

【B5】PHPテクニカルセッション



DEVELOPER CAMP

Delphi for PHP で作るリッチコンテンツブログ

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

アジェンダ



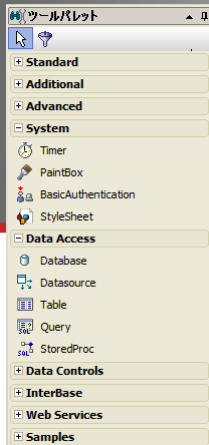
DEVELOPER CAMP

- コンポーネントをフル活用しよう
- お馴染み、データモジュール
- Blogデータの表示用ページ
- Blogデータの登録用ページ

The screenshot shows a Delphi IDE window titled "EditUnit". On the left, there is a calendar control for August 2008. The main area contains a text area with the following text:

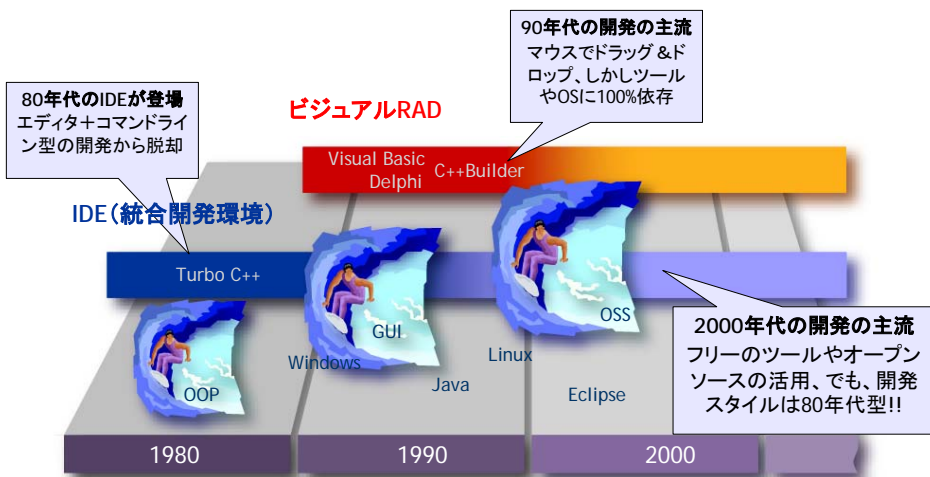
高橋(智)です。少し宣伝させていただきます。
カスタムメイドのDelphi for PHPの事が話題されます。あと1週間程度で公開に差し掛か
ります。PHP言語の学習や、Webアプリケーションの開発やシステム構築の専門家としてお
話します。PHPに興味がありましたら、ぜひ行楽のう(ア)ル版で読んでみてください。
[Delphi for PHPで仕上げるPHPウェブ開発]
http://www.embarcadero.com/Books/2726-4-07263_202_5.html

At the bottom, there is a "Save" button.



コンポーネントをフル活用しよう

開発環境の進歩と退化



- Webデザイン
 - WYSWYGであること
- AJAXが簡単に利用できる
 - あくまでも目的ではなく手段
- オープンソースを簡単に活用・再利用できる
 - プロプライエタリな世界に留まっていたはいけない
- コンポーネントアーキテクチャ
 - 重厚長大ではいけない、しかもオープンソースを活用して

こうした点に配慮しつつ、90年代型の開發生産性を
Web開発にも、もたらすことが重要



VCL for PHP

- 今回利用する VCL for PHP の コンポーネント



お馴染み、データモジュール

データベースのセットアップ

- RDBMSには、MySQL 5.0 を利用
- テーブル構造: create table blog(

ID	INTEGER	NOT NULL	AUTO_INCREMENT,
BLOGDATE	VARCHAR(10)	NOT NULL,	
TITLE	VARCHAR(255)	NOT NULL,	
ARTICLE	TEXT	NOT NULL,	

PRIMARY KEY(ID),
UNIQUE KEY(BLOGDATE));
- 主キーは、自動インクリメントを利用する
 - ADOdbをラップしたTableコンポーネントでは、行追加(INSERT)時に主キーを明示的に指定しなくてよい仕様
- データベース/テーブルの文字コードは utf8

