

The Great SharePoint Adventure 2007

Student Presentations Manual

This course includes the following presentations:

1. Roadmap to SharePoint 2007 Development
2. SharePoint Architecture
3. Master Pages and Site Branding
4. Web Part Development
5. Lists and Content Types
6. Forms Services with InfoPath 2007
7. Developing SharePoint Workflows with Visual Studio
8. The Business Data Catalog (BCD)
9. Web Content Management
10. Application Security



SharePoint 2007 Developer Roadmap

Getting Started with SharePoint Development



Logistics

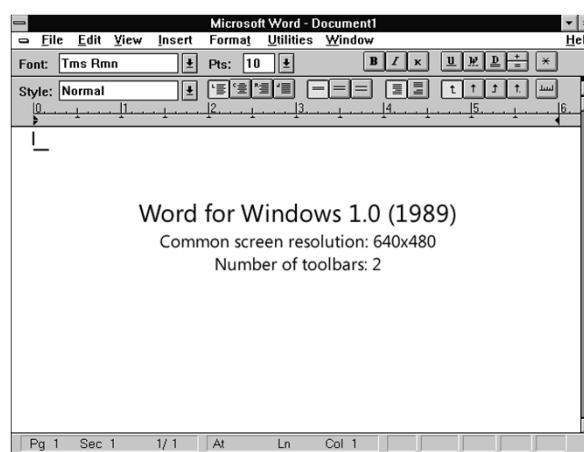
- Basic Human Needs
 - Bathrooms
 - Food and coffee
 - Meals
 - Class hours

Agenda

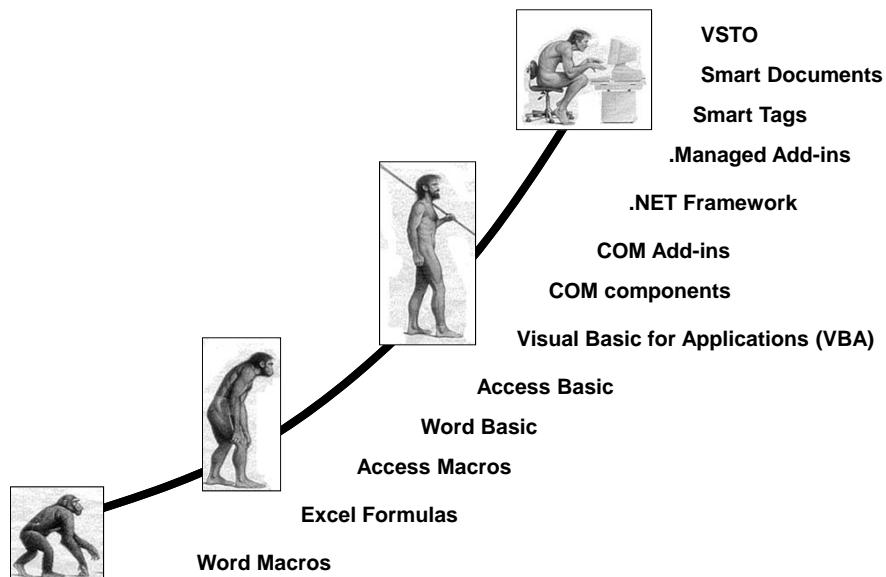
- Architectural overview of SharePoint 2007
 - Windows SharePoint Services 3.0 (WSS)
 - Microsoft Office SharePoint Server 2007 (MOSS)
- Basic WSS Terminology
- WSS as a collaboration solution
- Customizing WSS Sites
- Overview of MOSS components and services

Microsoft Office Through the Ages

- It all started off with a modest productivity tool from a medium-sized company in Redmond

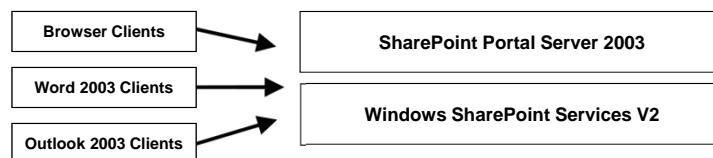


Evolution of the Office Developer

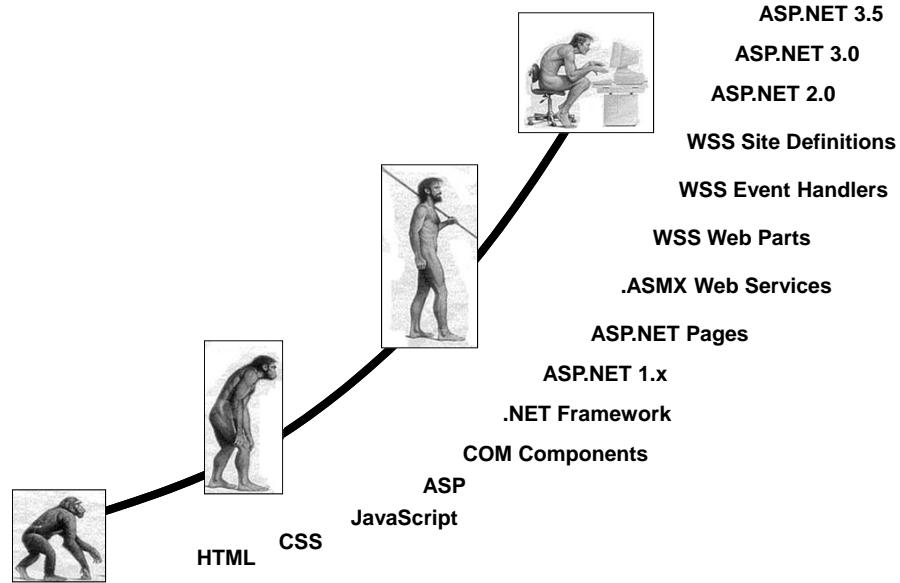


Office 2003 Server Components

- Windows SharePoint Services (WSS v2)
 - Site and Workspace Provisioning Engine
 - Accessibility from browser and Office client applications
 - Out-of-the-box Collaboration Services
- MS Office SharePoint Portal Server 2003 (SPS)
 - Aggregation and search features
 - Social networking (Profiles, Audiences, My Sites)



Evolution of the Web/WSS Developer



Student Questionnaire

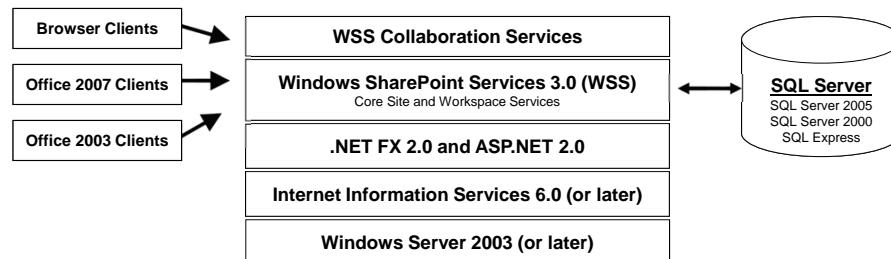
- What's Your Name?
- What Company are you with?
- How have you evolved as a Developer?
- Do you have experience with...
 - The .NET Framework and Visual Studio
 - ASP.NET (what was the latest version)
 - WSS 2.0 and SPS 2003
 - WSS 3.0 and MOSS

Introducing The Office 2007 System

- Windows SharePoint Services 3.0 (WSS)
 - Licensed as part of Windows Server 2003
 - Site provisioning engine and core workspace services
 - Out-of-the-box collaboration features
 - A development platform
think of WSS as ASP.NET extensions
- Microsoft Office SharePoint Server 2007 (MOSS)
 - Licensed separately under its own SKUs
 - New components and services built on top of WSS 3.0
 - Unification of SPS 2003 and CMS 2002
 - Lots of functionality rolled in beyond SPS and CMS

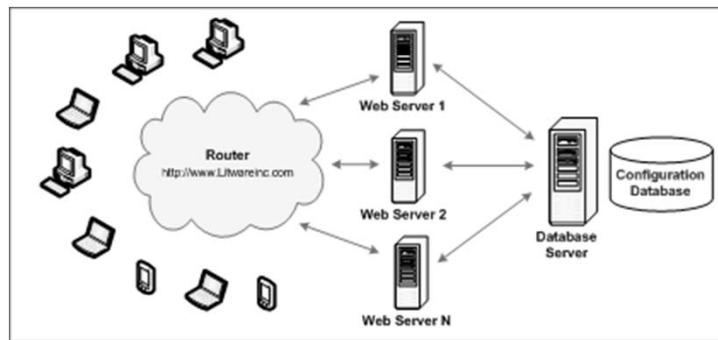
The WSS 3.0 Server-side Platform

- Windows SharePoint Services 3.0 (WSS)
 - An engine for creating/running/managing sites
 - Architecture designed to scale to 10,000s of sites
 - Platform for building Web application and solutions
 - Collaboration services included out-of-the-box



The WSS Farm

- WSS deployment based on a farm
 - Farm requires Web server(s) and database server
 - Farm can be single server or multi-server
 - Each farm has exactly one configuration database



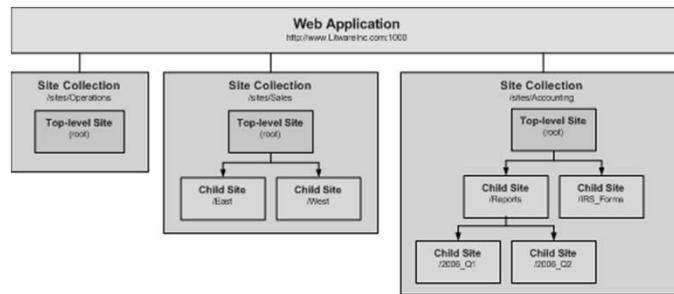
Web Applications

- Web Applications provide HTTP entry points
 - Web Applications based on IIS Web sites
 - Web Application defines one or more URL spaces
 - Web Application security configured independently



Site Collections and Sites

- Sites are partitioned using Site Collections
 - Site collection is scope for administrative privileges
 - Site collection always contains top-level site
 - Site collection may contain hierarchy of child sites
 - Web application can support 1000s of site collections



STSADM.EXE Command-line Utility

- Useful for running administrative commands
 - Can be used interactively from command line
 - Commands can be scripted using batch files

```
C:\>stsadm.exe -help CreateSite
stsadm.exe -o createsite
      [-url <url>]
      [-owneremail <someone@example.com>]
      [-ownerlogin <DOMAIN\name>]
      [-ownername <display name>]
      [-secondaryemail <someone@example.com>]
      [-secondarylogin <DOMAIN\name>]
      [-secondaryname <display name>]
      [-lcid <language>]
      [-sitetemplate <site template>]
      [-title <site title>]
      [-description <site description>]
      [-hostheaderwebapplicationurl <web application url>]
      [-quota <quota template>]

C:\>STSADM.EXE -o CreateSite -url http://LitwareInc.com/sites/Marketing2007
      -ownerlogin LITWAREINC\Administrator -owneremail administrator@litwareinc.com
      -sitetemplate STS#1
Operation completed successfully.

C:\>
```

WSS Central Administration (WSS CA)

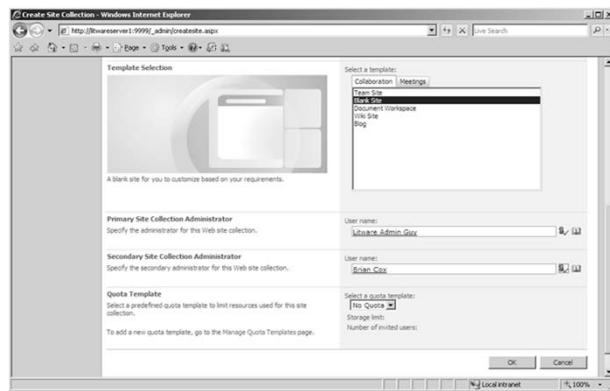
- WSS CA hosted in separate Web Application
 - Used by farm-level administrators
 - WSS CA pages have more links if MOSS is installed

Creating New Site Collections

- Steps to provisioning new site collection
 - Go Application Management tab of WSS CA
 - Click Link titled Create site collection
 - Fill out input form and click OK

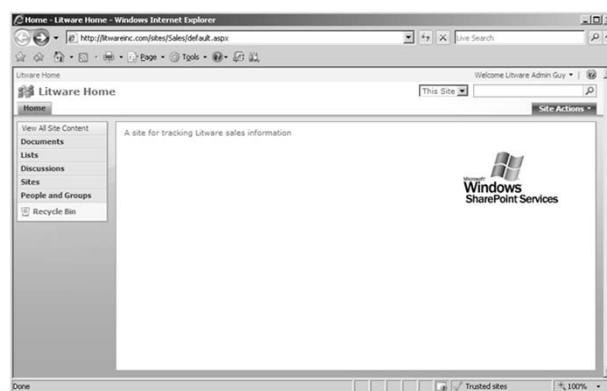
Creating New Site Collections (Part 2)

- Important site collection settings
 - Site template for top-level site
 - Site collection owner(s)



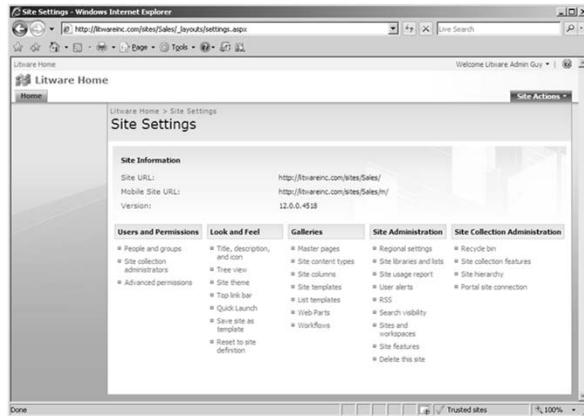
A New WSS Site

- New site collection has top-level sites
 - Site collection owner can provision site elements
 - Site collection owner can create child sites



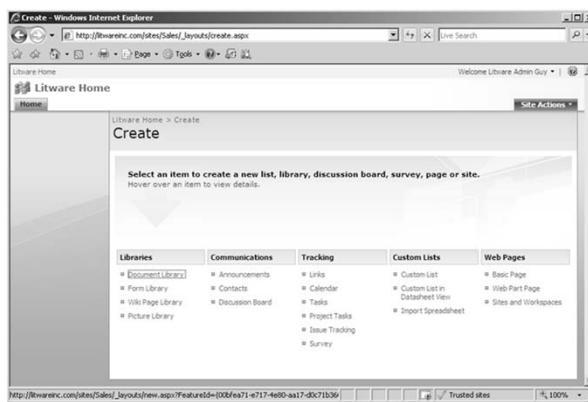
The Site Settings Page

- Site Settings accessible via Site Actions menu
 - Provides links for site and site collection administration



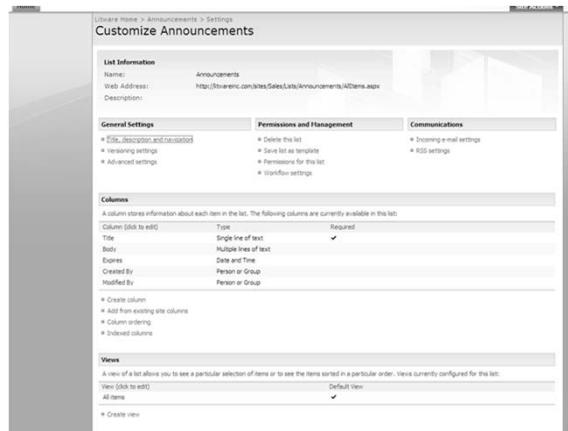
The Create Page

- Create page allows provisioning of site elements
 - WSS provides many collaboration list types out-of-box
 - You can also provision new pages and child site



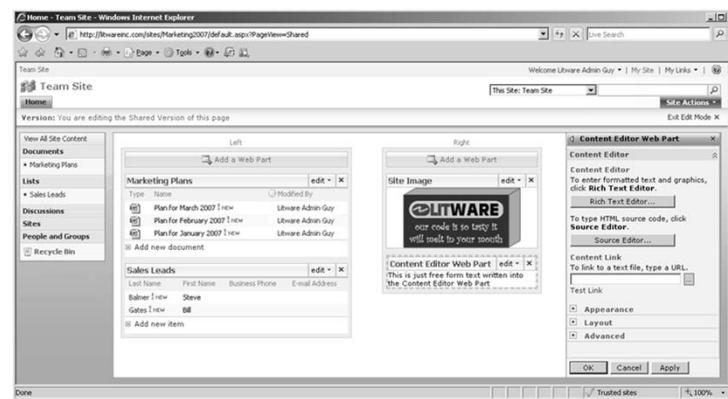
The List Settings Page

- Each List Instance provides a Settings Page
 - You can change list setting and add/remove columns

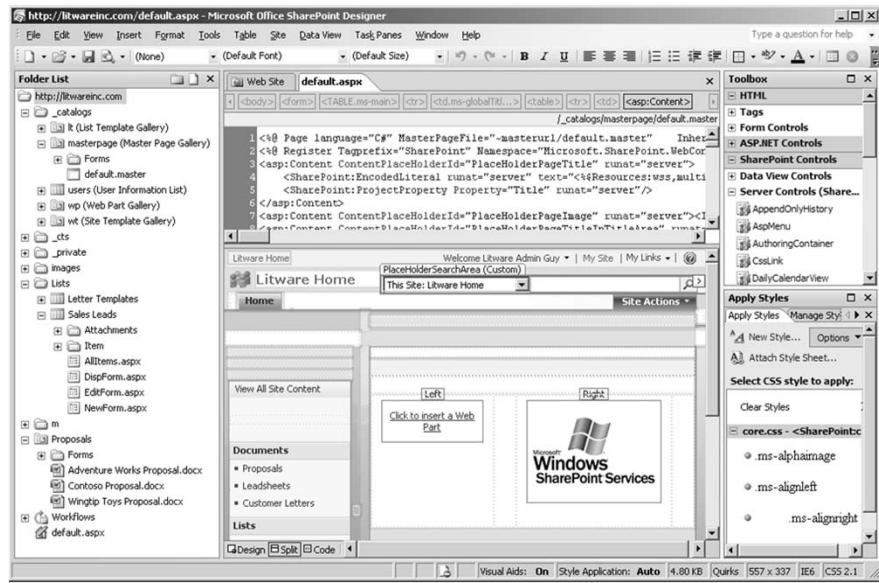


Page Customization using Web Parts

- Web Parts provide page-level customization
 - User can add Web Parts and modify their properties
 - Web Part support customization and personalization

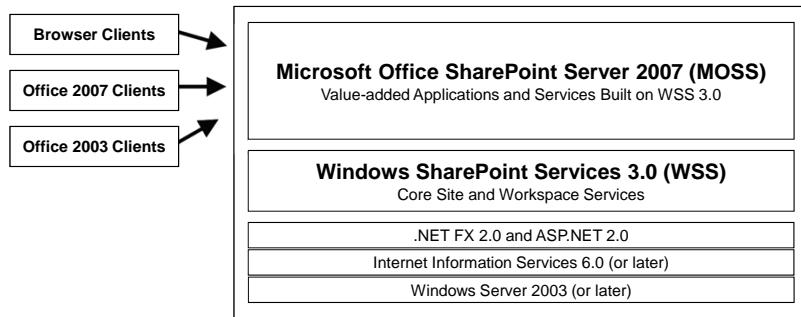


Customization with the SharePoint Designer



Microsoft Office SharePoint Server 2007

- Microsoft Office SharePoint Server 2007 (MOSS)
 - Components and services built on WSS 3.0

