をIDEに通知する
 - [コンポーネント]-[パッケージ]-[インストール済みパッケージ] で ken.package.php の選択をON→OFF→ON



EMBARCADERO
TECHNOLOGIES®

DEVELOPER CAMP

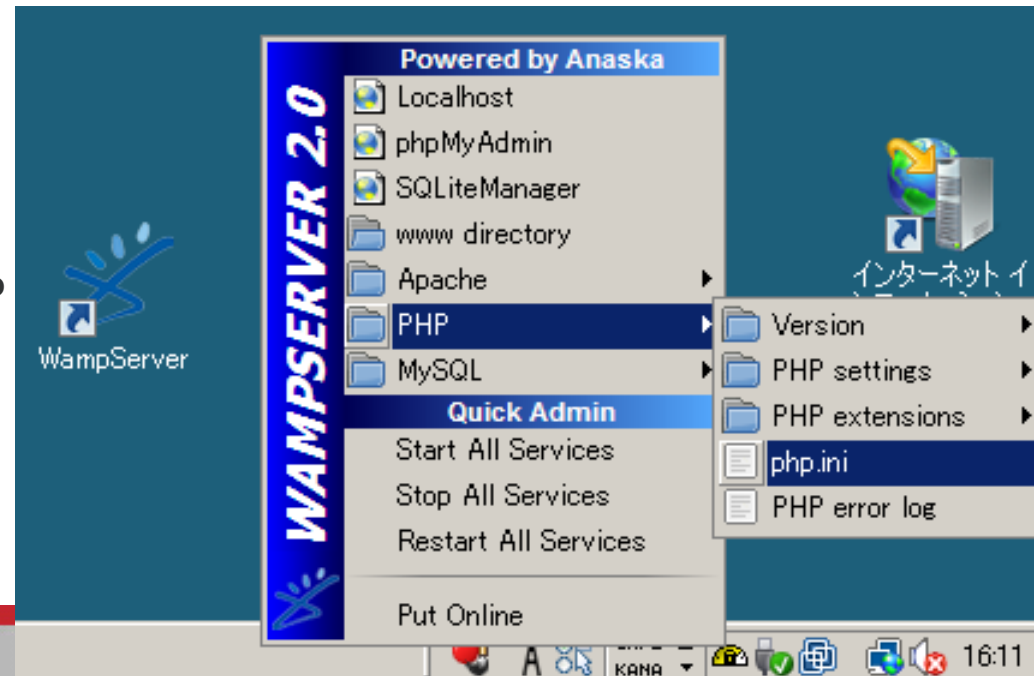
WAMP環境の構築

- 6/23の時点で(私が知っている)利用可能な、WAMP環境
 - WampServer 2.0h
 - <http://www.wampserver.com/en/>
 - PHP 5.2.9-2
 - Apache 2.2.11
 - MySQL 5.1.33
 - XAMPP for Windows 1.7.1
 - <http://www.apachefriends.org/en/xampp-windows.html>
 - PHP 5.2.9
 - Apache HTTPD 2.2.11
 - MySQL 5.1.33
 - WAMPStack 1.1-2
 - <http://bitnami.org/stack/wampstack>
 - PHP 5.2.8
 - Apache to 2.2.11
 - MySQL 5.1.30

- WampServer 2.0h **オススメ!**
 - PHPが常に最新!!
 - とても使い易いサービス管理用のGUIツール有り
 - サービス名として「wampapache」「wampmysqld」を使用
- XAMPP for Windows 1.7.1
 - PHPがちょっと古い
 - サービス管理用のGUIツール有り
 - サービス名として「Apache2.2」「mysql」を使用
 - 1.7.1のインストーラには、my.cnf,my.iniの読み込み設定に不具合あり
- WAMPStack 1.1-2
 - PHPがちょっと古い
 - サービス管理用のGUIツール無し
 - サービス名として「wampstackApache」「wampstackMySQL」を使用

- Windows Server 2008 SP2 上の WampServer 2.0h で試す
 - WampServer 2.0h のインストール
 - DLL(例: Project1.dll)を C:¥wamp¥bin¥php¥php5.2.9-2¥ext にコピー
 - php.ini の設定(パスに注意!!)
 - C:¥wamp¥bin¥apache¥Apache2.2.11¥bin¥php.ini
 - extension=Project1.dll

