Phalanger

Compiling PHP for .NET and Silverlight

http://php-compiler.net

Tomáš Petříček, C# MVP (tomas@tomasp.net)

Formerly by: Tomáš Matoušek, Ladislav Prošek, et al.

Agenda

Phalanger Introduction

- What is Phalanger? And what is it good for?

Phalanger for Silverlight

- How (and why) to write Silverlight client apps in PHP

.NET and PHP interoperability

- Calling C# code from PHP
- Calling PHP code from C#

Interesting Compilation Problems

- Conditional type binding
- Calling overloaded methods

DEMO

Let’s look at some cool demo first :-)!

http://tomasp.net/files/copter/default.html

Introduction

- Phalanger: PHP Language Compiler for .NET
  - Initially targeting .NET Framework 1.1
  - Now using .NET 2.0, Mono and Silverlight 2.0
  - We’re planning to use (parts of) DLR in the future

- What can you do with Phalanger?
  - Compile various open-source PHP applications for .NET
  - Use PHP functions & objects from (other) .NET languages
  - Use .NET classes from PHP programs
  - **Write Silverlight applications in PHP**

Agenda

Phalanger Introduction

- What is Phalanger? And what is it good for?

Phalanger for Silverlight

- How (and why) to write Silverlight client apps in PHP

.NET and PHP interoperability

- Calling C# code from PHP
- Calling PHP code from C#

Interesting Compilation Problems

- Conditional type binding
- Calling overloaded methods

Phalanger: PHP for Silverlight

- Similar as other Silverlight dynamic languages
  - XAML file is loaded by Silverlight plugin in the browser
  - PHP source code referenced using XAML element:
    `<php:PhalangerLoader Source="script.phpx" />`
  - Downloaded, compiled and executed on the client
  - “Loader” element, compiler and runtime in one DLL file

```
<Canvas xmlns="http://schemas.microsoft.com/client/2007"
        xmlns:php="clr-namespace:PHP.Silverlight;
                   assembly=Bin/PhpNetCore.dll"
        Width="800" Height="600" Background="White">
    <php:PhalangerLoader Source="simpledemo_php.phpx" />
</Canvas>
```

Phalanger: PHP for Silverlight

- Simple PHP example ("script.phpx")
  - Creating XAML elements from PHP programmatically

```
<?
include("slutils.phpx");

$canv = slcreate("Canvas", array(
    "Canvas.Name" => "smiley"),
    array(slcreate("Ellipse", array(
            "Width" => 100, "Height" => 100,
            "Fill" => "#FFFF00")),
          slcreate("Image", array(
            "Source" => "Smiley.png"))),
    $canvas);
?>
```

DEMO

Simple Silverlight Demo

What do you need to write Silverlight app in PHP?

Phalanger: PHP for Silverlight

- What makes PHP a great language for Silverlight?

```
// Move smiley to random location
$canvas->smiley->{ "Canvas.Left" } = rand(100, 600);
```

- Dynamic access to XAML elements
    - Provided by `CanvasWrapper` using __`get` function
- Nice way for working with *attached properties*
    - Properties like `Canvas.Left` are not part of the object
    - In PHP they can be read/set using ordinary property syntax

Phalanger: PHP for Silverlight

- What makes PHP a great language for Silverlight?

```
// Sending HTTP web request
$fh = fopen(sl_mkabsolute("simplecallback.php"), 'r');

// Working with the IsolatedStorage
$fh = fopen("score.txt", 'r');
```

- Subset of standard PHP functions is available
  - Make various common tasks very easy
  - Consistent with the server-side library
  - Can be extended by creating managed extensions

DEMO

Silverlight Photo Gallery

Developing Silverlight PHP apps in Visual Studio 2008

Agenda

Phalanger Introduction

- What is Phalanger? And what is it good for?