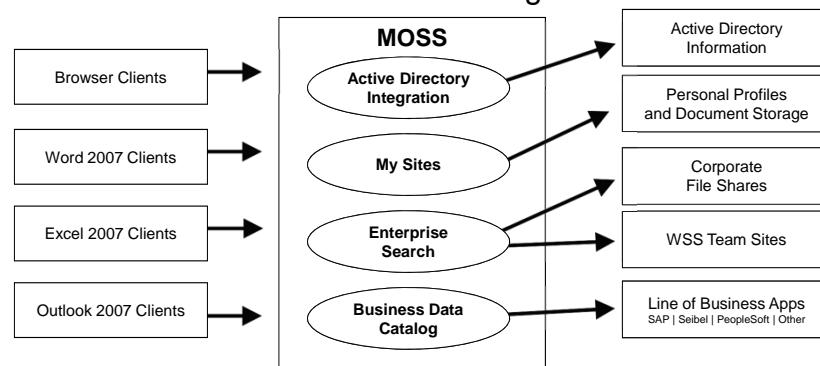


MOSS Services and Components

- What does MOSS Standard Edition provide?
 - Next-generation features of SPS 2003 (Portal)
 - Next-generation features of CMS 2002 (WCM)
- What does MOSS Enterprise Edition provide?
 - Forms Services
 - Business Data Catalog
 - Excel Services

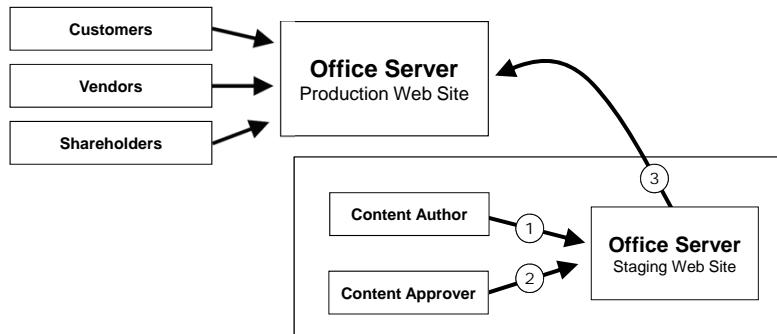
Portal and Search

- MOSS includes next-generation of SPS features
 - User profiles, audience targeting and MySites
 - Enterprise search
 - Introduces Business Data Catalog



Web Content Management

- WCM features designed for public Web sites
 - Core CMS features integrated into MOSS
 - Features for site branding and customized page layouts
 - Profession publishing features for content approval

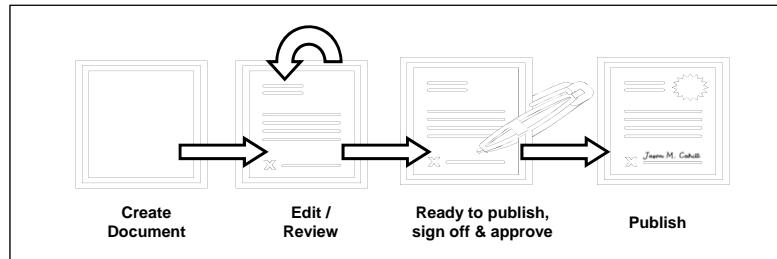


InfoPath 2007 and Forms Services

- InfoPath 2003
 - Capture business data with dynamic, XML-based forms
 - Rich data entry and validation
 - Integration with back-end LOB systems
- InfoPath 2007 and Forms Services
 - Ability to push InfoPath forms to browser-based clients
 - Integration with Office 2007 client applications

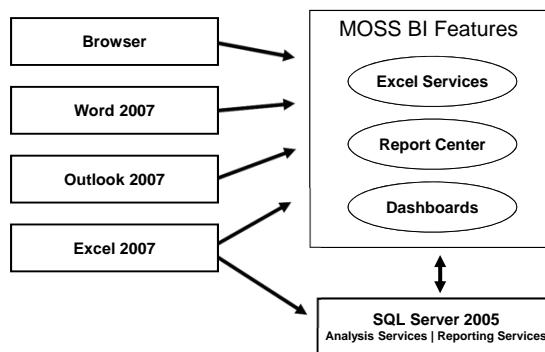
SharePoint 2007 Workflows

- WSS and MOSS provide rich workflow support
 - Support built on Windows Workflow Foundation (WF)
 - WSS provides development platform for workflows
 - MOSS provides several valuable workflows out-of-box

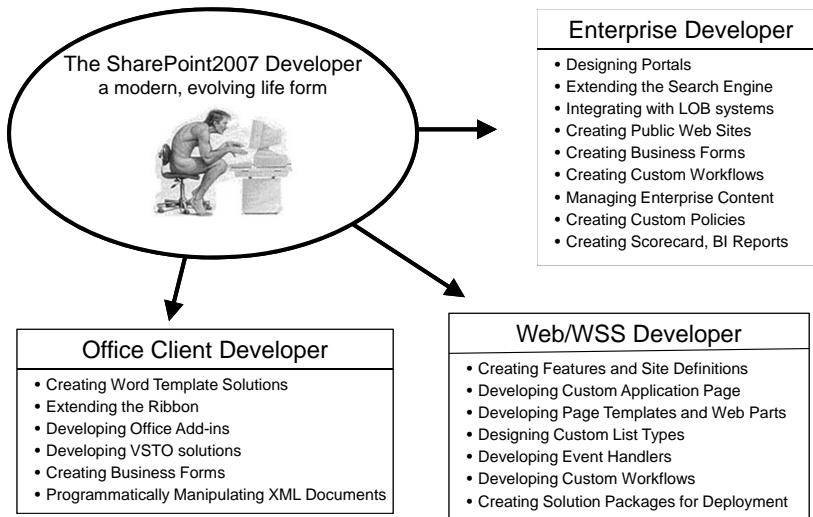


Business Intelligence

- The MOSS Vision for Business Intelligence (BI)
 - Provide business insight to all employees
 - Lead to better, faster, more relevant decisions
 - Integrate with BI features of SQL Server and Excel



What Do "SharePoint Developers" Build?



Schedule of Lectures

1. **Roadmap to WSS Development** << YOU ARE HERE
2. **Developing Features**
3. **SharePoint Architecture**
4. **Page Design and Provisioning**
5. **Master Pages and Site Branding**
6. **Web Part Development**
7. **AJAX Web Parts**
8. **Integrating Silverlight 2**
9. **Lists and Content Types**
10. **Document Libraries**
11. **Site Definitions**
12. **Forms Services and InfoPath 2007**
13. **Introduction to SharePoint Workflows**
14. **Creating MOSS Collaboration Portals**
15. **Web Content Management (WCM) with MOSS**
16. **The Business Data Catalog**
17. **Excel Services and Report Center**
18. **SharePoint Application Security**

Summary

- Architectural overview of SharePoint 2007
 - Windows SharePoint Services 3.0 (WSS)
 - Microsoft Office SharePoint Server 2007 (MOSS)
- Basic WSS Terminology
- WSS as a collaboration solution
- Customizing WSS Sites
- Overview of MOSS components and services

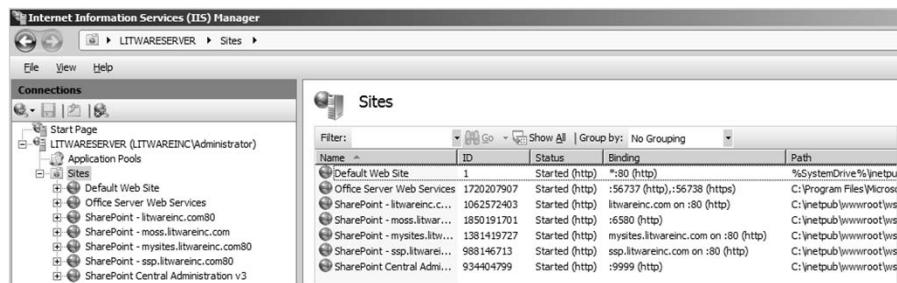


Agenda

- WSS Integration with ASP.NET 2.0
 - IIS Web sites and Web Applications
 - The farm and the configuration database
 - Web Application and Content Database
- content databases
- The web.config file
- Site pages versus application pages
- Creating custom application pages
- Deployment using Solution Packages

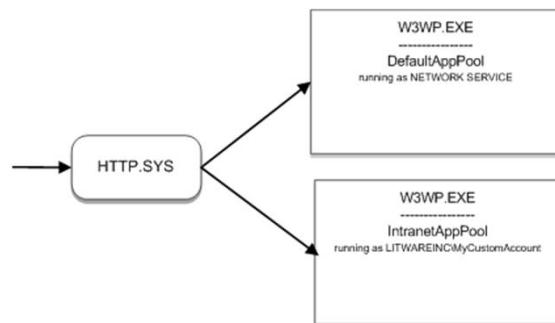
IIS Web Sites

- WSS depends on IIS Web sites for...
 - HTTP listener mechanism
 - Process management through Application Pools
 - Security and user authentication



IIS Application Pools

- IIS dispatches requests to Application Pools
 - Each Application Pool configured to run in own process
 - IIS lets you configure Application Pool identity
 - App Pool identity can be local or domain account



The ASP.NET Framework

- ASP.NET is a productivity framework on top of IIS
 - Integrated with IIS via ISAPI extension (aspnet_isapi.dll)
 - Provides abstractions such as page, request, response
 - Integrates with Visual Studio and managed code

The web.config file

- Provides configuration for ASP.NET runtime

```
<configuration>
  <system.web>

    <customErrors mode="On" />
    <httpRuntime maxRequestLength="51200" />
    <authentication mode="Windows" />
    <identity impersonate="true" />
    <authorization>
      <allow users="*" />
    </authorization>
  </system.web>
</configuration>
```

ASP.NET Pages

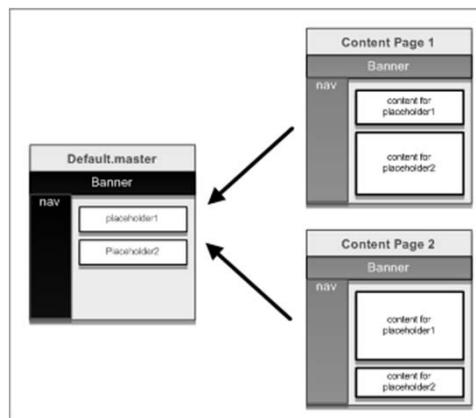
- ASP.NET development typically based on pages
 - Pages are deployed as .ASPX files to Web server
 - .ASPX files parsed and compiled on first request
 - Compiled page class inherits from **System.Web.UI.Page**

```
<%@ Page Language="C#" %>
<script runat="server">
    protected override void OnLoad(EventArgs e) {
        lblDisplay.Text = "Hello, ASP.NET";
    }
</script>

<html>
<body>
    <form id="frmMain" runat="server">
        <asp:Label runat="server" ID="lblDisplay" />
    </form>
</body>
</html>
```

Master Pages in ASP.NET

- ASP.NET 2.0 introduces Master Pages
 - Defines common layouts used across content pages



Linking Content Page to Master Page

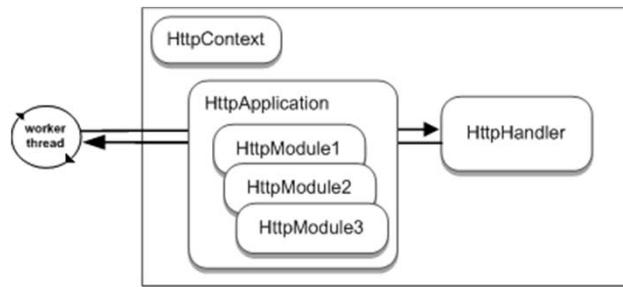
```
<!-- default.master -->
<%@ Master %>
<html><body>
  <form id="frmMain" runat="server">
    <table width="100%">
      <tr><td> <h1>Listware Inc. </h1><hr /></td></tr>
      <tr>
        <td <!-- Display Main Body of Page -->
          <asp:ContentPlaceholder ID="PlaceholderMain" runat="server" />
        </td>
      </tr>
    </table>
  </form>
</body></html>
```

```
<!-- content page linking to default.master -->
<%@ Page Language="C#" MasterPageFile="~/default.master" Title="Page 1" %>

<asp:Content ID="Main" ContentPlaceholderID="PlaceholderMain">
  Unique page content goes here
</asp:Content>
```

The HTTP Pipeline of ASP.NET

- ASP.NET processing based on HTTP pipeline
 - HttpApplication and HttpModule act as interceptors
 - HttpHandler acts as endpoint for request
 - All object types can be replaced with custom code
 - HttpContext object available anywhere in pipeline



The WSS-extended Web Application

Name	Type
_app_bin	File Folder
_controltemplates	Virtual Directory
_layouts	Virtual Directory
_vti_bin	Virtual Directory
_vti_pvt	File Folder
_wpresources	Virtual Directory
App_Browsers	File Folder
App_GlobalResources	File Folder
bin	File Folder
wresources	File Folder
global.asax	ASP.NET Server Application
web.config	XML Configuration File

- **Web Applications extend IIS and ASP.NET**
 - IIS wildcard application map sends all requests to ASP.NET
 - ASP.NET extended using common objects inside HTTP pipeline
 - Web Application configured with WSS system virtual directories
 - `_layouts`
 - `_controltemplates`
 - `_vti_bin`
 - `_wpresources`

The WSS-extended web.config file

- **WSS replaces HttpApplication object**

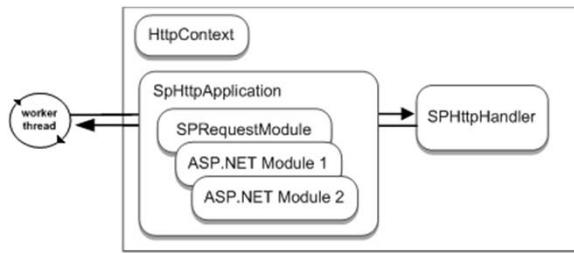
```
<!-- global.asax file at root of WSS Web Application -->
<@Application Inheritance="Microsoft.SharePoint.ApplicationRuntime.SPHttpApplication" >
```

- **WSS configures pipeline with its own HttpHandler and HttpModule**

```
<!-- web.config file at root of WSS Web Application -->
<configuration>
  <system.web>
    <httpHandlers>
      <remove verb="GET,HEAD,POST" path="*" />
      <add verb="GET,HEAD,POST" path="*"
           type="Microsoft.SharePoint.ApplicationRuntime.SPHttpHandler,..." />
    </httpHandlers>
    <httpModules>
      <clear />
      <add name="SPRequest"
           type="Microsoft.SharePoint.ApplicationRuntime.SPRequestModule,..." />
      <!-- other standard ASP.NET httpModules added back in -->
    </httpModules>
  </system.web>
</configuration>
```

WSS Web Applications

- WSS extends HTTP pipeline with custom objects
 - Configuration added to every WSS Web Application
 - Modifications made to web.config file and IIS metabase



- Different and superior architecture than WSS 2.0
 - WSS 2.0 architecture based on problematic ISAPI filter

WSS Extensions to the web.config file

```
<configuration>
  <configSections>
    <sectionGroup name="SharePoint">
      <section name="SafeControls" type="..."/>
      <section name="RuntimeFilter" type="..."/>
      <section name="WebPartLimits" type="..."/>
      <section name="WebPartCache" type="..."/>
      <section name="WebPartWorkflow" type="..."/>
      <section name="WebPartControls" type="..."/>
      <section name="SafeMode" type="..."/>
      <section name="MergedActions" type="..."/>
      <section name="PeoplePickerWildcard" type="..."/>
    </sectionGroup>
  </configSections>
  <SharePoint>
    <SafeMode />
    <WebPartLimits />
    <WebPartCache />
    <WebPartControls />
    <SafeControls />
    <PeoplePickerWildcard />
  </SharePoint>
</configuration>
```

Important Debugging Settings

```

<configuration>
  <configSections>...
  <sharePoint>
    <safeMode MaxControls="200" callstack="false" DirectFileDependencies="10"
      <PageParserPaths>...
      </PageParserPaths>
    </safeMode>
    <webPartLimits MaxZoneParts="50" PropertySize="1048576" />
    <webPartCache Storage="CacheObject" />
    <webPartControls datasheetControlGuid="65BCBEE4-7728-41a0-97BE-14E1CAE36A/
    <safeControls>...
    <peoplePickerWildcards>
      <clear />
      <add key="AspNetSqlMembershipProvider" value="%" />
    </peoplePickerWildcards>
    <mergedActions>...
    <blobCache location="C:\blobCache" path=".(gif|jpg|png|css|js)$" maxsize=
    <runtimeFilter Assembly="Microsoft.Office.Server, Version=12.0.0.0, Cultur
  </sharePoint>
  <system.web>
    <securityPolicy>...
    <httpHandlers>...
    <customErrors mode="On">
      <httpRuntime maxRequestLength="51200" />
    </customErrors>
  </system.web>

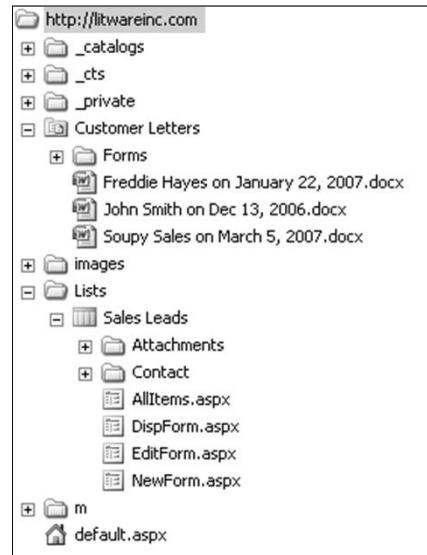
```

Annotations:

- A callout arrow points from a box labeled "set to true" to the "callstack" attribute of the `<safeMode>` section.
- A callout arrow points from a box labeled "set to Off" to the "mode" attribute of the `<customErrors>` section.