The screenshot shows an IDE with several windows. At the top, there's a 'データエクスプローラ' (Data Explorer) window showing a tree view of databases: DB2, InterBase, MySQL, and MYSQLTEST. Under MYSQLTEST, there are 'Tables' (app, blog) and 'Indices' (ARTICLE, BLOGDATE, ID, TITLE, boottest). A red arrow points from the 'blog' table in the Data Explorer to the 'tbblog1' object in the 'オブジェクトインスペクタ' (Object Inspector) window. Below the Data Explorer, there are two 'オブジェクトインスペクタ' windows. The left one is for 'dbriken1 Database' and the right one is for 'tbblog1 Table'. Both have red exclamation mark icons next to their titles.

Database

- CharSet
- Connected

Table

- Active
- LimitCount

- DatabaseコンポーネントのOnAfterConnectイベント
 - DB接続を utf8 にする
- データモジュールのインスタンスを返す関数
 - function GetDBModule() {...}
 - \$global を忘れずに
- Blogデータを保存(INSERT or UPDATE)するメソッド
 - public function SaveBlog(\$date, \$title, \$article) {...}
 - GetDBModule()->SaveBlog('2008/8/20', 'タイトル', '<HTML>...');
- Blogデータを取得(SELECT)するメソッド
 - public function LoadBlog(\$date) {...}
 - \$blog = GetDBModule()->LoadBlog('2008/8/20');
- ある月のBlog日の日付の配列を返すメソッド
 - public function GetBlogDays(\$year, \$mon) {...}
 - \$blogdays = GetDBModule()->GetBlogDays('2008', '8');

Database, Table - オープンソースライブラリ「ADOdb」をコンポーネント化

```
function dbriken!AfterConnect($sender, $params)
{
    $this->dbriken!->execute("set character set utf8");
}

public function SaveBlog($date, $title, $article)
{
    $this->tbblog!->LimitCount = 1;
    $this->tbblog!->Filter = "BLOGDATE=" . GetDBModule()->dbriken!->QuoteStr($date);
    $this->tbblog!->open();
    if( $this->tbblog!->RecordCount == 0 )
        $this->tbblog!->append();
    $this->tbblog!->BLOGDATE = $date;
    $this->tbblog!->TITLE = $title;
    $this->tbblog!->ARTICLE = $article;
    $this->tbblog!->post();
    $this->tbblog!->close();
}

public function LoadBlog($date)
{
    $title = "";
    $article = "";
    $this->tbblog!->LimitCount = 1;
    $this->tbblog!->Filter = "BLOGDATE=" . GetDBModule()->dbriken!->QuoteStr($date);
    $this->tbblog!->open();
    if( $this->tbblog!->RecordCount == 1 )
    {
        $title = $this->tbblog!->TITLE;
        $article = $this->tbblog!->ARTICLE;
    }
    $this->tbblog!->close();
    return array($title, $article);
}

public function GetBlogDays($year, $mon)
{
    $blogdays = array();
    $this->tbblog!->LimitCount = 31;
    $like = $year . "/" . $mon . "/%";
    $this->tbblog!->Filter = "BLOGDATE LIKE " . GetDBModule()->dbriken!->QuoteStr($like);
    $this->tbblog!->open();
    while( !$this->tbblog!->EOF )
    {
        array_push($blogdays, $this->tbblog!->BLOGDATE);
        $this->tbblog!->next();
    }
    $this->tbblog!->close();
    return $blogdays;
}
```

Copyright ©2008 Embarcadero Technologies, Inc.
 本文書の一部または全部の転載を禁止します。

1



Unit1

2008/8/20

書籍: Delphi for PHPではじめるPHPビジュアルWeb開発

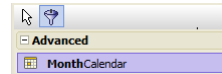
高橋(管)です。少し遅らせてください。
 カオシステムさんからDelphi for PHPの本が出版されます。あと1週間程
 まで後押ししたいと思います。PHP言語の基礎学習や、マスターページを利用し
 たブログエントリーの作り方を解説されています。PHPに興味がありましたら、ぜひ
 村原のライバル本を買って読んでください。
 [Delphi for PHPではじめるPHPビジュアルWeb開発]
<http://www.cutt.co.jp/book/978-4-87783-202-5.html>

