

Forensic Development

- Scott Koon
- Fred Hutchinson Cancer
Research Center
- www.lazycoder.com

Forensic Development

- What does that mean?
- Figuring out code you didn't write.

Where does this code come from?

- Legacy code
 - “Code that works”
- Forced on you
 - Off the shelf
 - Secretary Special

Finding out how the application works

- If you have the source code
- If you don't have the source code

If you have the source code

- Just start reading
 - ASP.NET, JSP – remember you're only getting 1/2 the story.
 - Reading code is hard

If you have the source code

Reverse engineer to UML

 View the code in the object browser

If you have the source code

- Run the app
 - Step through in the debugger
 - Watch lists
 - Lots of breakpoints
 - Trace one variable of interest

If you don't have the source code

- .NET code

- .NET Reflector - <http://www.aisto.com/roeder/dotnet/>

- ILDASM

- Java code

- JODE - <http://jode.sourceforge.net/>

- Mocha - <http://www.brouhaha.com/~eric/software/mocha/>

- C/C++

- Assembly

If you don't have the source code

- Obfuscated code
 - Garbage in the method signatures
 - Start passing in odd strings
 - It might not be obfuscated, it might not be English.

Finding out how the app works

- Try to break it.
 - QA knows more about how your code works than you do.
 - Find your bad input in the apps output.
 - Helps you trace a path through the app.

Finding out how the app works

- Data pump applications
 - SQL Profiler – live it, learn it.
 - Tip - Filter DB name
 - The “Events” and “Filters” tabs are your best friends.

Finding out how the app works

- SQL Profiler – cont.
 - Helpful events
 - Stored Procedures
 - RPC Output Params
 - SP:Starting
 - RPC: Starting
 - TSQL
 - SQL:StmtStarting
 - Errors and Warnings
 - Everything

Finding out how the app works

- Oracle
 - QBQL
 - SQL *Repository

Finding out how the app works

- Error handling and logging
 - ASP.NET trace enabled
 - Server log
 - /var/log
 - Event log
 - Logging tools
 - Log4*

Finding out how the app works

- Web applications
 - Always “view source” as your first action.

How to extend the code

Web Applications

Client side script

Flatten out the DOM differences

```
var isMozilla = !(document.all);
```

```
if(isMozilla) {
```

```
    HTMLElement.prototype.__defineGetter__("innerText", function () {
```

```
        var r = this.ownerDocument.createRange();
```

```
        r.selectNodeContents(this);
```

```
        return r.toString();
```

```
    });
```

```
}
```

How to extend the code

- Web services

- Bridges

- Java -> .NET bridge - JNBridge

- Java -> COM bridge - http://www.alphaworks.ibm.com/tech/dtjcb?open&S_TACT=104AHW61&S_CMP=GR&ca=dgr-dw01awdtjcb

- Java -> C/C++

- JNI

- PHP -> .NET

- Built in

- Ruby -> .NET

How to extend code

Ruby -> .NET

`ArrayList.new # -> an ArrayList`

Ruby event handlers

```
button = Button.new button.click.add do |sender, args|
  puts "Button clicked!"
end
```