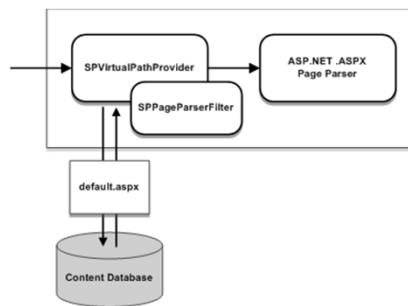
The Virtual File System of a Site

- Site is a virtual file system
 - made up of folders and files
 - Pages are files
 - Documents are files
 - Stored in content database
- How can you look at it?
 - SharePoint Designer
 - Windows Explorer (WebDav)



Processing Pages within a Site

- WSS stores.aspx files in content database
 - Retrieved using SPVirtualPathProvider object
 - Page based on page templates on Web server
 - Non-customized pages can be ghosted
 - Customized pages cannot be ghosted

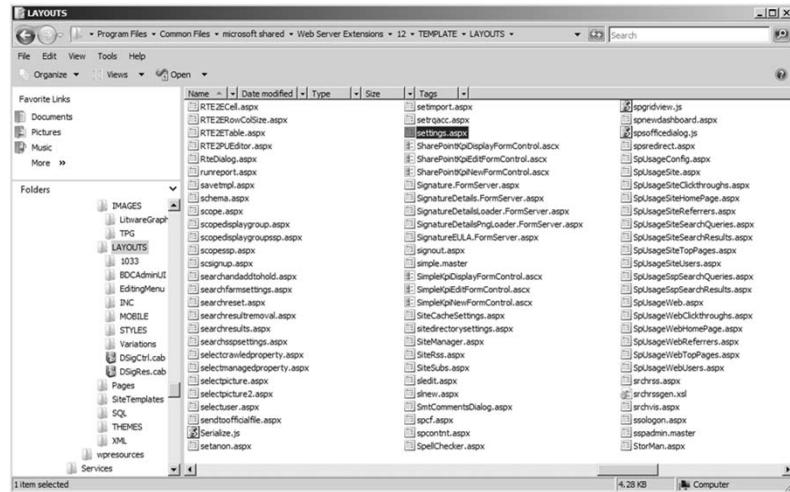


The _layouts Virtual Directory

- Files in _layouts directory accessible to all sites
 - _layouts provides access to common resources
 - _layouts contains files for images, CSS and JavaScript
 - _layouts contains Application Pages
- All these URLs resolve to the same page
 - `http://LitwareInc.com/_layouts/settings.aspx`
 - `http://LitwareInc.com/sites/Vendors/_layouts/settings.aspx`
 - `http://LitwareInc.com:1001/sites/Accounting/_layouts/settings.aspx`

Application Pages

- Standard Application Pages are part of WSS



Site Pages Versus Application Pages

- Site Pages exist within virtual file system of site
 - They may or may not be ghosted
 - They support customization via Web Parts
 - They support customization via SharePoint Designer
 - Customized pages impact performance and security
 - Application Pages are deployed once per farm
 - They do not support customization or Web Parts
 - They are parsed/compiled as classic ASP.NET pages
 - They run faster than Site Pages
 - They always support code behind

Creating Custom Application Pages

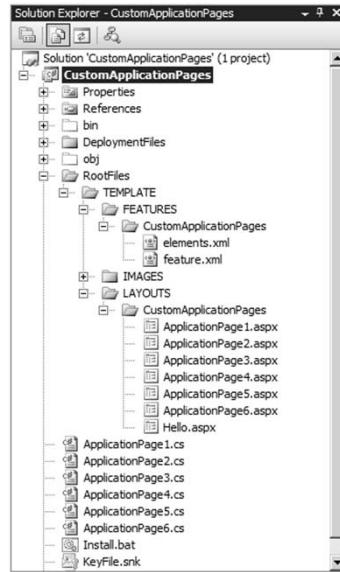
- Steps to creating a custom Application Page
 - Inherit from LayoutsPageBase
 - Link to application.master
 - Add server-side controls and code
 - Deploy to LAYOUTS directory



'Hello World' Custom Application Page

```
<%@ Assembly Name="Microsoft.SharePoint, [full 4-part name]"%>
<%@ Page Language="C#" MasterPageFile="~/_layouts/application.master"
Inherits="Microsoft.SharePoint.WebControls.LayoutsPageBase" %>
<%@ Import Namespace="Microsoft.SharePoint" %>
<script runat="server">
    protected override void OnLoad(EventArgs e) {
        // SPWeb site = SPContext.Current.Web;
        SPWeb site = this.Web; // base class provides access to WSS objects
        Label SiteTitle = site.Title;
        Label SiteID = site.ID.ToString().ToUpper();
    }
</script>
<asp:Content ID="Main" contentplaceholderid="PlaceHolderMain" runat="server">
    Site Title: <asp:Label ID="lblSiteTitle" runat="server"/><br />
    Site ID: <asp:Label ID="lblSiteID" runat="server" />
</asp:Content>
<asp:Content ID="PageTitleArea" runat="server"
contentplaceholderid="PlaceHolderPageTitleArea" >
    The Quintessential 'Hello World' of Application Page
</asp:Content>
```

Demo: CustomApplicationPages



Adding a Feature for Navigation

- Feature can be used with custom applications
 - Custom actions provide navigation menu items

```
<?xml version="1.0" encoding="utf-8" ?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <!-- Add Menu Command to Site Actions Dropdown -->
  <CustomAction Id="HelloApplicationPage"
    GroupId="SiteActions"
    Location="Microsoft.SharePoint.StandardMenu"
    Sequence="2000"
    Title="Hello World Application Page"
    Description="Getting up and going with inline code">

    <Url Action.Url = "~site/_layouts/CustomApplicationPages/Hello.aspx"/>

  </CustomAction>
</Elements>
```

Adding an ECB Menu Item

- Custom ECB menu items can be added to lists
 - Redirect to application page
- Registration Types
 - List
 - Content Type
 - File Extension

```
<CustomAction
  Id="CustomAppli cati onPage4"
  RegistrationType="List"
  RegistrationId="101"
  ImageUrl="/_layouts/images/GORTL.GIF"
  Location="EditControlBlock"
  Sequence="240"
  Title="Appli cati on Page 4" >
<Url Action="~site/_layouts/CustomAppli cati onPages/
  Appli cati onPage4.aspx?Itemld={Itemld}&ListId={ListId}" />
</CustomAction>
```

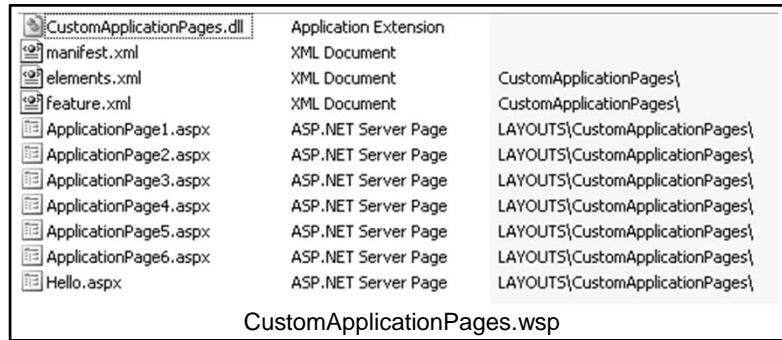


Deployment using Solution Packages

- Evolution of Web Part Packages from WSS 2.0
 - Solution Package is a CAB file with .wsp extension
 - Solution Package contains a manifest
 - Solution Package contains files required on Web server
- What can be deployed via a Solution Package
 - Feature definitions
 - Application Pages
 - Assembly DLLs
 - And much more...

Deployment using Solution Packages

- WSS Deployment done with Solution Packages
 - Solution Package is CAB file with .wsp extension
 - Created using DDF file and MAKECAB.EXE
 - Deployed using STSADM.EXE or WSS Central Admin



Solution Package Manifest

- Solution Manifest read by WSS installer

```
<Solution SolutionId="9EFFE92B-781D-4c99-BBC0-432D248B899D"
          xmlns="http://schemas.microsoft.com/sharepoint/">

  <FeatureManifests>
    <FeatureManifest Location="CustomApplicationPages\feature.xml" />
  </FeatureManifests>

  <Templates>
    <TemplateFile Location="LAYOUTS\CustomApplicationPages\Hello.aspx"/>
    <TemplateFile Location="LAYOUTS\CustomApplicationPages\ApplicationPage1.aspx"/>
    <TemplateFile Location="LAYOUTS\CustomApplicationPages\ApplicationPage2.aspx"/>
    <TemplateFile Location="LAYOUTS\CustomApplicationPages\ApplicationPage3.aspx"/>
    <TemplateFile Location="LAYOUTS\CustomApplicationPages\ApplicationPage4.aspx"/>
    <TemplateFile Location="LAYOUTS\CustomApplicationPages\ApplicationPage5.aspx"/>
    <TemplateFile Location="LAYOUTS\CustomApplicationPages\ApplicationPage6.aspx"/>
  </Templates>

  <Assemblies>
    <Assembly Location="CustomApplicationPages.dll"
              DeploymentTarget="Global Assembly Cache" />
  </Assemblies>
</Solution>
```

Solution Package: install vs. deploy

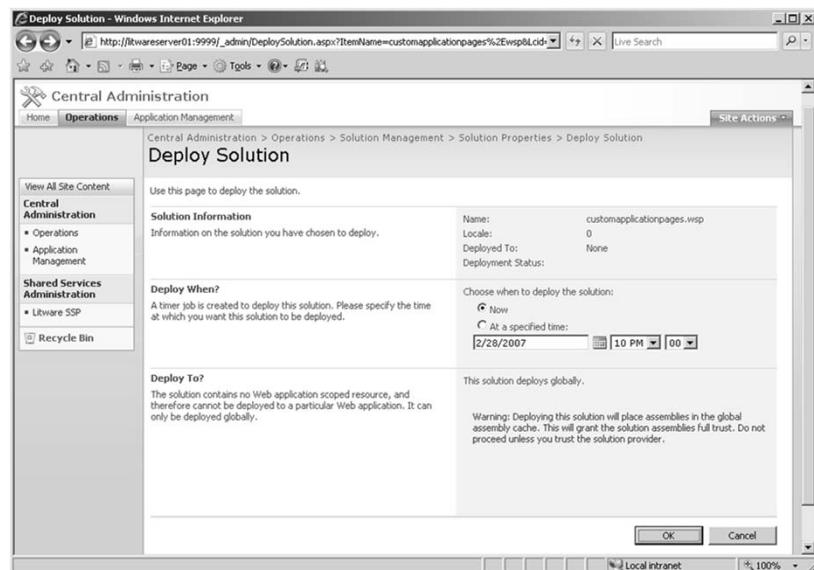
- Solution Package Installation
 - WSP file copied into configuration database
 - Done using **addsolution** operation of STSADM.EXE
- Solution Package Deployment
 - WSP files copied to each FE Web Server and deployed
 - Done using **deploysolution** operation of STSADM.EXE

```
REM – a batch file named DeploySolutionPackage.cmd from CustomApplicationPage project
Echo Generating Solution Package CustomApplicationPages.wsp
If EXIST CustomApplicationPages.wsp del CustomApplicationPages.wsp
cd ..
makecab /f Solution\cab.ddf
cd package

Echo Installing CustomApplicationPages.wsp in WSS Solution Package Store
%STSADM% -o addsolution -fIename CustomApplicationPages.wsp
%STSADM% -o execadmsvcjobs

Echo Deploying Solution Package CustomApplicationPages.wsp
%STSADM% -o deploysolution -name CustomApplicationPages.wsp -immediate -allowGacDeployment
%STSADM% -o execadmsvcjobs
```

Deploying Solution Packages



Summary

- WSS Integration with ASP.NET 2.0
 - IIS Web sites and Web Applications
 - The farm and the configuration database
 - Web Application and Content Database
- content databases
- The web.config file
- Site pages versus application pages
- Creating custom application pages
- Deployment using Solution Packages



Page Design and Provisioning

Adding Customizable Pages

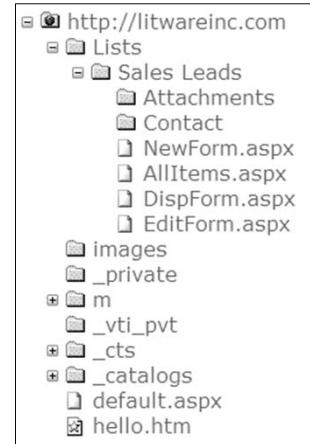


Agenda

- Page parsing and Safe Mode restrictions
- Creating custom page templates
- Provisioning page instances
- Designing Web Part Pages

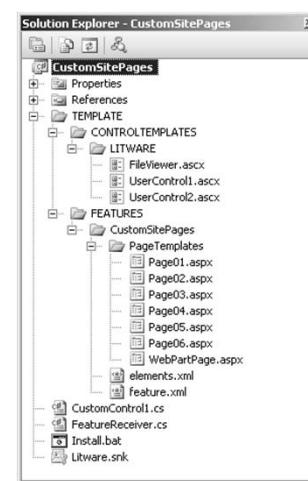
Site Page Fundamentals

- Site Pages are part of site
 - Represented with SPFile objects
 - Structured in SPFolder objects



Demo: CustomSitePages

- Important Concepts
 - Page template vs. page instance
 - Page customization
 - SafeMode processing



'Hello World' Page Template

- Page Template can be added to feature
 - MasterPageFile points to ~masterurl /default.master
 - progid adds support for SharePoint Designer

```
<%@ Page MasterPageFile="~masterurl /default.master"
   meta: progid="SharePoint.WebPartPage.Document" %>

<asp:Content runat="server" ContentPlaceHolderID="PlaceHolderMain">
    <h3>Hello World</h3>
    A simple page template used to create site pages
</asp:Content>
```

Provisioning a Page Instance

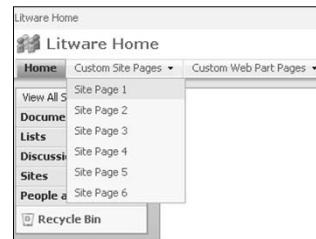
- Module element used to provision page instance
 - File element per page instance
 - Supports page ghosting



```
<Elements xml:ns="http://schemas.microsoft.com/sharepoint/">
    <Module Path="PageTemplates" Url="SitePages" >
        <File Url="Page01.aspx" Type="Ghostable" />
    </Module>
</Elements>
```

Adding Navigation Support for Pages

- Navigation nodes can be added
 - Can be added during feature activation
 - Can be added to top-link bar
 - Can be added to QuickLaunch
 - Nodes created as SPNavigationNode

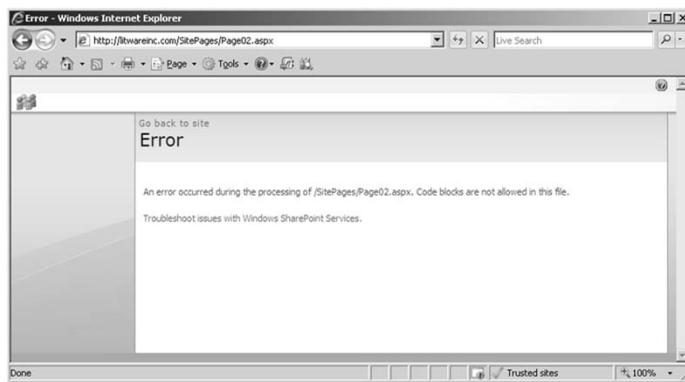


```
public class FeatureReceiver : SPFeatureReceiver {
    public override void FeatureActivated(SPFeatureReceiverProperties properties) {
        // get a hold off current site in context of feature activation
        SPWeb site = (SPWeb)properties.Feature.Parent;
        SPNavigationNodeCollection topNav = site.Navigation.TopNavigation;

        // create dropdown menu for custom site pages
        SPNavigationNode DropDownMenu1 =
            new SPNavigationNode("Custom Site Pages", "", false);
        topNav[0].Children.AddAsLast(DropDownMenu1);
        DropDownMenu1.Children.AddAsLast(
            new SPNavigationNode("Site Page 1", "SitePages/Page01.aspx"));
    }
}
```

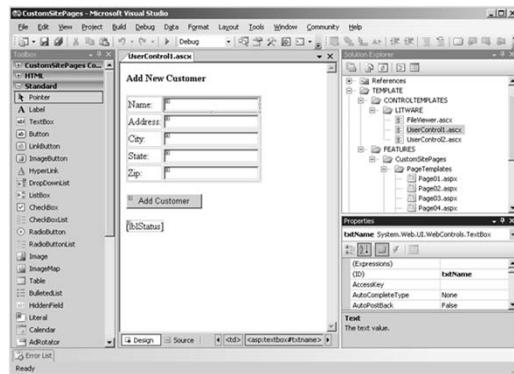
Safe Mode Processing

- Customized site pages run in SafeMode
 - They do not support inline code
 - They only support controls registered as SafeControls



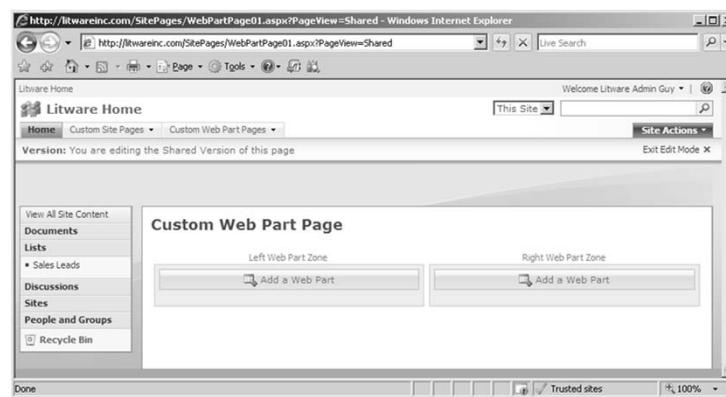
Designing Pages with Controls

- Two kinds of ASP.NET controls
 - Custom controls
 - User controls



Designing Web Part Pages

- Creating a Web Part Page template
 - Inherit from WebPartPage
 - Add one or more Web Part Zones



```

<%@ Page Language="C#" MasterPageFile="~masterurl/default.master"
    Inherits="Microsoft.SharePoint.WebPartPages.WebPartPage, [asm name]"
    meta:progId="SharePoint.WebPartPage.Document" %>

<%@ Register Tagprefix="WebPartPages"
    Namespace="Microsoft.SharePoint.WebPartPages"
    Assembly="Microsoft.SharePoint, [asm name]" %>

<asp:Content ID="main" runat="server" ContentPlaceholderID="PlaceholderMain">

<h3>Custom Web Part Page</h3>

<table width="100%">
    <tr>
        <td valign="top" style="width: 50%">
            <WebPartPages:WebPartZone ID="Left" runat="server"
                FrameType="TitleBarOnly"
                Title="Left Web Part Zone" />
        </td>
        <td valign="top" style="width: 50%">
            <WebPartPages:WebPartZone ID="Right" runat="server"
                FrameType="TitleBarOnly"
                Title="Right Web Part Zone" />
        </td>
    </tr>
</table>
</asp:Content>

```

Adding Web Parts into Zones

- Web Parts can be pre-populated into zones
 - Can be done declaratively through CAML
 - Can be done programmatically through WSS OM

```

<File Url="WebPartPage.aspx" Name="WebPartPage03.aspx" Type="Ghostable" >
    <!-- Add a Web Part to right zone -->
    <AllUsersWebPart WebPartZoneID="Right" WebPartOrder="0">
        <![CDATA[
            <WebPart xmlns="http://schemas.microsoft.com/WebPart/v2"
                xmlns:wp="http://schemas.microsoft.com/WebPart/v2/Image">
                <Assembly>Microsoft.SharePoint, [asm name]</Assembly>
                <TypeName>Microsoft.SharePoint.WebPartPages.ImageWebPart</TypeName>
                <FrameType>None</FrameType>
                <Title>Watch My Gears Run</Title>
                <wp:ImageLink>/_layouts/images/GEARS_AN.GIF</wp:ImageLink>
            </WebPart>
        ]]>
    </AllUsersWebPart>
</File>

