

# Migrating ASP.NET Applications



Four ways to convert them  
to SharePoint

Paul Papanek Stork, SharePoint Server MVP,  
MCT, MCSE+I, MCSA, MCSD, MCDBA, MCITP, MCPD  
[Paul.Stork@Mindsharp.com](mailto:Paul.Stork@Mindsharp.com)

# About the Speaker...

- Paul Papanek Stork, MVP, MCT, MCSE+I, MCSA, MCSD, MCDBA, MCITP, MCPD
  - Senior Instructor/Consultant at Mindsharp
  - <http://www.mindsharp.com>
  - [Paul.Stork@mindsharp.com](mailto:Paul.Stork@mindsharp.com)
  - Contributing Author, *Developer's Guide to Windows SharePoint Services 3.0 & Microsoft Office SharePoint Server 2007 Best Practices*
  - Author, upcoming October 2009  
*MCTS: Windows SharePoint Services 3.0 Configuration Study Guide (70-631) by Wiley*



# Agenda

- The Challenge of Legacy Applications
- Four Conversion Methodologies
  - Page Viewer Web Part
  - Run in Virtual Application Directory
  - Uploaded Pages with Separate Class Library
  - Refactor as Connected Web Parts

# The Legacy Application Problem

- Legacy ASP.NET Applications
  - Based on .aspx pages
  - Integrated Development environment
  - Code beside sheets or inline code
  - Code re-use and integration a challenge
- SharePoint Applications
  - Based on Web parts
  - No direct development environment
  - Separate class libraries
  - Web part re-use and integration through connections

# Page Viewer Web Part

- Uses an iFrame embedded inside a Web Part
- Advantages
  - Simplest Conversion
- Disadvantages
  - Authentication may be a challenge
  - Not High Performance
  - Requires maintaining existing servers



MINDSHARP

# Demo

# Demo

## Page Viewer Web Part



# Virtual Application Directory

- Create Virtual Directory in Web Application
- Advantages
  - Can be mapped to multiple zones
  - Requires few changes
  - Can use SharePoint branding
- Disadvantages
  - May require code changes to run

# Creating a Virtual Directory

1. Create a Virtual Directory in IIS manager
2. Remove Application setting
3. Copy pages and code behind files
4. Edit pages to use a customized SharePoint masterpage



MINDSHARP

Demo  
DEMO

# Virtual Application Directory



# Uploaded Pages with Class Library

- Upload ASPX pages to SharePoint Library
- Re-compile Code in Class Library
- Advantages
  - Supports SharePoint Branding
- Disadvantages
  - Code must be modified and recompiled
  - No support for Inline Code
  - Some code changes required

# Creating an Uploaded Page

1. Upload ASPX pages to document library
2. Add code behind file to Class Library project
3. Add Namespace to class
4. Add protected declarations for controls declared on page
5. Remove AutoEventWireup and Codefile
6. Change Inheritance= to point at Class Library
7. Mark class as Safe in Web.Config
8. Adjust Code Access Security settings



# Demo

Demo

## Uploaded Pages with a Class Library



# Refactor as Web Parts

- Re-factor web page as a set of Web Parts
- Advantages
  - Web Parts allow for re-use
  - Can be configured by users
  - Can leverage SharePoint infrastructure
- Disadvantages
  - Requires the most effort

# Refactoring as Web Parts

1. Logically break code into UI components
2. Define data passed between components
3. Create Web Parts for each component
4. Add Web Parts to a page
5. Configure connections between Web parts.



MINDSHARP

# Demo

# DGWO

## Connected Web Parts





MINDSHARP

# Questions?

QUESTIONS?

**THANK YOU FOR ATTENDING!**

Please be sure to fill out  
your session evaluation!