- サービス(Apache)の再起動
- PHPアプリの転送
 - サブフォルダにVCL for PHP
 - コンポーネントは vcl/ に



- IIS で PHP を実行するには？
 - **FastCGI + 非スレッドセーフ**なPHPエンジン(32bit) によるハイパフォーマンスなPHP環境の構築が機能
 - php-cgi.exe - FastCGIで再利用されるプロセスモジュール
 - **php5.dll** - 非スレッドセーフなPHPコアモジュール
 - Windows Server 2003 の IIS6.0
 - FastCGI拡張のセットアップを手動で...
 - Windows Server 2008 の IIS7.0
 - ハンドラマッピングでカンタンセットアップ **オススメ!**
- た・だ・し、php4delphiで作成した拡張モジュールは、**スレッドセーフ**なPHPコアモジュール「**php5ts.dll**」を参照するため、IISでの利用には注意が必要！
 - スレッドセーフ版のPHPエンジンを使用します
 - php4delphiを書き換えれば対応は可能でしょう



EMBARCADERO
TECHNOLOGIES®

DEVELOPER CAMP


WIMP 環境の構築

● WI?P環境の構築手順

- Windows Server 2008 SP2 を用意する
- PHP5のインストール
 - スレッドセーフ版の“php-5.2.x-Win32.zip”をダウンロードし、C:¥PHPに展開
 - <http://www.php.net/downloads.php>
 - php.iniの編集
- DLL(例: Project1.dll)を C:¥PHP¥ext にコピー
- IIS7.0 のインストール
 - [コントロールパネル]-[管理ツール]-[サーバー マネージャ]-[役割]-[役割サービスの追加]-[Webサーバー]-[アプリケーション開発]-[CGI]
- IISマネージャでFastCGIのハンドラマッピングの追加
 - [ハンドラマッピング]-[モジュールマップの追加]
 - 要求パス - *.php
 - モジュール - FastCGIModule
 - 実行可能ファイル - C:¥PHP¥php-cgi.exe
- IISマネージャでアプリケーションプールの追加

1.

役割サービスの追加



役割サービスの選択

Web サーバー (IIS) にインストールする役割サービスを選択:
役割サービス(R):

- Web サーバー (インストールされています)
 - HTTP 基本機能 (インストールされています)
 - アプリケーション開発 (インストールされています)
 - ASP.NET
 - .NET 拡張性
 - ASP
 - CGI (インストールされています)
 - ISAPI 拡張
 - ISAPI フィルタ
 - サーバー側インクルード
 - 状態と診断 (インストールされています)
 - セキュリティ (インストールされています)
 - パフォーマンス (インストールされています)
- 管理ツール (インストールされています)
- FTP Publishing Service

2.

モジュール マップの編集

要求パス(P):
*php
例: *bas, wsvc.axd
モジュール (M):
FastCgiModule
実行可能ファイル (オプション)(E):
C:\PHP\php-cgi.exe
名前(N):
PHP under FastCGI
要求の制限(R)...
OK キャンセル

3.

アプリケーション プールの追加

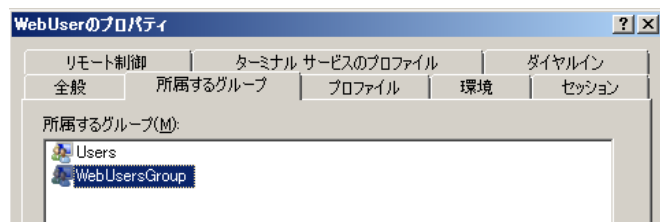
名前(N):
PHPPool
Net Framework バージョン(E):
NET Framework v2.0.50727
マネージ パイプライン モード(M):
統合
 アプリケーション プールを直ちに開始する(S)
OK キャンセル