```

Summary

- Page parsing and Safe Mode restrictions
- Creating custom page templates
- Designing Web Part Pages
- Master Pages
- Branding a site collection with a custom feature
- Understanding and extending core.css



Developing Web Parts

Creating User Interface Components that
Support Customization and Personalization

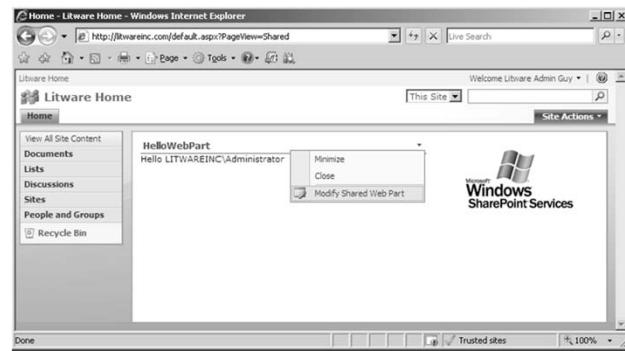


Agenda

- Developing ASP.NET Web Parts for WSS 3.0
- Persistent Web Part properties
- Importing Web Parts into the Web Part Gallery
- Creating a feature to for deploying Web Parts
- Advanced Web Part Techniques

Web Parts

- Web Parts are used to build portal-style applications
 - Content is modular, consistent and easy to navigate
 - Configurable chrome: border and title bar
 - Web Parts support for customization/personalization

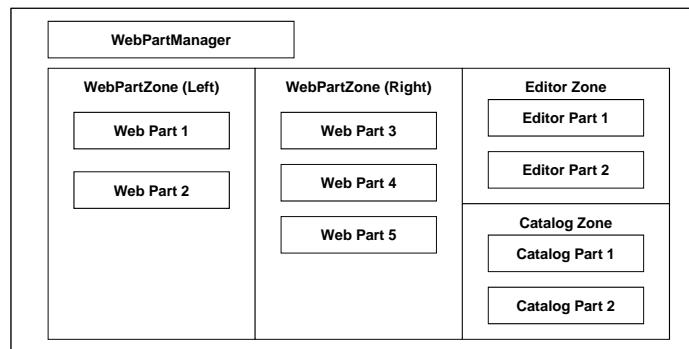


Web Part History

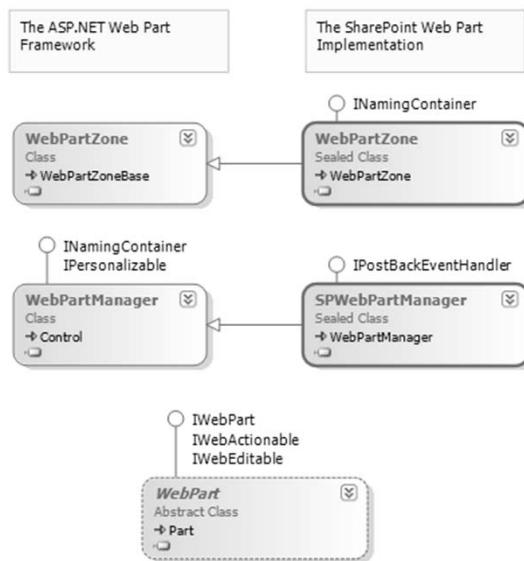
- Windows SharePoint Services 2.0 (WSS V2)
 - Designed with its own Web Part infrastructure
 - WSS serializes/stores/retrieves personalization data
- ASP.NET 2.0
 - Designed with a newer universal Web Part infrastructure
 - Serializes/stores/retrieves personalization data
 - More flexible and more extensible than WSS
 - ASP.NET 2.0 does not support WSS v2 Web Parts
- Windows SharePoint Services 2007 (WSS V3)
 - Supports WSS V2 style Web Parts
 - Supports ASP.NET 2.0 style Web Parts (preferred)

ASP.NET Web Part Page Structure

- Web Part Page in ASP.NET 2.0
 - One instance of the WebPartManager class
 - One or more Web Part Zones
 - Optionally an Editor Zone and/or a Catalog Zone

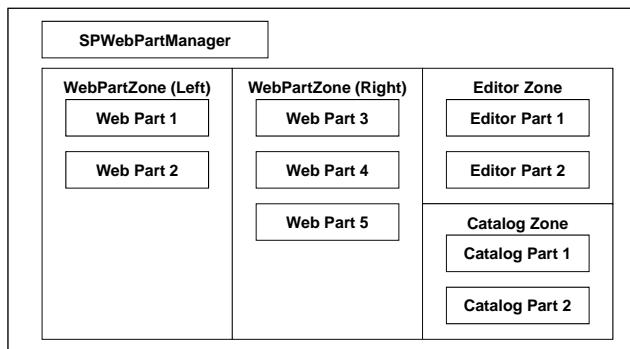


SharePoint's Web Part Implementation



WSS Web Part Page Structure

- Web Part Pages in WSS
 - Inherits from the WSS WebPartPage base class
 - Contains one SPWebPartManager control
 - Contains one or more WSS WebPartZone controls



Overview of Developing Web Parts

1. Create a new class library DLL project
 - Create class that inherits from ASP.NET Web Part class
 - Override methods as required (e.g. RenderContents)
2. Deploy Web Part DLL
 - Compile DLL into \bin directory
 - Configure DLL in web.config file SafeControl list
3. Import Web Part into a WSS site collection
 - Add Web Part class to Web Part Gallery
 - Add Web Part to zone on a Web Part Page

ASP.NET 2.0 Web Parts

- Web Parts derive from the WebPart base class
 - All Web Parts inherit common functionality

```
using System;
using System.Web.UI;
using System.Web.UI.WebControls.WebParts;

namespace LiftwareWebParts {

    public class HelloWorld : WebPart {

        protected override void RenderContents(HtmlTextWriter writer) {
            writer.WriteLine("Hello, world");
        }
    }
}
```

Persistent Web Part Properties

- Web Parts support persistent properties
 - Customization data is seen by all users
 - Personalization data is seen only by one user

```
namespace LiftwareWebParts {
    public class HelloWorld : WebPart {

        protected string _ZipCode;

        [Personalizable(true), WebBrowsable(true),
        WebDisplayName("Zip Code"),
        WebDescription("used to track user zip code")]
        public string ZipCode {
            get{ return _ZipCode; }
            set{ value = _ZipCode; }
        }
        //...
    }
}
```

Web Part As A Safe Control

- Web Parts usually run on Web Part Pages
 - Web Parts must be registered as Safe in web.config file
 - You must add entry to web.config before testing

```
<!-- web.config in Web Application root directory -->

<configuration>
  <SharePoint>
    <SafeControls>
      <SafeControl Assembly="AcmeWebParts"
                   Namespace="AcmeWebParts"
                   TypeName="*"
                   Safe="True" />
    </SafeControls>
  </SharePoint>
</configuration>
```

Web Part Security Caveats

- Web Parts in \bin subject to security restrictions
 - Security restrictions from Code Access Security (CAS)
 - You might want to turn off security during development
- You can choose between three built-in levels

WSS_Minimum (default for WSS V3)
WSS_Medium
Full

```
<!-- web.config -->
<configuration>
  <system.web>
    <!-- default setting for WSS and MOSS -->
    <trustLevel="WSS_Minimal" originUrl="" />
  </system.web>
</configuration>
```

The Web Part Gallery (WPG)

- The WPG is scoped at Site Collection level
 - Contains list of Web Parts available to place on pages
 - Contains .webpart files and .dwp files

Litware Sales Site > Web Part Gallery

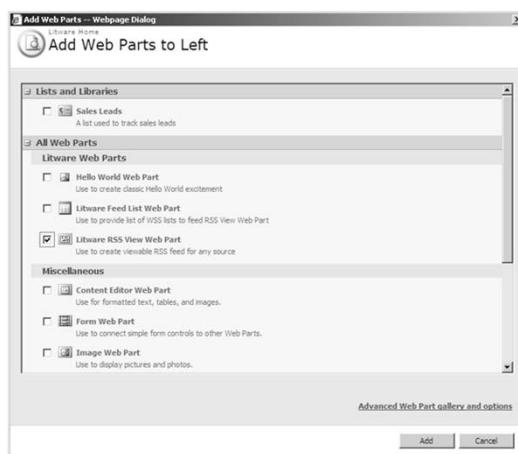
Web Part Gallery

Use this Web Part Gallery to store and retrieve Web Parts. The Web Parts in this gallery are available to this site and all sites under it.

Type	Web Part	Edit	Modified	Modified By
	DemoAspWebPart.webpart [NEW]		1/2/2006 10:23 PM	LitwareInc Administrator
	DemoHybridWebPart.webpart [NEW]		1/2/2006 10:23 PM	LitwareInc Administrator
	DemoWssWebPart.dwp [NEW]		1/2/2006 10:23 PM	LitwareInc Administrator
	MSContentEditor.dwp		12/30/2005 11:03 AM	LitwareInc Administrator
	MSImage.dwp		12/30/2005 11:03 AM	LitwareInc Administrator
	MSMembers.dwp		12/30/2005 11:03 AM	LitwareInc Administrator
	MSPageViewer.dwp		12/30/2005 11:03 AM	LitwareInc Administrator
	MSSimpleForm.dwp		12/30/2005 11:03 AM	LitwareInc Administrator
	MSXml.dwp		12/30/2005 11:03 AM	LitwareInc Administrator

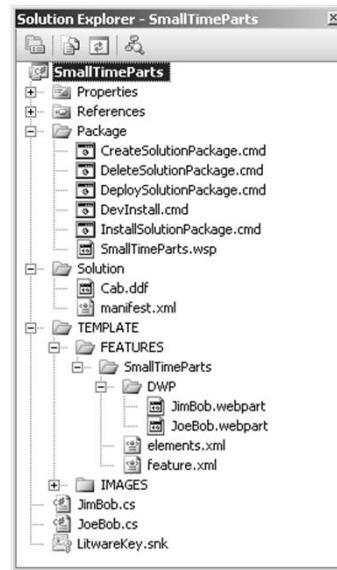
Adding Web Parts from the Gallery

- WSS provides standard dialog for adding parts



Demo: SmallTimeParts

- Important Concepts
 - .webpart files
 - Web Part Deployment Feature
 - Web Part Solution Package
 - Custom CAS Settings



Provisioning .webpart files

- .webpart file needs to be included with WP deployment feature

```
<webParts>
  <webPart xml ns="http://schemas.microsoft.com/WebPart/v3">
    <metaData>
      <type name="SmallTimeParts.JimBob, SmallTimeParts, [full 4-part assembly name]" />
      <importErrorMessage>Cannot import this Web Part.</importErrorMessage>
    </metaData>
    <data>
      <properties>
        <!-- standard Web Part properties -->
        <property name="ChromeType" type="chrometype">Default</property>
        <property name="Title" type="string">Jim Bob's Web Part</property>
        <property name="Description" type="string">Some valuable description goes here</property>
      </properties>
    </data>
  </webPart>
</webParts>
```

- Modules is then used to provision .webpart file into Web part Gallery

```
<!-- this module goes in the feature used to deploy your Web Parts -->
<Module Name="SmallTimeParts" List="113" Url="_catalogs/wp" Path="dwp" RootWebOnly="true">

  <File Uri="JimBob.webpart" Type="GhostableInList" >
    <Property Name="Group" Value="A Set of Small Time Web Parts" />
  </File>

</Module>
```

Solution Manifest for WP Deployment

```
<Solution SolutionId="DEADBEEF-BADD-BADD-BADD-BADBADBADBAD"
  xmlns="http://schemas.microsoft.com/sharepoint/">

  <FeatureManifests>
    <FeatureManifest Location="SmallImageParts\feature.xml" />
  </FeatureManifests>

  <TemplateFiles>
    <TemplateFile Location="IMAGES\TPG\compass.gif" />
    <TemplateFile Location="IMAGES\TPG\SmallCompass.gif" />
    <TemplateFile Location="IMAGES\TPG\SmallBubbles.gif" />
  </TemplateFiles>

  <Assemblies>
    <Assembly DeploymentTarget="WebApplication" Location="SmallImageParts.dll">
      <SafeControls>
        <SafeControl Assembly="SmallImageParts, [full 4-part assembly name]"
          Namespace="SmallImageParts" TypeName="*" Safe="True"/>
      </SafeControls>
    </Assembly>
  </Assemblies>

  <CodeAccessSecurity>
    <!-- use custom CAS policy if needed for deployment in \bin -->
  </CodeAccessSecurity>
</Solution>
```

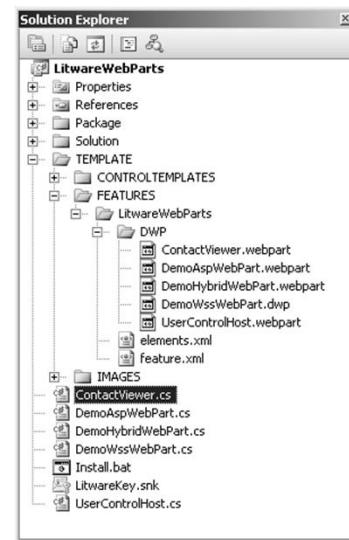
Solution Manifest for WP Deployment

```
<Solution SolutionId="DEADBEEF-BADD-BADD-BADD-BADBADBADBAD"
  xmlns="http://schemas.microsoft.com/sharepoint/">
  <!-- other solution elements omitted for clarity -->
  <CodeAccessSecurity>

    <PolicyItem>
      <!-- create permission set for this policy -->
      <PermissionSet class="NamedPermissionSet" version="1">
        Description="Permission set for SmallImageParts assembly"
        <!-- add generic .NET CAS security permission -->
        <Permission class="SecurityPermission" version="1"
          Flags="Execution, UnmanagedCode, ControlThread" />
      <!-- add ASP.NET hosting permission -->
      <Permission class="AspNetHostingPermission" version="1" Level="High" />
      <!-- add SharePoint permission -->
      <Permission class="Microsoft.SharePoint.Security.SharePointPermission"
        version="1" ObjectModel="true" Impersonate="true" UnsafeSaveOnGet="true" />
    </PermissionSet>
    <!-- add assembly to be associated with this policy -->
    <Assemblies>
      <Assembly Name="SmallImageParts" />
    </Assemblies>
  </PolicyItem>
</CodeAccessSecurity>
</Solution>
```

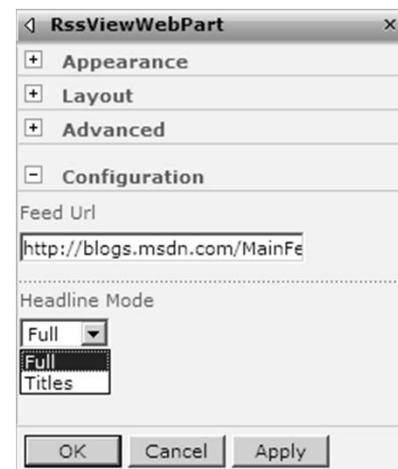
Demo: LitwareWebParts

- Important Concepts
 - Editor Parts
 - Web Part Verbs
 - Web Part Connections
 - Asynchronous Processing



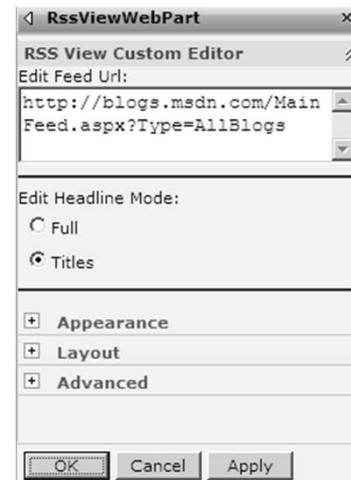
Standard Editor Parts

- WSS provides standard editor parts



A Custom Editor Part

- Custom Editor Parts provide more control
 - Control over rendering
 - Control over validation



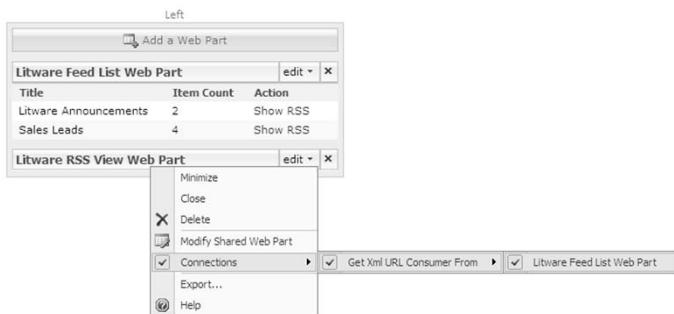
Web Part Verbs

- Used to add menu items to Web Parts
 - Supports client-side handlers through JavaScript
 - Supports server-side handlers through managed code



Web Part Connections

- ASP.NET provides Web Part Connection model
 - Provider Web Part supplies data
 - Consumer Web Parts retrieve data
 - WSS provides UI elements to establish connections



Async Processing with Web Parts

- Critical for Web Parts that call across network
 - Async tasks initiated from OnPreRender event

```
protected override void OnPreRender(EventArgs e) {
    // begin async request
    this.Page.RegisterAsyncTask(
        new PageAsyncTask(new BeginEventHandler(BeginXmlRequest),
            new EndEventHandler(EndXmlRequest),
            new EndEventHandler(XmlRequestTimeout),
            null,
            true));
}

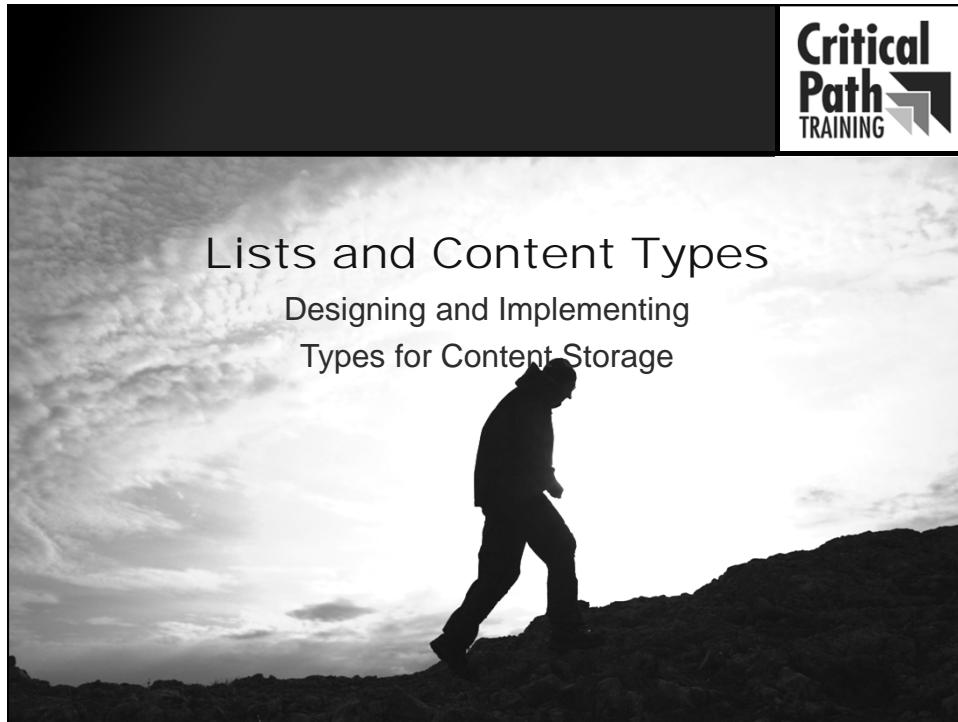
IAsyncResult BeginXmlRequest(object src, EventArgs args,
    AsyncCallback callback, object state) {
    // process task on secondary thread
}

void XmlRequestTimeout(IAsyncResult ar) {
    // deal with timeout scenario
}

void EndXmlRequest(IAsyncResult ar) {
    // finish up task before moving into rendering phase
}
```

Summary

- Developing ASP.NET Web Parts for WSS 3.0
- Persistent Web Part properties
- Importing Web Parts into the Web Part Gallery
- Creating a feature to for deploying Web Parts
- Advanced Web Part Techniques



The background image shows a silhouette of a person walking away from the viewer on a rocky path. The sky is filled with dramatic, swirling clouds. In the top right corner of the slide, there is a white rectangular box containing the Critical Path Training logo.