EMBARCADERO TECHNOLOGIES.
 DEVELOPER CAMP

Blogデータの表示用ページ

• MonthCalendar

- オープンソースライブラリ「JSCalendar」をコンポーネント化
- OnUpdate Javascriptイベント
 - 日付をクリックしたタイミング
- OnBeforeShow イベント
 - カレンダーの表示形式をカスタマイズ



• 今回、Javascriptコードを記述しますが、MonthCalendarコンポーネントを継承した拡張コンポーネントを作成して隠蔽すれば、再利用可能です

```

echo "<script type='text/javascript'>\n";
echo "var blogdays = [];\n";
foreach ($blogdays as $v) {
    echo "    '$v', 'v',";
}
echo "\n";
echo "function avDateStatusFunc(date) {\n";
echo "    var chk = date.getFullYear()+"/"+(date.getMonth()+1)+"/"+date.getDate();\n";
echo "    for(var i in blogdays) if(blogdays[i]==chk) return false;\n";
echo "    return true;\n";
echo "}\n";
echo "</script>\n";
}

function ViewUnitCreate($sender, $params)
{
    $this->MonthCalendar1->calendar->set_option("dateStatusFunc", "avDateStatusFunc");
}

function MonthCalendar1$Update($sender, $params)
{
    ?>
    var y = event.date.getFullYear();
    var m = event.date.getMonth() + 1; // integer, 0..11";
    var d = event.date.getDate(); // integer, 1..31";
    window.location = "ViewUnit.shtm?View=" + y + "&M=" + m + "&D=" + d;
    return false;
}
</?php
    
```

• Label x 3個

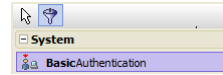
- 日付
- タイトル
- HTMLデータ

Copyright ©2008 Embarcadero Technologies, Inc. All Rights Reserved.
本文書の一部または全部の転載を禁止します。

The screenshot shows a web browser window with a calendar for August 2008 on the left. The main content area displays a registration page for a blog. The page title is "Delphi for PHPで記述するPHPレジスタールWw". The page content includes a message in Japanese: "高機能です。少し宣伝させていただきます。カオスシステムからDelphi for PHPの本が出版されます。あっと驚かされる内容で是非読んでください。PHP関連の書籍や、ブログページも利用したブログエントリの作成が楽々です。PHPの書籍がほしい。ぜひお探しのページで検索してください。" followed by a link: "[Delphi for PHPで記述するPHPレジスタールWeb版] http://www.out.co.jp/Book/978-4-897763-202-9.html". Below the text is a "Path:" field and a "保存" (Save) button.

• BasicAuthentication

- 「BASIC認証」をコンポーネント化
- ErrorMessageプロパティ
- OnAuthenticate イベント
 - 戻り値 true or false
- Execute() メソッド
 - ページのOnCreateイベントで呼び出す
 - ページのOnShowイベントなどでは遅すぎる!! なぜ?



```

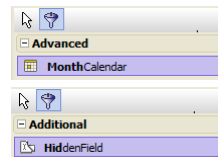
function BasicAuthentication1Authenticate($sender, $params)
{
    if (($params['username'] == 'ken') && ($params['password'] == 'password'))
        return true;
    else
        return false;
}

function EditUnitCreate($sender, $params)
{
    $this->BasicAuthentication1->Execute();
}
    
```

• MonthCalendar - Blogデータの表示用ページと同じ

• HiddenField

- HTMLの<input type="hidden" ...>をコンポーネント化
- 編集用のページの日付をHTMLに埋め込む
- [保存]ボタンをクリック時に、編集するBlogの日付をサーバ側で取得



```

$this->HiddenField1->Value = $editdate;
$this->MonthCalendar1->_calendar->set_option('dateStatusFunc', 'myDateStatusFunc');
}

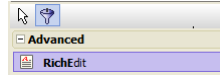
function EditUnitShow($sender, $params)
{
    $editdate = $this->HiddenField1->Value;
    $this->MonthCalendar1->Date = $editdate;
    $this->Label1->Caption = $editdate;
    $blog = GetDBModule()->LoadBlog($editdate);
    $this->Edit1->Text = $blog[0]; // title
    $this->RichEdit1->Text = $blog[1]; // article
    // $this->RichEdit1->Lines = array($blog[1]); // 上と同じ
}

function Button1JSClick($sender, $params)
{
    echo $this->Button1->ajaxCall("SaveBlog");
    return false;
}

function SaveBlog($sender, $params)
{
    $editdate = $this->HiddenField1->Value;
}
    
```