Phalanger for Silverlight

- How (and why) to write Silverlight client apps in PHP

.NET and PHP interoperability

- Calling C# code from PHP
- Calling PHP code from C#

Interesting Compilation Problems

- Conditional type binding
- Calling overloaded methods

Using .NET libraries from PHP

- What is needed to use .NET classes?
- Support for namespaces and generics
  - Syntax for namespaces will change to be compatible with PHP 5.3

```
import namespace System:::Collections:::Generic;
$d = new Dictionary<:string, string:>;
$d->Add("en", "Hello world!");
$d->Add("hx", 42);              // OK – implicit conversion!
```

- Support for creating .NET delegates

```
// add a few elements to the dictionary:
$this->MouseLeftButtonDown->Add(
    new MouseEventHandler(array($this, "MouseDown")));
```

- Overloaded methods – will get to them soon!

Calling PHP code from C#: “Pure” mode

- Compiler produces useable .NET classes

```
// PHP class in “pure” mode
[Export]
class Texy {
    public function parse($text) { /* ... */ }
}
```

```
// C# signature of the type
class Texy {
    object parse(object text);
}
```

- Some restrictions to the language are needed
  - C#-like visibility – all files compiled at once
  - No inclusions or global code is allowed
  - Static “Main” method used as an entry-point

Calling PHP code from C#: “Duck typing”

- How can we call a PHP object from C#?

```
// PHP class declaration
class SampleObj {
  function Add($a, $b) {
    return $a + $b;
  }
  function NewRandom() {
    return new System:::Random();
  }
}
```

- PHP classes are compiled into .NET classes
  - The signature is incompatible with C#
  - The class may not exist at compile-time

Calling PHP code from C#: “Duck typing”

- Declare a signature of the type using interface:

```
[DuckType]
public interface ISampleObj {
    int Add(int p1, int p2);
    double Add(double p1, double p2);
    Random NewRandom();
}
```

- Calling PHP object “SampleObj” via the interface
  - Phalanger creates an implementation of the interface
  - Necessary type conversions are inserted

```
ISampleObj so = ctx.NewObject<ISampleObj>("SampleObj");
int r1 = so.Add(24, 18);        // returns 42
double r2 = so.Add(24.5, 17.9); // returns 42.4
Random rnd = so.NewRandom();
```

DEMO

Duck Typing Interop

Reading XML data using PHP “SimpleXml” extension.

Agenda

Phalanger Introduction

- What is Phalanger? And what is it good for?

Phalanger for Silverlight

- How (and why) to write Silverlight client apps in PHP

.NET and PHP interoperability

- Calling C# code from PHP
- Calling PHP code from C#

Interesting Compilation Problems

- Conditional type binding
- Calling overloaded methods

Conditional Type Bindings

- Conditional declaration of classes:

```
if ($test) {
  class A { /* version 1 */ }
} else {
  class A { /* version 2 */ }
}
```

  - Compiled as two .NET classes (`A$1`, `A$2`)
  - Later, one ‘version’ is added to the list of declared types

- Problem with unknown base type:

```
class B extends A { /* ... */ }
```

  - Compilation is delayed until execution
  - Generates code calling PHP “eval” function

Calling CLR Overloaded Methods

- Type not known at compile time:

```
void WriteLine(int value);
void WriteLine(string value);
/* ... */
```

```
$value = "Hello world!";
Console::WriteLine($value);
```

- Generate a ‘stub’ to select the overload at run-time:
  - Select the first option matching the implicit conversion rules
  - First attempt:

```
The result is:
0
```

  - Implicit conversion **string -> int** exists!

Calling CLR Overloaded Methods

- Better approach:
  - Rate quality of the conversion
    (Exact match, Precision loss, Changing “domain”, Explicit, Failure)
  - Select the best option using quality of conversion
  - More complex, but gives the expected results!

- What about .NET 2.0 nullable types?

```
$o->DoubleNull = 3.14159;      // Assignment: double? = double
$o->IntNull = $o->DoubleNull; // Assignment: int? = double?
```

  - Conversion rules for converting to (and from) nullable types
  - Rules for converting between different nullable types

Thanks!
Questions & Answers?

More Information

Phalanger Web Site: http://php-compiler.net

Source Code: http://www.codeplex.com/Phalanger

Discussion @ PHP.Net

http://news.php.net/php.on.dlr

My Blog: http://tomasp.net/blog

Contact: tomas@tomasp.net