**Critical
Path
TRAINING**

Lists and Content Types

Designing and Implementing
Types for Content Storage

Agenda

- Content storage enhancements in WSS 3.0
- Querying data in lists
- WSS storage fundamentals
 - Site columns
 - Custom field types
 - Content types
- Provisioning lists and document libraries
- Event handling with receiver classes

Motivation: Content Storage in WSS

- All storage in WSS is based on the concept of lists
 - Everything is modeled in terms of rows and columns
 - The Document Library is really just a hybrid list
- WSS adds value on top of the generic list concept
 - Transparent content storage in SQL Server
 - Automatic generation of the user interface

Platform Storage Enhancements in WSS3

- Parity between lists and document libraries
 - Folders are supported for lists as well as document libraries
 - Versioning is supported for list items as well as documents
 - Events are supported on lists as well as in document libraries
- List and Document Library Enhancements
 - New productivity-oriented built-in field types
 - Wide list support allowing 100s of columns (e.g. surveys)
 - Custom column indexing to improve performance
 - Cross web queries, list views and lookup fields
 - Enhanced versioning with major and minor versions
 - Lists and document libraries automatically support RSS feeds

Accessing List Data

- Updating list data

```
SPLIstItem newItem = list.Items.Add();
newItem["Title"] = "Litware Goes Public!";
newItem["Body"] = "We all live in exciting times.";
newItem["Expires"] = DateTime.Now + TimeSpan.FromDays(2);
newItem.Update();
```

- Enumerating through list items

```
foreach (SPLIstItem item in list.Items) {
    foreach (SPField field in list.Fields) {
        if (field.Hidden != true && !field.ReadOnlyField)
            Console.WriteLine("{0} = {1}", field.Title, item[field.Id]);
    }
}
```

SPQuery

- SPQuery supports CAML-based queries

- Faster access than enumerating through all list items
- Limited to a single list per query

```
SPQuery query = new SPQuery();
query.ViewFields = @"<FieldRef Name='Title' /><FieldRef Name='Expires' />";
query.Query =
@"<Where>
<Lt>
<FieldRef Name='Expires' />
<Value Type='DateTime'>
    <Today />
</Value>
</Lt>
</Where>";

SPLIst list = site.Lists["Litware News"];
SPLIstItemCollection items = list.GetItems(query);
foreach (SPLIstItem expiredItem in items) {
    Console.WriteLine(expiredItem["Title"]);
}
```

SPSiteDataQuery

- SPSiteDataQuery can extend across lists/sites
 - Introduced in WSS 3.0
 - Scope can be Site, SiteCollection or Recursive

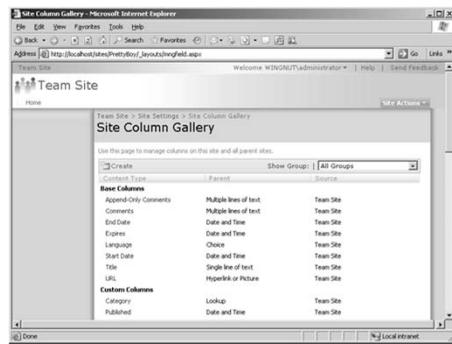
```
SPSiteDataQuery query = new SPSiteDataQuery();
query.Lists = @"<Lists ServerTemplate='104' />";
query.ViewsFolders = @"<Folders FieldRef Name='Title' /><Folders FieldRef Name='Created' />";
query.Webs = "<Webs Scope='SiteCollection' />";
string queryText =
@"
<Where>
  <Eq>
    <FieldRef Name='Created' />
    <Value Type='DateTime'>
      <Today />
    </Value>
  </Eq>
</Where>";
query.Query = queryText;
DataTable table = site.GetSiteData(query);
foreach (DataRow row in table.Rows) {
  Console.WriteLine(row["Title"].ToString()); }
```

Issues with Managing Content

- Problems with managing content in large companies
 - There are many document types identified in an organization, but there is no clear way to enforce standards
 - There's a need to create different types of documents and store them all in one central location
 - Content management applications should make a list of actions available to users depending on the type of content or document
- WSS provides new features to solve these problems
 - Site Columns
 - Content Types

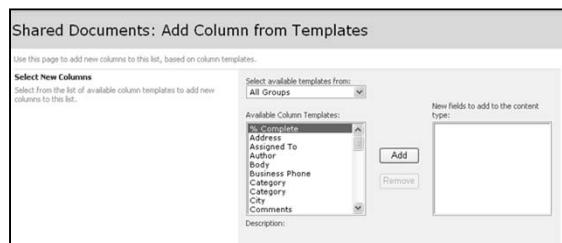
Site Columns

- Site columns are reusable column definitions
 - Site columns can be reused across multiple lists
 - Site columns are scoped to site in the Site Column Gallery
 - Site columns are visible within the site collection to all child sites



Using a Site Column in a List

- Site Columns can be used in List Definitions
 - A Site column represents a reusable, named column definition
 - Site columns are used in lists, document libraries or content types
 - Updates to a site column can optionally be pushed out to lists, document libraries and content types where it has been used



Demo: Creating Site Columns

Litware Inc > Site Settings > Site Column Gallery Site Column Gallery		
Use this page to manage columns on this site and all parent sites.		
Create	Show Group: All Groups	
Site Column	Type	Source
Base Columns		
Append-Only Comments	Multiple lines of text	Litware Inc
Categories	Single line of text	Litware Inc
End Date	Date and Time	Litware Inc
Language	Choice	Litware Inc
Start Date	Date and Time	Litware Inc
URL	Hyperlink or Picture	Litware Inc
Workflow Name	Single line of text	Litware Inc
Core Contact and Calendar Columns		
Address	Multiple lines of text	Litware Inc

Introduction to Content Types

- Foundation for content management in WSS v3
 - Reusable definition for list schema
 - Defines constraints and requirements for an item type
 - Created by users and developers
 - Reused and extended by users

Examples for Content Types

- Proposals for software projects
 - Requires author
 - Requires data for scheduling and budgeting
 - Requires reviews by technical and finance departments
- Customer presentation
 - Requires author
 - Requires reviews by legal and art departments
- Customer report for consulting work
 - Requires consultant name
 - Requires hourly billing information

Content Types

- A content type definition can include...
 - Columns to represent metadata or properties
 - A document template on which to base documents of this type
 - Custom forms for New, Edit, and Display use with content type
 - Event handlers
 - Workflows

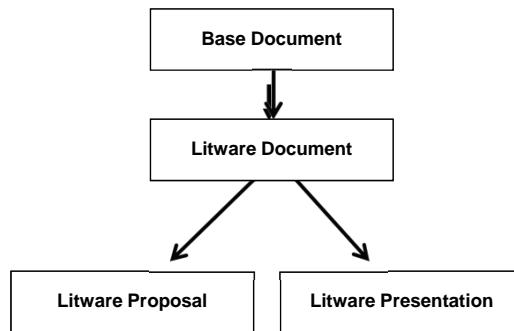
Supporting Multiple Content Types

- Lists can support multiple content types
 - Makes it possible to support heterogeneous content
 - The “New button” becomes a dropdown list
 - Input and display forms change depending on content type



Inheriting Content Types

- Allows base definition reuse across multiple types
 - Core properties can be defined in base content types
 - The Base content type is inherited by more specific content types



Demo: Creating Content Types

Litware Inc > Site Settings > Site Content Type Gallery

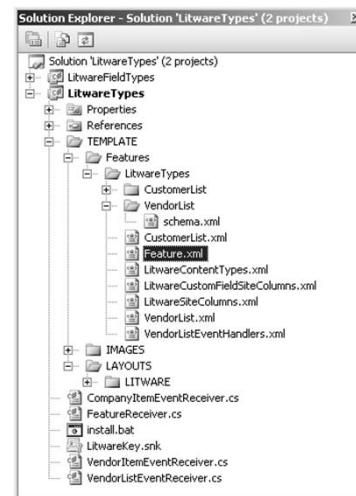
Site Content Type Gallery

Use this page to create and manage content types declared on this site and all parent sites. Content types visible on this page are available for use on this site and its subsites.

Create	Show Group: All Groups	
Site Content-Type	Parent	Source
Document Content Types		
Basic Page	Document	Litware Inc
Document	Item	Litware Inc
Dublin Core Columns	Document	Litware Inc
Form	Document	Litware Inc
Link to a Document	Document	Litware Inc
Master Page	Document	Litware Inc
Picture	Document	Litware Inc
Web Part Page	Basic Page	Litware Inc
Folder Content Types		
Discussion	Folder	Litware Inc
Folder	Item	Litware Inc
List Content Types		
Announcement	Item	Litware Inc
Contact	Item	Litware Inc

Demo: LitwareTypes

- **Important Concepts**
 - Defining WSS types in features using CAML
 - Defining site columns
 - Custom field types
 - Defining content types
 - Defining list schemas
 - Creating event handlers



WSS 3.0 Events

- Events architecture has significantly improved
 - Events are supported for lists, document libraries and content types
 - Events are supported for changes to list schema as well as items
 - Events are supported at site collection and site level
 - Events are supported for incoming email messages
 - Support for synchronous events and asynchronous events
 - Synchronous events occur before the fact and are cancelable
- How do you get events to work
 - Create a custom class inheriting a WSS receiver class
e.g. SPItemEventReceiver or SPWebEventReceiver
 - Compile class into assembly DLL and install in GAC
 - Add event configuration by installing and activating a feature

Item-level Events

Define the receiver class by inheriting from SPItemEventReceiver

```
namespace Litware {
    public class TimesheetEventReceiver : Microsoft.SharePoint.SPItemEventReceiver {
        public override void ItemUpdating(SPItemEventProperties properties) {
            SPWeb web = properties.OpenWeb();
            SPListItem timesheet = web.Lists[properties.ListId].GetItemById(properties.ListItemId);
            // check to make sure date is not day in future
            if (Convert.ToDateTime(timesheet["Submitted On"]).CompareTo(DateTime.Today) > 0) {
                properties.ErrorMessage = "You cannot enter future timesheets";
                properties.Cancel = true;
                return;
            }
        }
    }
}
```

Register receiver class through either OM code or feature element

```
SPList list = web.Lists["Timesheets"];
list.EventReceivers.Add(SPEventReceiverType.ItemAdding,
    "LitwareAssembly, [asm name]",
    "Litware.TimesheetEventReceiver");
```

Summary

- Content storage enhancements in WSS 3.0
- Querying data in lists
- WSS storage fundamentals
 - Site columns
 - Custom field types
 - Content types
- Provisioning lists and document libraries
- Event handling with receiver classes



Forms Services and InfoPath 2007

Designing browser-based forms to
capture schema-validated XML data

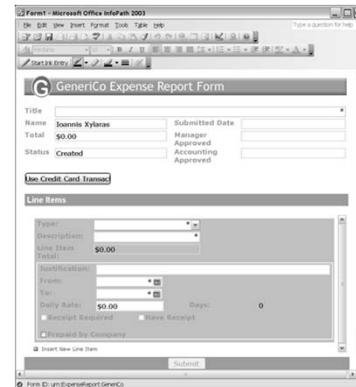


Agenda

- Background in InfoPath 2003
- The InfoPath Forms Designer
- Integration with WSS forms libraries
- Forms Services Architecture
- Designing server-side forms with InfoPath 2007
- Visual Studio Tools for Applications(VSTA)

The Role of InfoPath in Office 2003

- InfoPath was introduced with Office 2003
 - Platform for next generation of electronic forms
- InfoPath Forms
 - Captures XML data
 - Based on XML Schema
 - Requires little/no code

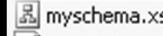
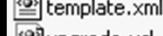
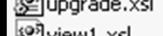


Challenges with InfoPath 2003

- Companies like InfoPath 2003, but...
 - They want better support for offline scenarios
 - They want greater reach (browser-based clients)
 - They want a better code-behind model
- InfoPath 2007 introduces several improvements
 - Improved offline support through wizard
 - Forms Services extended InfoPath forms to browser
 - IT People Responsible for the Deployment
 - Code-behind using Visual Studio Tools Applications

Inside an InfoPath Form

- InfoPath form is a CAB file with .XSN extension
 - Contains manifest with form metadata (XSF)
 - Contains an XML Schema (XSD)
 - Contains XSL transforms for view rendering
 - Contains XML files with data

 manifest.xsf	Microsoft Office InfoPath Form Definition File
 myschema.xsd	XML Schema File
 sampledata.xml	XML Document
 template.xml	XML Document
 upgrade.xsl	XSL Stylesheet
 view1.xsl	XSL Stylesheet

LitwareBugReport.xsn

Security – Trust Levels



Restricted

- Deployed via email, no auto-updates
- No data connections, no managed code
- Not applicable for browser forms



Domain

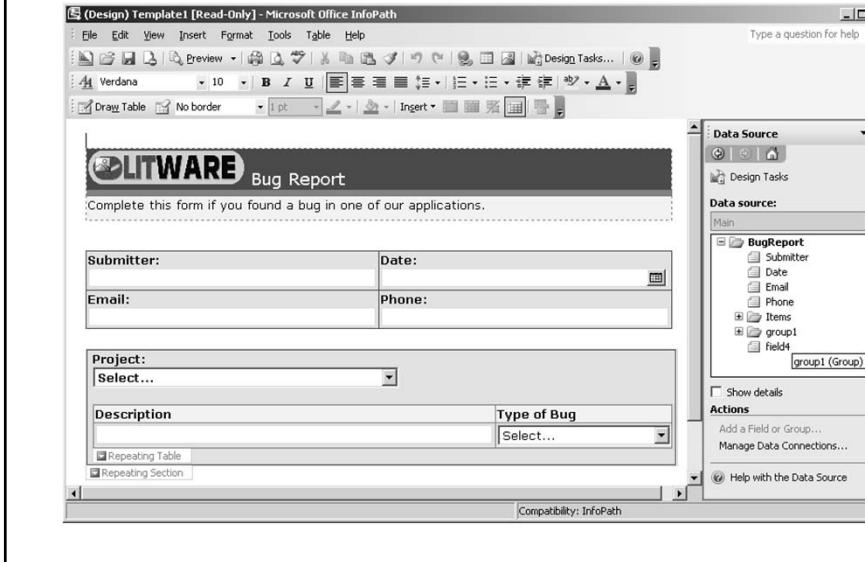
- Deployed to SharePoint library, browser forms
- Connect to own server only, no code for browser forms
- Use trusted Data Connection Library for cross-domain



Full Trust

- Installed, Digitally Signed, or .NET Code Group
- Must be admin-deployed for browser forms
- Connect to any server, managed code in browser forms

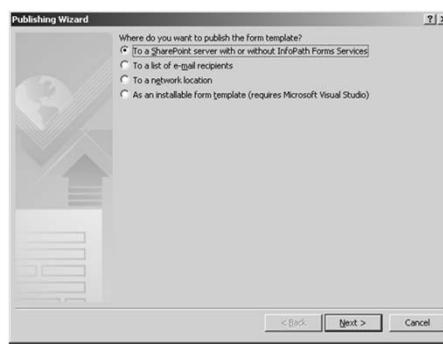
Demo: The InfoPath Forms Designer



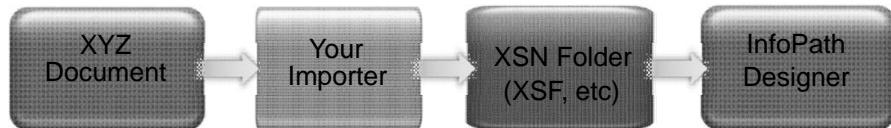
InfoPath Integration with WSS

▪ Forms Libraries

- A document library with a .XSN document template
- Introduced with InfoPath 2003 and WSS 2.0
- Create by users through InfoPath Publishing command

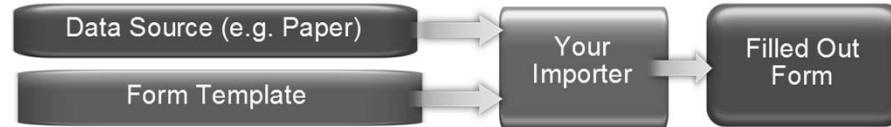


Template Importing



- Built-in support for Word, Excel documents
- Extensible framework
 - Options and progress only
 - IFormTemplateConverter
- Use in combination with the Design Checker

Data Importing



- No OOB solutions for this
- Extensible framework
 - Any custom UI
 - IIInfoPathDataImporter

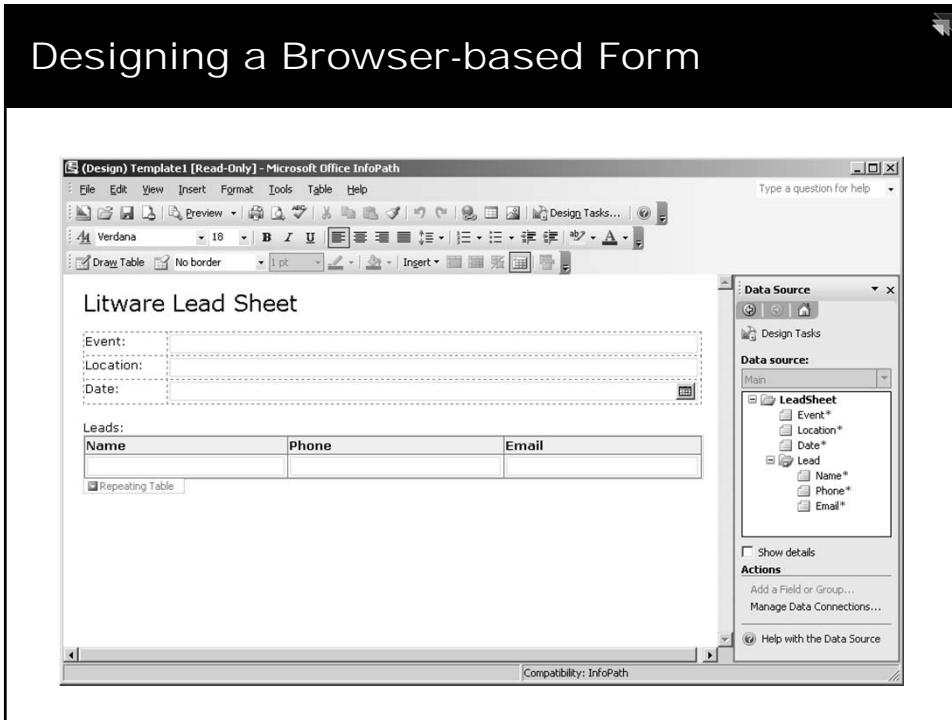
Browser-based Forms

- Forms Services provides HTML rendering
 - Forms must be designed using InfoPath 2007
 - Forms must be designed to be browser compatible