`button.performClick # -> prints <<Button clicked!>>`

How to extend code

PHP -> .NET

```
<?php
    $stack = new DOTNET("mscorlib", "System.Collections.Stack");

    $stack->Push(".Net");
    $stack->Push("Hello ");

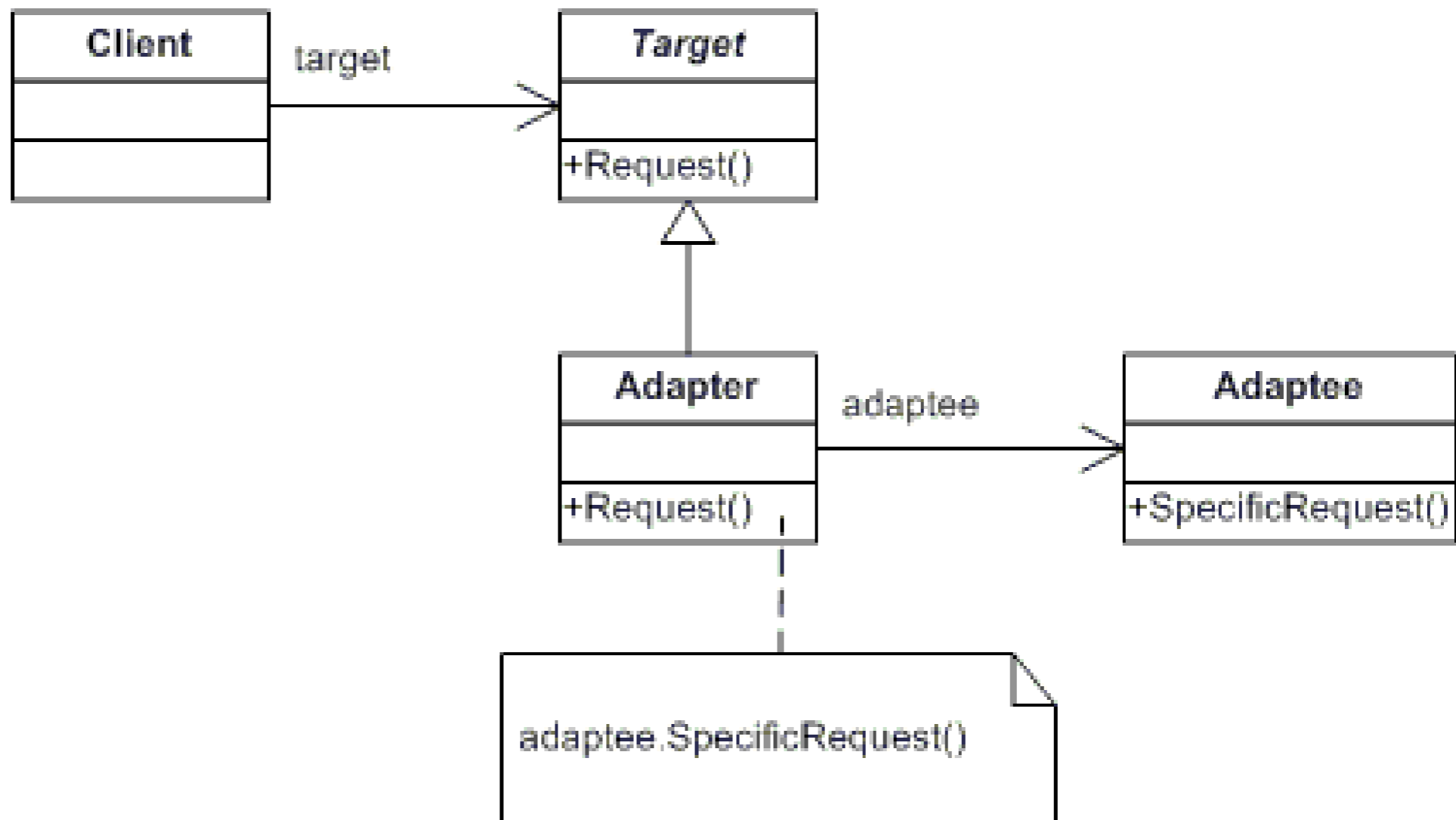
    echo $stack->Pop() . $stack->Pop();
?>
```

How to extend code

- Java -> .NET
 - JNBridge Pro generate .NET proxy classes for Java classes.
 - JNI can be used also.
 - JNI.NET - <http://caffeine.berlios.de/site/>
 - IKVM - <http://www.ikvm.net/>
 - Runtime conversion of bytecode to MSIL

How to extend code

- Adapter pattern
 - Also called the wrapper pattern



Discussion:

- When do you scrap and application and re-write?