- RichEdit

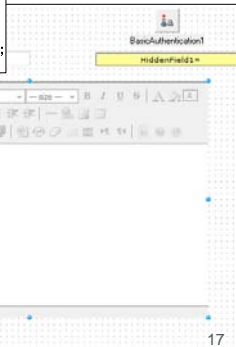
- オープンソースライブラリ「Xinha」をコンポーネント化
- HTMLデータを編集可能
- Textプロパティで、HTMLデータを文字列として取得



```

$blog = GetDBModule()->LoadBlog($editdate);
$this->Edit1->Text = $blog[0]; // title
$this->RichEdit1->Text = $blog[1]; // article
// $this->RichEdit1->Lines = array($blog[1]); // 上と同じ
}

function SaveBlog($sender, $params)
{
    $editdate = $this->HiddenField1->Value;
    GetDBModule()->SaveBlog($editdate, $this->Edit1->Text, $this->RichEdit1->Text);
}
    
```



- Label - 日付
- Edit - タイトル

[保存]ボタンをAjax対応にするには

- VCL for PHPは、オープンソースライブラリ「Xajax」を内蔵
- Button の ButtonTypeプロパティを btNormal にする
- Button の OnClickイベントは使わない!!
- Button の OnClick Javascriptイベントを使う!!
 - echo で Ajax用のJavaScriptコードを出力
 - echo \$this->コンポーネント->ajaxCall("メソッド名");
- Page の UseAjaxプロパティを true にセット

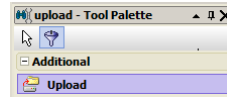
```

function Button1JSClick($sender, $params)
{
    echo $this->Button1->ajaxCall("SaveBlog");
    return false;
}
<?php

function SaveBlog($sender, $params)
{
    $editdate = $this->HiddenField1->Value;
    GetDBModule()->SaveBlog($editdate, $this->Edit1->Text, $this->RichEdit1->Text);
}
    
```

- Uploadコンポーネントをページに配置してみる

- <input type="file" ...> をコンポーネント化



- しかし...

- ButtonのOnClickイベントであれば、ファイルデータを取得できるが
 - Ajaxを使うと、ファイルデータは送信されない!!

- 一般的なデザインでは...

- 画像ファイルのアップロード専用のページを作成
 - インラインフレームを使って、画面の一部として表示

```
<iframe id="KenIFrame1_iframe" src="UploadUnit.php" width="185" height="41"></iframe>
```

- VCL for PHP には iframe をラップしたコンポーネントは無い

- 簡単にカスタムコンポーネントを自作できます!!
 - QC#61977 - <http://qc.codegear.com/wc/qcmain.aspx?d=61977>

```
<?php
require_once("vcl/vcl.inc.php");
use_unit("stdCtrls.inc.php");
class KenIFrame extends CustomControl {
function __construct($owner = null) {
parent::__construct($owner);
}
function dumpContents() {
if( ($this->ControlState & csDesigning) != csDesigning ) {
echo "<iframe id='".$this->name_iframe.'" src='".$this->url.'"";
echo " width='".$this->width.'" height='".$this->height.'">";
}
}
else {
echo " '".$this->url.'"";
}
}
protected $_url="";
function getUrl() { return $this->url; }
function setUrl($value) { $this->url=$value; }
function defaultUrl() { return ""; }
}
?>
```

The screenshot shows a VCL IDE environment. On the left, a code editor displays PHP code for a custom component class named KenIFrame. The code includes a constructor, a dumpContents method that outputs an iframe tag with dynamic attributes, and protected properties and methods for URL management. In the center, a 'keni - Tool Palette' window shows the KenIFrame component. On the right, a preview window displays the rendered output of the component, which is an orange-themed form with an 'Upload' button and a 'BasicAuthentication1' label. A blue arrow points from the tool palette to the preview window, indicating the component's visual representation.