Sites and Browser-based Publishing

- Activate MOSS Standard and Enterprise features
 - Should be done for target site and site collection

Designing a Browser-based Form



Publishing a Browser-based Form

- Saved up to WSS Forms Library
 - MOSS uses document template .xsn file for rendering



Forms Library Settings

- Important Forms Library settings
 - template.xsn is the editable InfoPath form template
 - Open browser-enabled documents
The default is to open with InfoPath rich client if possible

Litware Home > Leadsheets > Settings > Advanced Settings

Form Library Advanced Settings: Leadsheets

Content Types Specify whether to allow the management of content types on this form library. Each content type will appear on the new button and can have a unique set of columns, workflows and other behaviors.	Allow management of content types? <input type="radio"/> Yes <input checked="" type="radio"/> No
Document Template Type the address of a template to use as the basis for all new files created in this document library. When multiple content types are enabled, this setting is managed on a per content type basis. Learn how to set up a template for a library.	Template URL: <input type="text" value="Leadsheets/Forms/template.xsn"/> (Edit Template)
Browser-enabled Documents Specify how to display documents that are enabled for opening both in a browser and a client application. If the client application is unavailable, these documents will always be displayed as Web pages in the browser.	Opening browser-enabled documents <input type="radio"/> Open in the client application <input checked="" type="radio"/> Display as a Web page

Browser-based Rendering

- Browser-based rendering for wide reach
 - Based on DHTML and JavaScript
 - Tested with IE, FireFox, Netscape & hand-held devices

Litware Lead Sheet

Event: Tech Ed Europe 2006

Location: Barcelona

Date: 11/6/2006

Leads:

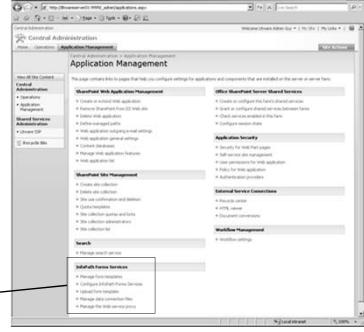
Name	Phone	Email
Bob Hinkle	(123)456-6789	a@b.com
Fred Dinkle	(987)654-3210	c@d.com
Cindy Sprinkle	(866)248-1357	e@f.com

Forms Services Administration

- Part of WSS Central Administration
- Used to upload/manage forms and data connections

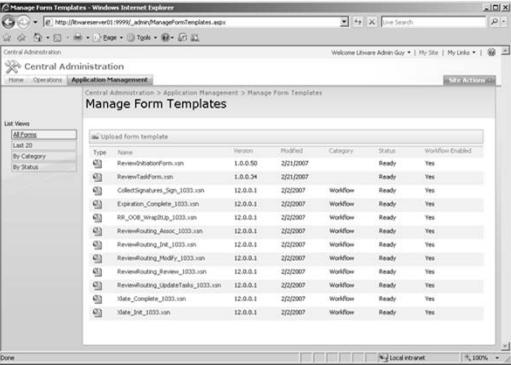
InfoPath Forms Services

- Manage form templates
- Configure InfoPath Forms Services
- Upload form template
- Manage data connection files
- Manage the Web service proxy



Administrator Uploaded Form Templates

- Some forms must be uploaded by administrator
 - Forms with code and/or forms with data connections
 - Benefit: deployed at farm scope not at site scope



Supporting the .NET Developer

- Visual Studio Tools for Applications (VSTA)
 - Provided with InfoPath 2007 out-of-box
 - Lowers the bar for forms with managed code
 - Compatible with Visual Studio Tools for Office
- Visual Studio Tools for Office (VSTO)
 - Embedded designer for professional developers
 - One IDE for all your projects: Workflow, etc.
 - Integrated toolbox, project wizard, etc.

Summary

- Background in InfoPath 2003
- The InfoPath Forms Designer
- Integration with WSS forms libraries
- Forms Services Architecture
- Designing server-side forms with InfoPath 2007
- Visual Studio Tools for Applications(VSTA)



SharePoint Workflows

Using the Windows Workflow Foundation to
Attach Business Logic to Items and Documents



Agenda

- Windows Workflow Foundation (WF) Primer
- Creating WF programs in Visual Studio
- Creating workflow templates for WSS
- Workflow associations and workflow instances
- Creating and waiting on WSS tasks
- Integrating workflow input forms

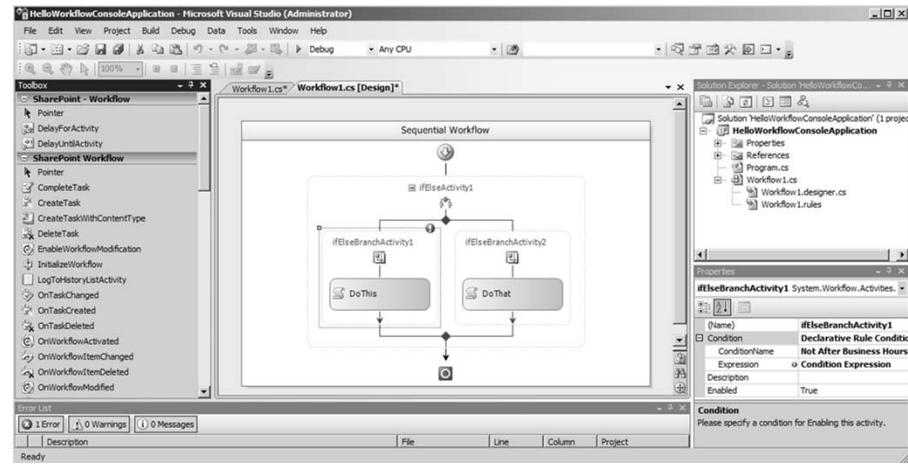
Reactive Programs

- Automating a business process
 - Often requires program with episodic behavior
 - Program waits around and then reacts to some event
- How would you automate document approval?
 - In a Windows Forms application...
 - In an ASP.NET Application

Windows Workflow Foundation (WF)

- What is the Windows Workflow Foundation?
 - Development platform for building reactive programs
 - Set of development tools integrated with Visual Studio
 - Runtime components that ship with .NET FX 3.0
- Windows Workflow Foundation concepts
 - WF program
 - Workflow instance
 - Activities

Visual Studio Workflow Designer

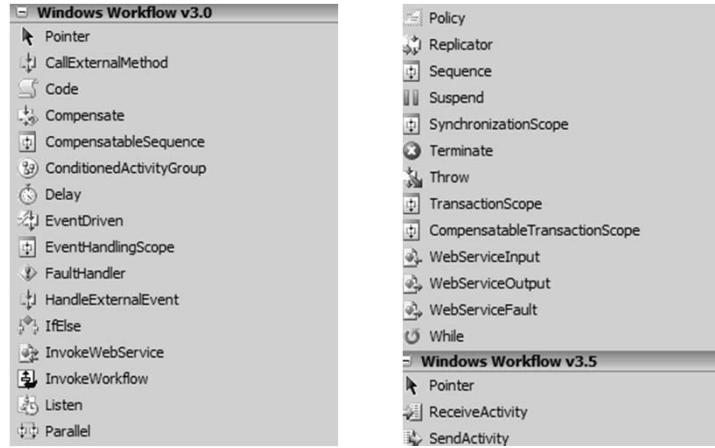


Activities

- An activity is...
 - atomic set instructions used complete a unit of work
 - reusable component used to compose WF programs
- Activities are like controls in forms development
 - You drag and drop them onto a design surface
 - You modify their properties through property sheet
 - You generate event handlers and write code inside
- Activities are different than controls
 - Activities are resumable

WF Base Activity Library

- Standard WF activities provide basic building blocks

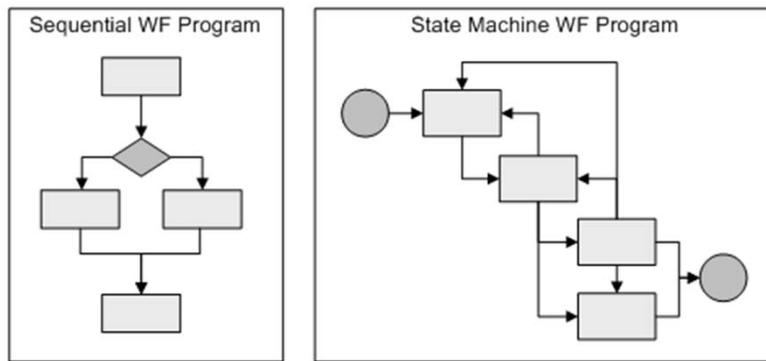


Composite Activities

- Composite Activities can contain children
 - Composite activity controls execution of children
 - Composite activity can encapsulate control-of-flow
 - Examples: IfElse, While, Sequence, Parallel, Replicator
- WF program is itself a composite activity
 - WF program models a tree of activities

WF Program Types

- WF provides two main styles of WF programs
 - Sequential WF program modeled as flow chart
 - State machine WF program models using states



The WF Runtime

```

using System;
using System.Workflow.Runtime;
using System.Workflow.Runtime.Hosting;

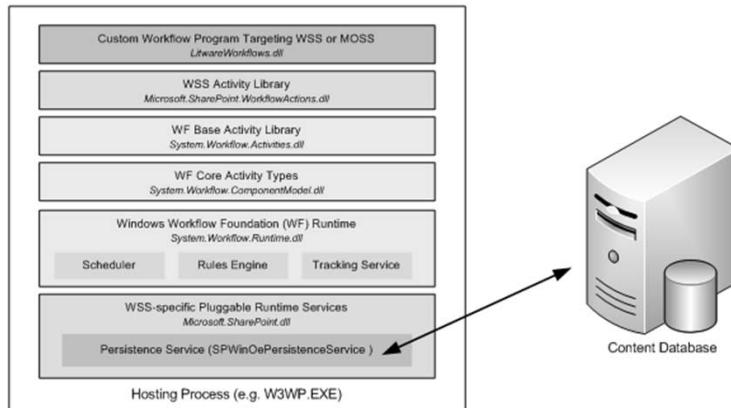
namespace HelloWorkflowConsoleApplication {
    class Program {
        static void Main() {
            // start WF runtime
            using(WorkflowRuntime workflowRuntime = new WorkflowRuntime()) {
                AutoResetEvent waitHandle = new AutoResetEvent(false);
                workflowRuntime.WorkflowCompleted +=
                    delegate(object sender, WorkflowCompletedEventArgs e) {
                        waitHandle.Set();
                    };
                workflowRuntime.WorkflowTerminated +=
                    delegate(object sender, WorkflowTerminatedEventArgs e) {
                        Console.WriteLine(e.Exception.Message);
                        waitHandle.Set();
                    };

                // create and start workflow instance
                WorkflowInstance instance = workflowRuntime.CreateWorkflow(
                    typeof(WorkflowConsoleApplication.Workflow));
                instance.Start();
                waitHandle.WaitOne();
            }
        }
    }
}

```

WF Runtime Services

- Custom services can be written and plugged in
 - WSS provides its own persistence service



SharePoint Workflow Concepts

- Design goals for WF integration with WSS
 - Use WF to attach logic to items and documents
 - Add a human dimension on top of WF
 - Maintain self-service capabilities common in WSS
 - Create strong developer story for custom WF programs
 - Provide valuable WF programs out-of-box with MOSS
- The human dimension
 - Any SharePoint workflow can assign tasks to users
 - Users can see the status of any workflow instance

SharePoint Workflow Actors

- Workflow Template
 - WF Program and optionally workflow input forms
 - A feature to install it inside WSS farm
- Workflow Association
 - Binding of workflow template to list or content type
 - A named instance containing parameterized data
- Workflow Instance
 - A running instance of a WF program attached to an item

Creating a Workflow Association

Add a Workflow: Proposals

Use this page to set up a workflow for this document library.

Workflow Select a workflow to add to this document library. If the workflow template you want does not appear, contact your administrator to get it added to your site collection or workspace.	Select a workflow template: Collect Feedback Collect Signatures Disposition Approval Three-state	Description: Use this workflow to track items in a list.
Name Type a name for this workflow. The name will be used to identify this workflow to users of this document library.	Type a unique name for this workflow: My First Workflow Association	
Task List Select a task list to use with this workflow. You can select an existing task list or request that a new task list be created.	Select a task list: Tasks	Description: Task list for workflow.
History List Select a history list to use with this workflow. You can select an existing history list or request that a new history list be created.	Select a history list: Workflow History	Description: History list for workflow.
Start Options Specify how this workflow can be started.	<input checked="" type="checkbox"/> Allow this workflow to be manually started by an authenticated user with Edit Items Permissions. <input type="checkbox"/> Require Manage Lists Permissions to start the workflow. <input type="checkbox"/> Start this workflow to approve publishing a major version of an item. <input type="checkbox"/> Start this workflow when a new item is created. <input type="checkbox"/> Start this workflow when an item is changed.	

Next **Cancel**

Starting a Workflow Instance

- Users can manually start workflows

The left screenshot shows a 'Proposals' document library in SharePoint. A context menu is open over a document named 'Wingtip Toys Proposal'. The 'Workflows' option in the menu is highlighted with a red arrow. The right screenshot shows the 'Workflows' page for this specific proposal, displaying various workflow options and status information.

The Workflow Status Page

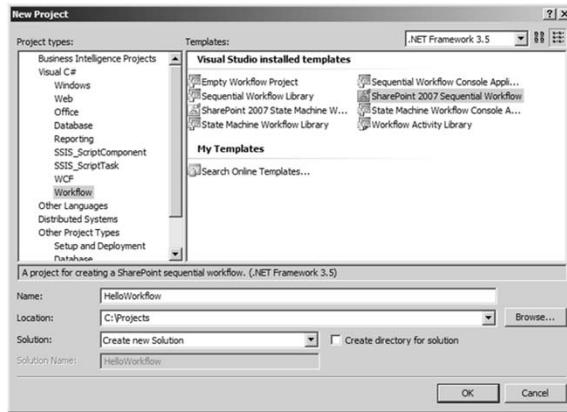
- Any user can see the status of a workflow instance

The screenshot shows the 'Workflow Status' page for a workflow named 'wf1'. It displays the following details:

- Workflow Information:**
 - Initiator: Litware Admin Guy
 - Started: 1/29/2007 2:05 AM
 - Last Run: 1/29/2007 2:05 AM
 - Document: Adventure Works Proposal
 - Status: In Progress
- Tasks:** A task titled "Disposition approval: Adventure Works Proposal.docx" is listed.
- Workflow History:** A table showing event logs with columns: Date Occurred, Event Type, User ID, and Description. There are no items in the history list.

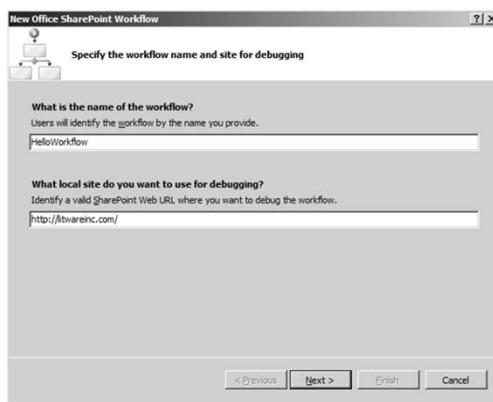
Creating a Workflow Template Project

- Creating a SharePoint Workflow Project in Visual Studio 2008



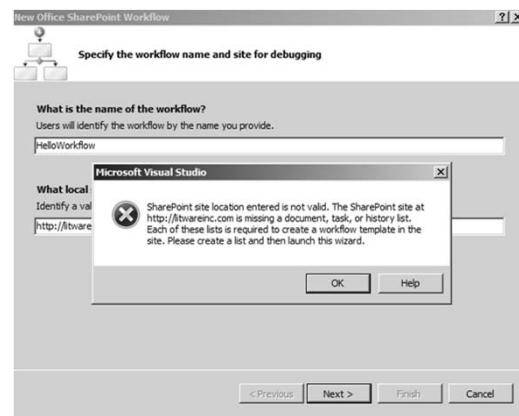
Complete the Wizard

- Step 1 – Specify SharePoint URL
 - Enter the name of the workflow
 - Specify the URL to SharePoint site



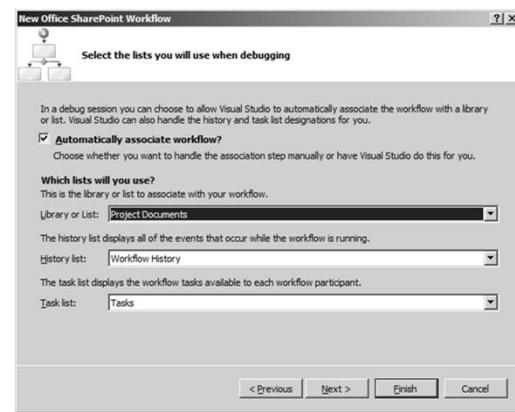
Complete the Wizard

- Following lists need to be available:
 - Document Library
 - Tasks list
 - History list



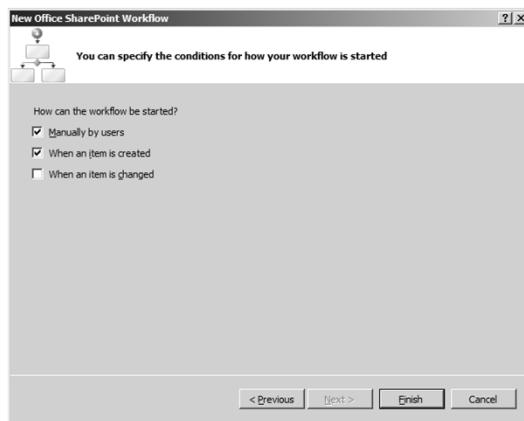
Complete the Wizard

- Step 2 – select the necessary lists
 - List or document library to associate workflow
 - History list
 - Tasks list



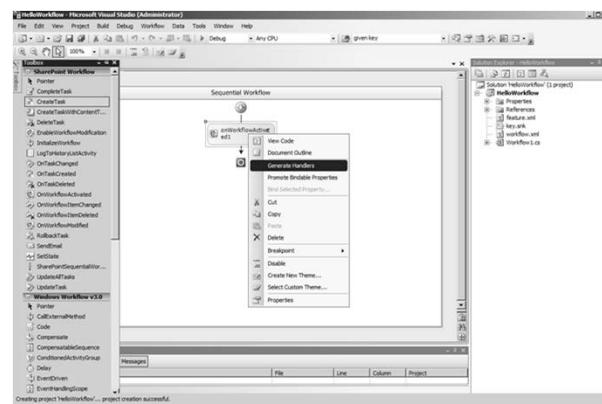
Complete the Wizard

- Step 3 – decision on how to start the workflow



Developing the WF Program

- Getting around inside the Workflow Designer
 - Learn to move between Designer View and Code View
 - Get to Know the Activities in the SharePoint Activity Library