参考資料(英語): <http://www.microsoft.com/japan/opensource/php/learning/default.msp>

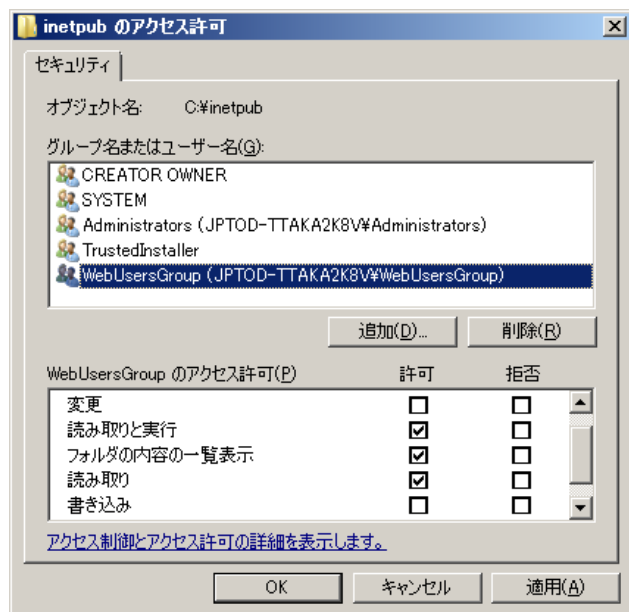
● IIS向けWebサイトの構築手順

- Webアプリケーション実行専用のユーザー&グループの追加
 - 例: グループ - WebUsersGroup
 - 例: ユーザー - WebUser
- Webサイト用のフォルダを作成
 - アクセス許可を特定のユーザーグループに制限する
- IISマネージャで[Webサイトの追加]
 - アプリケーションプールの選択
 - 物理パスの選択
 - [接続]-[パス資格情報]-[特定のユーザー] → WebUser
 - [テスト接続]ボタンでアクセスの可否を確認
- IISおよびWebサイトの(再)起動
- PHPアプリの転送
 - サブフォルダにVCL for PHP
 - コンポーネントは vcl/ に

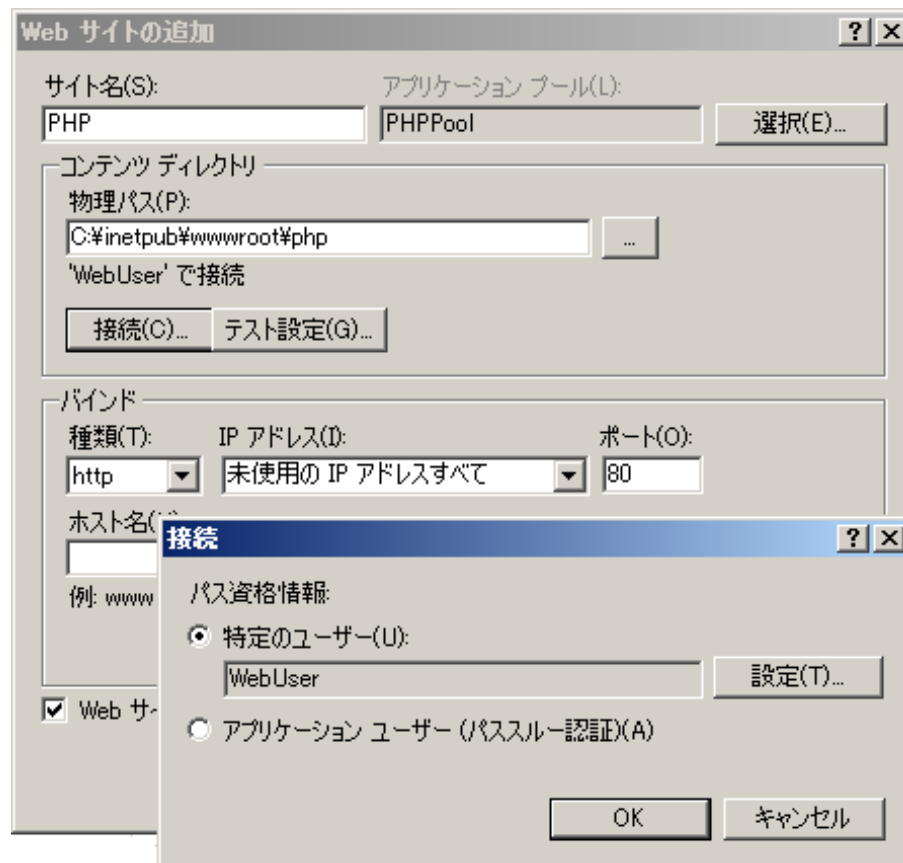
1.



2.



3.



参考資料(英語): <http://www.microsoft.com/japan/opensource/php/learning/default.msp>

- Windows版 PHP5向けの拡張モジュールの作成
 - .DLLを作成すれば、PHP環境を強化できる
 - Delphi for Win32 を使えば、とてもカンタンに作成できる
- PHP関数のカプセル化
 - コンポーネント化することで、開発を効率化できる
 - Delphi for PHP / VCL for PHP の強力なフレームワークとの連携
- WAMP / WIMP 環境の構築
 - Apacheベースのスタックを使えば、素早くWebサイトを構築可能
 - IISベースのWebサイト構築は基本的に手動だが、FastCGIが利用可能

ご静聴ありがとうございました

Q&A