Working in Code View

- Here is what you get as a starting point

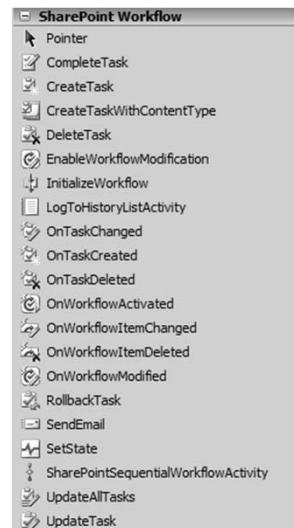
```
using Microsoft.SharePoint.Workflow;
namespace HelloWorkflow {
    public partial class Workflow1 : SharePointSequentialWorkflowActivity {
        // code to call wizard-generate code
        public Workflow1() {
            InitializeComponent();
        }

        // default fields added by project template
        public Guid workflowId = default(System.Guid);
        public SPWorkflowActivationProperties workflowProperties =
            new SPWorkflowActivationProperties();

        // TODO: add fields here
        // TODO: add event handlers here
    }
}
```

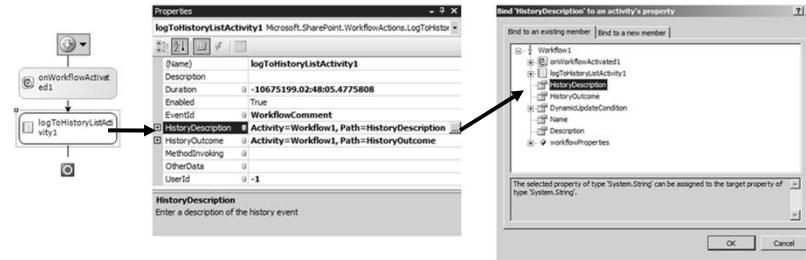
SharePoint Activity Library

- WSS-specific activities used to create SharePoint WF Programs



Data Bound Properties

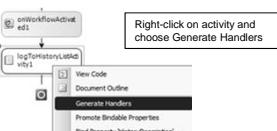
- WF supports data binding of properties
 - Allows for declarative flow of data between activities
 - Used extensively for creating SharePoint WF programs



```
public partial class Workflow1 : SharePointSequentialWorkflowActivity {
    // other members removed for clarity
    public String HistoryDescription;
    public String HistoryOutcome;
}
```

Generating Event Handlers

- Generate event handlers to add code
 - Event handlers can program against WF objects



```
public class Workflow1 : SharePointSequentialWorkflowActivity {
    public SPWorkflowActivationProperties workflowProperties =
        new SPWorkflowActivationProperties();
    public String HistoryDescription;
    public String HistoryOutcome;

    private void LogActivated_MethodInvoking(object sender, EventArgs e) {
        // Generate message using information of current item
        SPListItem item = workflowProperties.Item;
        // determine whether workflow is running on a standard item or a document
        if (item.File == null) {
            HistoryDescription = "Workflow started on item " + item.Title;
        } else {
            HistoryDescription = "Workflow started on document " + item.File.Name;
        }
        HistoryOutcome = "Workflow activation complete";
    }
}
```

Workflow Template Deployment

- Workflow Templates are deployed via Features
 - Feature must be scoped to site collection (Scope=Site)
 - Feature may contain multiple workflow templates

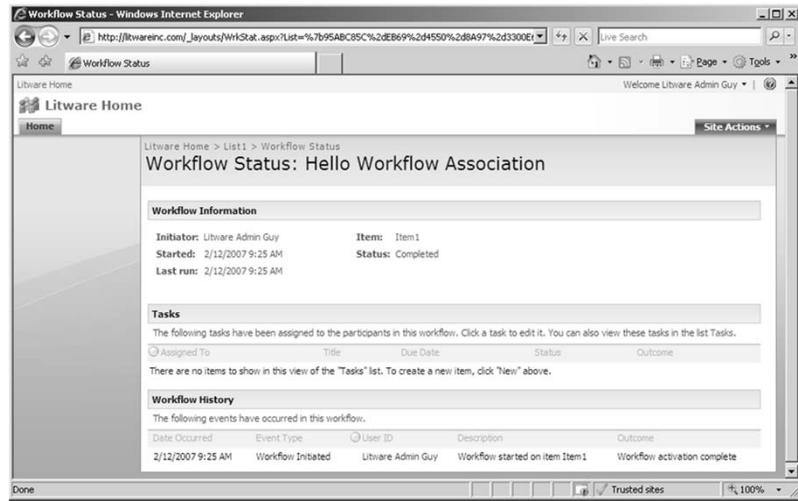
```
<Feature  
  Id="0CEED7AE-D327-41ad-BC33-B3F3F8A4DAD2"  
  Title="Hello World Workflow Template Feature"  
  Description="This feature installs our Hello World Workflow Template"  
  Version="12.0.0.0"  
  Scope="Site"  
  xmlns="http://schemas.microsoft.com/sharepoint/">  
  
  <ElementManifests>  
    <ElementManifest Location="workflow.xml" />  
  </ElementManifests>  
  
</Feature>
```

Workflow Template Definition

- Workflow Element defines Workflow Template
 - Must point to one specific WF program
 - WF program must be compiled into an assembly DLL
 - Assembly DLL must be installed in GAC

```
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">  
  <Workflow  
    Id="1EE1C818-DB7A-4a55-B21B-959D413C6A9C"  
    Name="Hello World Workflow Template"  
    Description="This WF template provides classic Hello World functionality"  
    CodeBaseClass="HelloWorkflow.Workflow1"  
    CodeBaseAssembly="HelloWorkflow, [four-part assembly name]">  
  
    <Categories/><!-- no categories needed -->  
    <MetaData /><!-- no metadata needed -->  
  </Workflow>  
</Elements>
```

Testing 'Hello World' Workflow Template

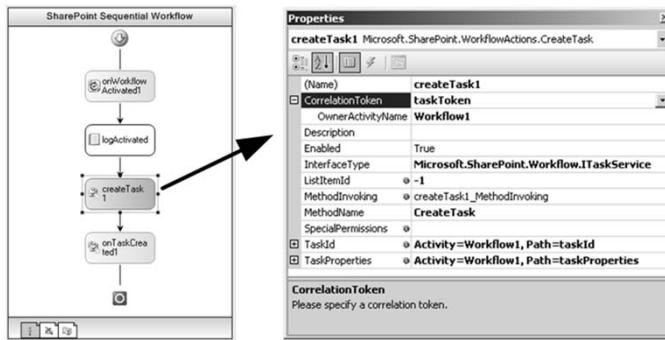


Creating and Waiting on Tasks

- SharePoint Workflows revolve around tasks
 - Represents significant value-add WSS brings to WF
 - Based on standard WSS tasks visible/editable by users
 - Users update tasks through browser or Office programs
 - Your code automatically wakes up and executes
- WSS Tasks are generated with subscriptions
 - WSS encapsulates the listener mechanism
 - WSS registers event handlers behind the scenes
 - You just add event activities and write event handlers

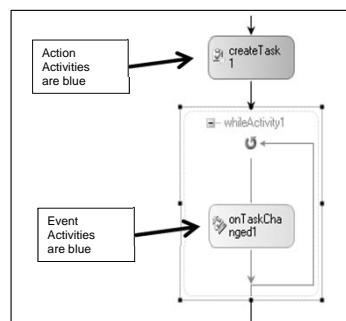
Task GUIDs and Correlation Tokens

- WSS sets up subscriptions for tasks
 - Based on registering event handlers
 - WSS needs way to identify certain task across activities
 - Each task is assigned a GUID and a correlation token



Action Activities vs. Event Activities

- Action activities perform work
 - Their event handlers fire before work is done
- Event activities run code in response to an event
 - Their event handlers run after the event has occurred



Initializing a New Task

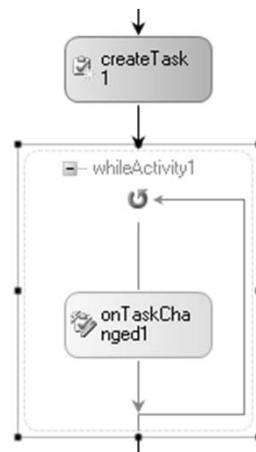
- Add event handler behind CreateTask activity
 - This event handlers fires before task creation
 - Gives you a chance to initialize task properties

```
// these fields are data-bound to properties of task activities
public Guid taskId = default(System.Guid);
public SPWorkflowTaskProperties taskProperties =
    new SPWorkflowTaskProperties();

private void createTask1_MethodInvoking(object sender, EventArgs e) {
    // generate new GUID used to initialize task correlation token
    taskId = Guid.NewGuid();
    // assign initial properties prior to task creation
    taskProperties.Title = "Task for " + workflowProperties.Item.Title;
    taskProperties.Description = "Please review and approve this item.";
    taskProperties.AssignedTo = @"LITWAREINC\BrianC";
    taskProperties.PercentComplete = 0;
    taskProperties.StartDate = DateTime.Today;
    taskProperties.DueDate = DateTime.Today.AddDays(2);
}
```

Waiting on a Task

- Event activity creates subscription
 - OnTaskChanged puts activity to sleep
 - Event handler fires upon modification
- While activity used to control flow
 - While activity loops until task complete



Creating Workflow Forms with ASP.NET

- Workflow input forms can be created in ASP.NET
- Benefits to creating workflow forms with ASP.NET
 - Can run from WSS-only farms
- Drawback to creating forms with ASP.NET
 - More coding involved

ASP.NET Workflow Form Integration

```
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <Workflow
    Name="ListwareWorkflowInfoPath"
    Description="Simple workflow with InfoPath forms."
    Id="48500BEB-D1BE-4ec4-8D21-5DEF76BEEDA8"
    CodeBasename="ListwareWorkflowInfoPath.Workflow1"
    CodeBaseAssembly="ListwareWorkflowInfoPath, [full assembly name]"
    TaskListContentTypeId="0x01080100C9C9515DE4E24001905074F980F93160"
    AssociationUrl="_layouts/Listware/ListwareApprovalAssoc.aspx"
    InstantiationUrl="_layouts/Listware/ListwareApprovalInit.aspx"
    ModificationUrl="_layouts/Listware/ListwareApprovalMod.aspx">
    <standard MOSS task content type>
    <custom application pages>
  </Workflow>
</Elements>
```

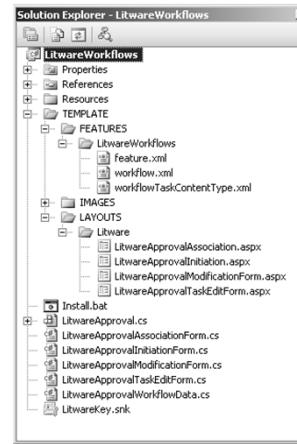
Integrating Workflow Input Forms

▪ Workflow Input Form Types

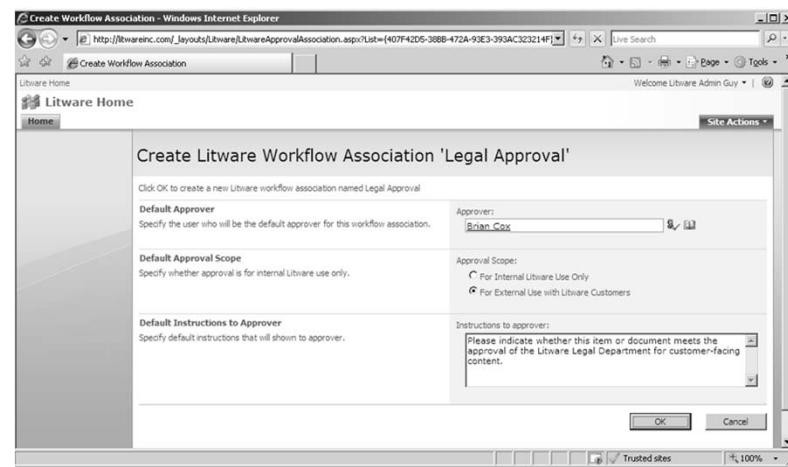
- Association Form
- Initiation Forms
- Modification Forms
- Task Edit Form

▪ Sample Project

- LitwareWorkflows



The Association Form



The Initiation Form

Start New Workflow Instance - Windows Internet Explorer

Start New Workflow Instance

Welcome Litware Admin Guy

Start New Litware Approval Workflow Instance on Item1

Click OK to start a new workflow instance from the Litware Approval workflow template on the item Item1 using the workflow association named Legal Approval.

Approver: Brian Cox

Approval Scope: For External Use with Litware Customers

Instructions to Approver:

Please indicate whether this item or document meets the approval of the Litware Legal Department for customer-facing communication. Make sure to check and then double check all copyright notices.

OK Cancel

Invoking the Modification Form

Workflow Status - Windows Internet Explorer

Workflow Status

Welcome Angela Baraboi

Litware Home > List1 > Workflow Status

Workflow Status: Legal Approval

Workflow Information

Initiator:	Angela Baraboi	Item:	Item1
Started:	2/13/2007 8:39 AM	Status:	In Progress
Last run:	2/13/2007 8:39 AM		

Tasks

The following tasks have been assigned to the participants in this workflow. Click a task to edit it. You can also view these tasks in the list Tasks.

Assigned To	Title	Due Date	Status	Outcome
Brian Cox	Approval required for Item1	2/15/2007	Not Started	

Workflow History

The following events have occurred in this workflow.

Date Occurred	Event Type	User ID	Description	Outcome
2/13/2007 8:39 AM	Workflow Initiated	Angela Baraboi	Workflow created on Item1	Workflow activated
2/13/2007 8:39 AM	Task Created	Angela Baraboi	Approval task assigned to LITWAREINC\brianc	Task status: Not Started

Done Trusted sites 100%

There is one link per modification

Modify this workflow with custom Litware form

The Task Edit Form

The screenshot shows a Windows Internet Explorer window displaying a SharePoint task form. The title bar reads "Approve or Reject Item - Windows Internet Explorer". The main content area is titled "Approve or Reject Item1". It contains several sections: "Item Requiring Approval" with a link to "Item1" in "List1" on the "Litware Home" site; "Instructions to Approver" with a note about copyright guidelines; "Approvers Comments" with a text box containing a comment; and "Approver Comments" again with another text box containing a comment. At the bottom are "Approve", "Reject", and "Cancel" buttons.

Creating Workflow Forms with InfoPath

- Workflow input forms can be created in InfoPath
- Benefits to creating workflow forms with InfoPath
 - Significantly better forms designer experience
 - Significantly less coding
 - Forms can be opened directly with Office client apps
- Drawback to creating forms with InfoPath
 - Workflow template will only run in MOSS farms
 - Workflow template will not run in WSS-only farms

InfoPath Workflow Form Intergration

```

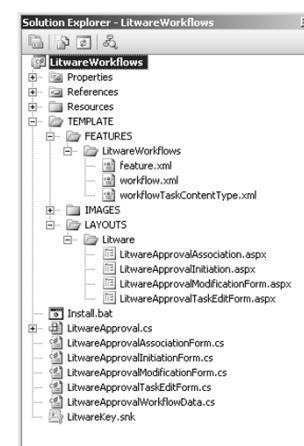
<Elements xml ns="http://schemas.microsoft.com/sharepoint/">
  <Workflow
    Name="LitwareWorkflow1InfoPath"
    Description="Simple workflow with InfoPath forms."
    Id="48500BEB-D1BE-4ec4-8D21-5DEF76BEEDA8"
    CodeBasisClass="LitwareWorkflow1InfoPath.Workflow1"
    CodeBasisAssembly="LitwareWorkflow1InfoPath, [full assembly name]"
    TaskListContentTypeId="0x01080100C9C9515DE4F24001905074F980F93160"
    AssociationUrl="_layouts/CstWrkflIP.aspx"
    InstantiationUrl="_layouts/IniWrkflIP.aspx"
    ModificationUrl=_layouts/ModWrkflIP.aspx">
    <standard MOSS task content type>
    <standard MOSS application pages>
  </Workflow>
</Elements>
urn:schemas-microsoft-com:office:infopath:ReviewInitiationForm2:-myXSD-2005-11-22T23-49-53

```

Integrating Workflow Input Forms

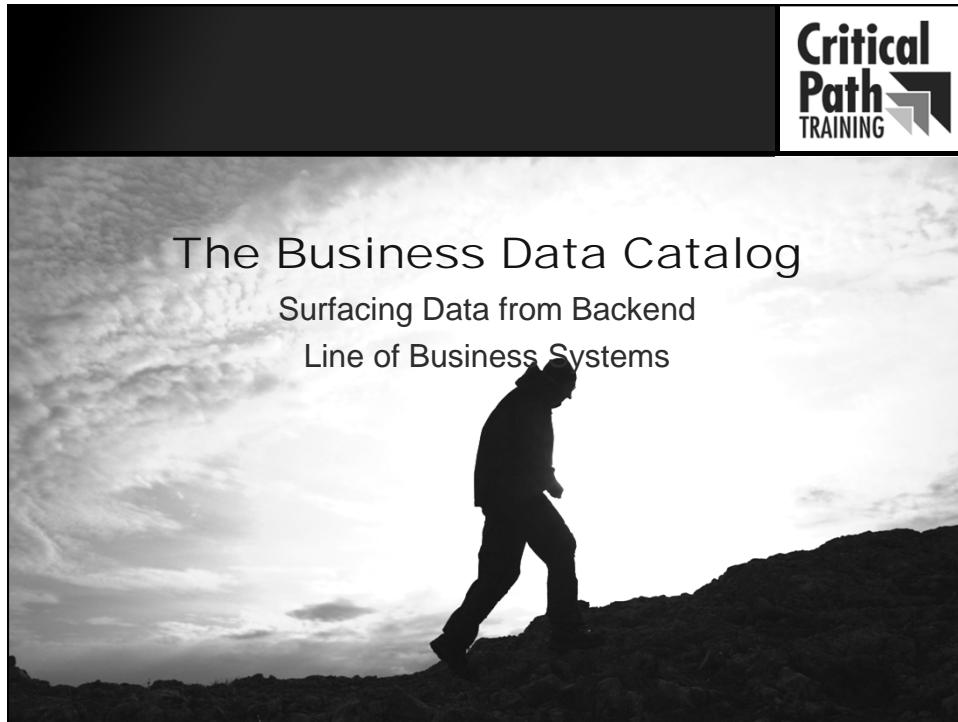
- Workflow Input Form Types
 - Association Form
 - Initiation Forms
 - Modification Forms
 - Task Edit Form

- Sample Project
 - LitwareWorkflows



Summary

- Windows Workflow Foundation (WF) Primer
- Creating WF programs in Visual Studio
- Creating workflow templates for WSS
- Workflow associations and workflow instances
- Creating and waiting on WSS tasks
- Integrating workflow input forms



The Business Data Catalog
Surfacing Data from Backend
Line of Business Systems

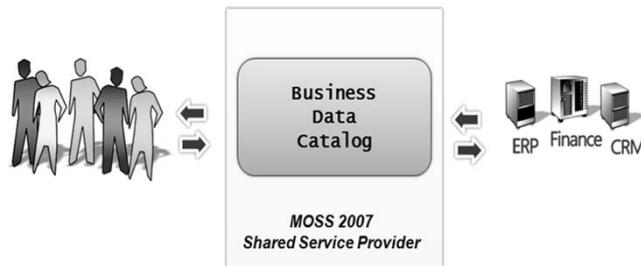


Agenda

- Motivation for the BDC
- Application Definition Files
- Application, Entities, Methods and Associations
- Using the built-in BDC Web Parts
- BDC integration with MOSS search
- Creating custom BDC Web Parts

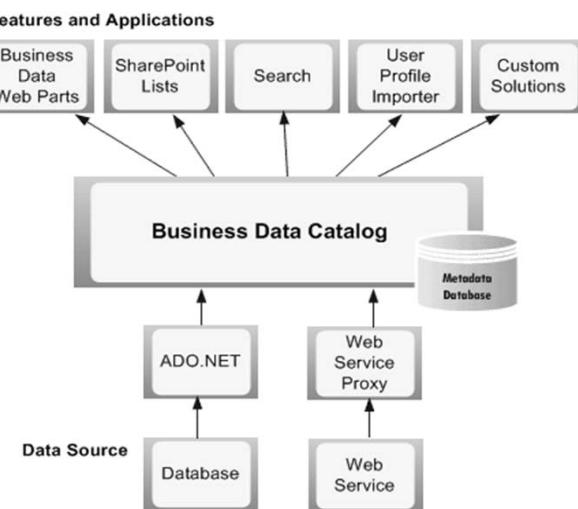
Why do we need the BDC?

- Used to surface data from backend LOB systems
 - LOB data can be shown on any page in a MOSS farm
 - LOB data can be surfaced without writing code



- Note: BDC only provides read-only access to LOB data

BDC Architecture



Application Definition Files

```

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<LobSystem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.microsoft.com/office/sp2005/bdcMetadata BDCMetadata.xsd"
  Type="Database" Version="1.0.0.0" Name="AdventureWorksSample"
  xmlns="http://schemas.microsoft.com/office/sp2005/bdcMetadata">

  <LobSystemInstances>
    <LobSystemInstance Name="AdventureWorksSample">
      <Properties>
        <Property Name="AuthenticationMode" Type="System.String">PassThrough</Property>
        <Property Name="DatabaseAccessProvider" Type="System.String">Sql Server</Property>
      </Properties>
    </LobSystemInstance>
  </LobSystemInstances>

  <Entities>
    <Entity EstimatedInstanceCount="10000" Name="Product"/>
    <Entity EstimatedInstanceCount="10000" Name="SalesOrder"/>
    <Entity EstimatedInstanceCount="10000" Name="Customer"/>
  </Entities>

  <Associations>
    <Association AssociationMethodEntityName="Customer"
      AssociationMethodName="GetSalesOrdersForCustomer"
      AssociationMethodReturnParameterName="SalesOrders"
      Name="CustomerToSalesOrder" IsCached="true">
      <SourceEntity Name="Customer" />
      <DestinationEntity Name="SalesOrder" />
    </Association>
  </Associations>
</LobSystem>

```

Importing Application Definition Files

The screenshot shows the 'Import Application Definition' page within the 'Shared Services Administration' section of Central Administration. The URL in the address bar is 'Litware SSP > Import Application Definition'. The page has a left navigation menu with options like 'View All Site Content', 'Back to Central Administration', 'Shared Services Administration', and 'Litware SSP'. The main content area contains a brief description of what an application definition is, a file input field for selecting the XML file ('F:\emos\l11_BDC\BDC_AppDef\AdventureWorks_HR.xml'), and several configuration options under 'File Type' (radio buttons for 'Model' and 'Resource') and 'Resources to import' (checkboxes for 'Localized Names', 'Properties', and 'Permissions'). At the bottom are 'Import' and 'Cancel' buttons.

Administration of BDC Applications

The screenshot shows the 'Shared Services Administration: Litware SSP' page. In the center, under 'View Application: AdventureWorks_HR', there is an 'Application Information' section with the following details:

Name:	AdventureWorks_HR
Type:	Database
Definition Version:	1.0.0.0
Access Account:	Logged-on user
Maximum Concurrent Connections:	Unlimited

Below this are links for 'Manage Permissions', 'Export Application Definition', and 'Delete Application'. Under the 'Entities' section, there are three items: Name (with a small icon), Department, and Employee.

Examining BDC Application Entities

The screenshot shows the 'Shared Services Administration: Litware SSP' page. In the center, under 'View Entity: Department', there is an 'Entity Information' section with the following details:

Name:	Department
Application:	AdventureWorks_HR
Crawlable:	No

Below this is a 'Fields(of default view)' section with three fields: Name (Type: System.String, Title: X, Display by Default: Yes), Department (Type: System.String), and Department ID (Type: System.Int16).

Under 'Relationships', there is one entry: Name (Related To: Employee, Role in Relationship: Source). Under 'Actions', there is one entry: Name (Type: Default: X, Value: http://ssp.litwareinc.com:80/ssp/admin/Content/Department.aspx?DepartmentID={0}). Under 'Filters(of finder method)', there are two entries: Name (Type: ComparisonFilter) and Name (Type: WildcardFilter).

Adding Actions to an Entity

The screenshot shows the 'Add Action' dialog box within the 'Shared Services Administration: Litware SSP' interface. The path in the breadcrumb navigation is 'Litware SSP > Business Data Catalog Applications > AdventureWorks_HR > Department > Add Action'. The dialog has several sections:

- Name:** A text input field labeled 'Action Name:' containing the value 'Open data in custom page'.
- URL:** A text input field labeled 'Navigate to this URL:' containing the value '[http://example.com/edit.aspx?id=0]'. Below it is a note: 'Example: http://example.com/edit.aspx?id=0'.
- URL Parameters:** A section where you can assign properties to parameters in the URL. It shows one parameter: '0' with 'DepartmentID' selected as the property.
- Icon:** A section where you can choose an icon. It shows three options: 'No icon' (radio button), 'Standard icon: Edit' (radio button selected), and 'The image at this URL:' (text input field).

At the bottom right are 'OK' and 'Cancel' buttons.

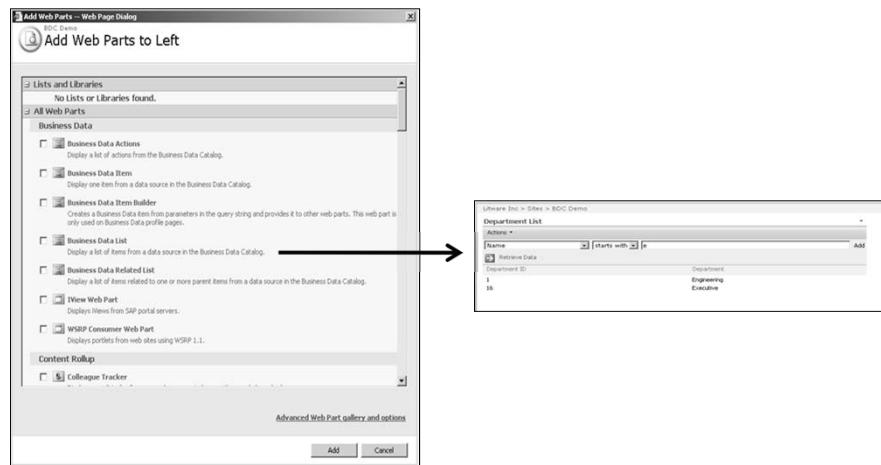
Administrating Security

The screenshot shows the 'Manage Permissions: Department' page within the 'Shared Services Administration: Litware SSP' interface. The path in the breadcrumb navigation is 'Litware SSP > Business Data Catalog Applications > AdventureWorks_HR > Department > Manage Permissions'. The page title is 'Manage Permissions: Department'. It includes a table showing permissions for a user/group:

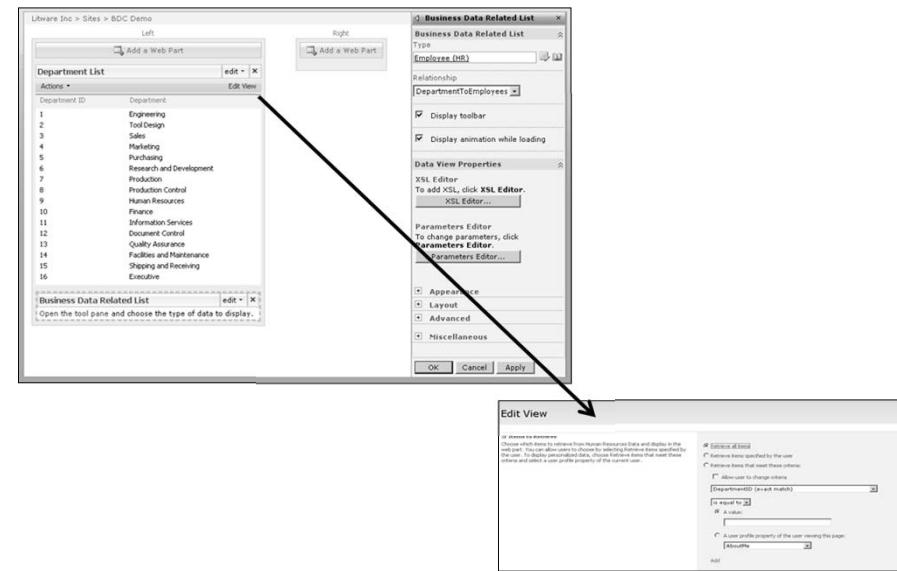
User/Group Name	Rights
LITWAREINC\administrator	Edit, Execute, Selectable in clients, Set Permissions

Below the table are buttons for 'Add Users/Groups', 'Remove Selected Users', 'Modify Permissions of Selected Users', and 'Copy all permissions to descendants'.

Using BDC Web Parts



Editing BDC Web Parts



Connecting Web Parts with Associations

The screenshot shows two SharePoint lists side-by-side. On the left is the 'Department List' with columns 'Actions', 'Department ID', and 'Department'. The data includes departments like Engineering, Tool Design, Sales, Marketing, Purchasing, Research and Development, Production, Production Control, Human Resources, Finance, Information Services, Document Control, Quality Assurance, Facilities and Maintenance, Shipping and Receiving, and Executive. On the right is the 'Employee List' with columns 'Actions', 'Employee ID', 'First Name', 'Last Name', 'Job Title', and 'Department'. The data includes employees like Kevin Brown, David Bradley, Sariah Hampadoungsataya, Mary Gibson, Jill Williams, Terry Eminhizer, Wanda Benshoof, John Wood, Mary Dempsey, and others. Arrows between the lists indicate associations between specific department and employee entries.

Searching through BDC Applications

The screenshot shows the 'Add Content Source' page under 'Search Settings > Content Sources'. The page has several sections:

- Name:** Employees
- Content Source Type:** Business Data (selected)
- Applications:** Crawl selected applications (Adventureworks_HR selected)
- Crawl Schedules:** Full crawl (None selected), Incremental crawl (None selected)
- Start Full Crawl:** Start full crawl of this content source (checkbox)

Adding BDC Columns to WSS Lists

The screenshot shows the 'Create Column' dialog box for a 'Colleagues' list. The 'Name and Type' section has 'Column name:' set to 'Department'. The 'Type a name for this column, and select the type of information you want to store in the column.' section is visible. The 'Additional Column Settings' section includes fields for 'Description:', 'Require that this column contains information:' (set to 'Yes'), 'Type:' (set to 'Department (DB)'), and 'Display this field of the selected type:' (set to 'Department'). There are also checkboxes for 'Display the actions menu' and 'Link this column to the profile page'. At the bottom, there are options to 'Add a column to show each of these additional fields:' (with 'Department' checked) and 'Add to default view' (with a checked checkbox).

Using the BDC API

- The BDC provides two main APIs
 - One is to administrate BDC metadata
 - The other is for access BDC application data

Summary

- Motivation for the BDC
- Application Definition Files
- Application, Entities, Methods and Associations
- Using the built-in BDC Web Parts
- BDC integration with MOSS search
- Creating custom BDC Web Parts



The background image shows a silhouette of a person walking away from the viewer on a rocky path. The path leads towards a dark, hilly landscape under a sky filled with scattered clouds. The overall mood is contemplative and professional.

**Critical
Path
TRAINING**

Web Content Management
Managing Content within Internet-facing Sites
using MOSS Publishing Portals

Agenda

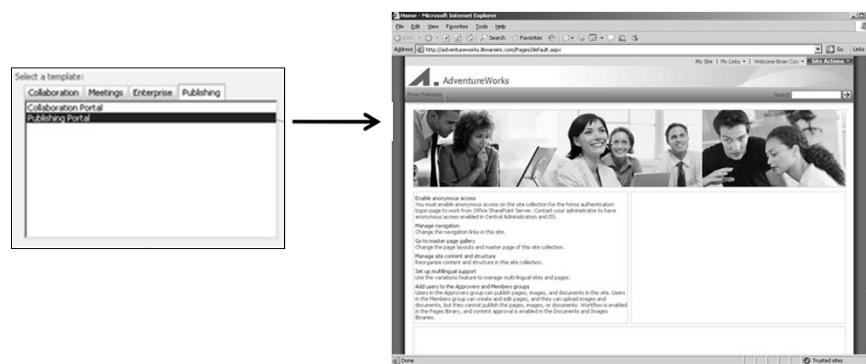
- The Publishing Site template
- The MOSS Approval Process
- Creating custom page layouts
- Converting Office documents
- Content Translation using Variations
- Optimization through Caching Profiles

MOSS WCM Features

- **Branding**
 - Define the look, feel, and navigation of the site
- **Decentralized Authoring**
 - Allow users to easily create and contribute content
- **Workflow/Scheduling**
 - Supervisors approve content before it is posted.
- **Data Integrity**
 - Enforce validation of content structure for publishing
 - Ensure content published/removed in timely manner

Creating A Publishing Portal

- Creating with WSS Central Administration
 - Create a site collection based on Publishing Portal

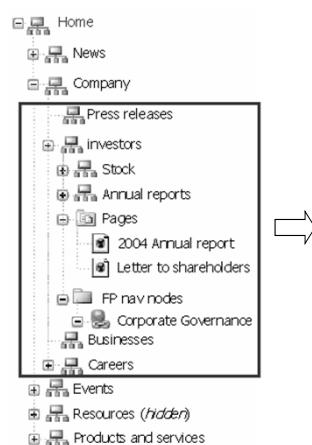


Site Hierarchy

- In the past a lot of confusion
 - Windows SharePoint Services 2003 → sites
 - SharePoint Portal Server 2003 → areas
 - Content Management Server 2002 → channels
- In SharePoint 2007 everything is a site

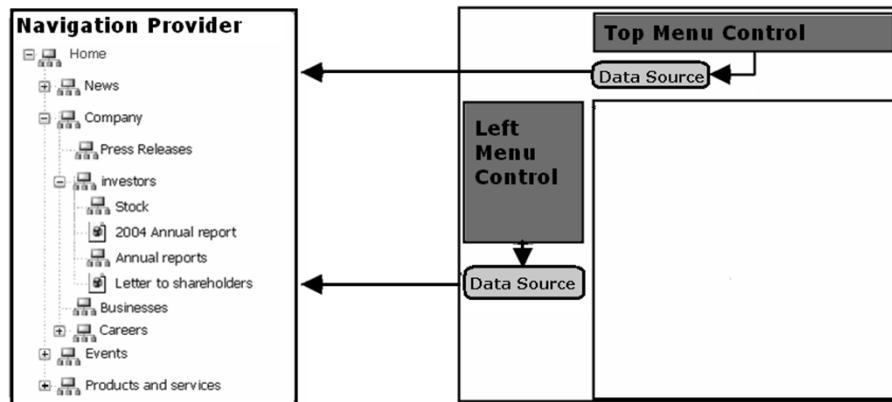


Navigation



- Dynamic navigation based on site hierarchy
- Includes webs, pages and authored links
- Navigation links trimmed based on security, workflow state and publishing schedule

Navigation and ASP.NET



- Based on the ASP.NET 2.0 navigation model
- Works with standard ASP.NET 2.0 navigation controls
- Out-of-the-box Hierarchy navigation provider
- High performance: provider support runtime object caching

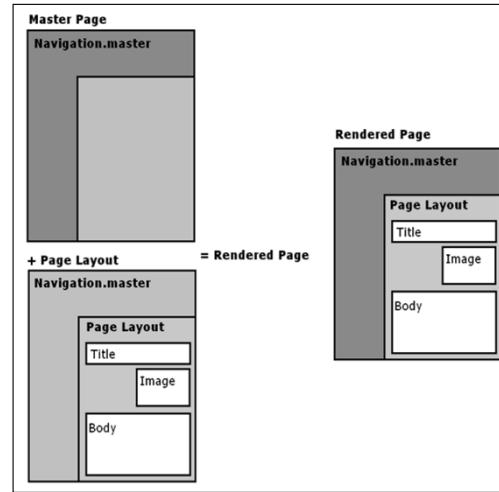
Site Content and Structure

Type	Name	Modified	Modified By	Approval Status	Scheduling Start Date	Scheduling End Date	Contact	Page Layout
File	default (default)	4/23/2006 1:18 PM	System Account	Approved				/catalog/masterpage/Welcome (1.0)

Type	Relationship	Title	Located In	Modified
File	Uses	WelcomeLinks.aspx	Litware Internet Site > Master Page Gallery	4/23/2006 1:17 PM
Image	Uses	PR.gif	Litware Internet Site > Images	4/23/2006 1:18 PM

Page = Master Page + Page Layout

- Master page defines banner and navigation
- Page layout ASPX defines how page content is rendered
- Possible scenario
 - 1-3 Master pages
 - 10-25 Page Layouts
 - 10s of 1000s of Content Pages

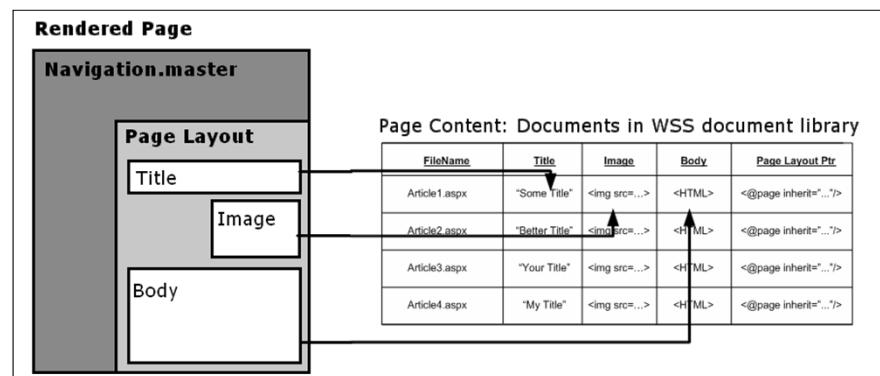


Page Layouts

- Page execution:
- Page URL requested
- Page layout executed in content of page
- Content server controls bind to page fields
- Rendered page returned

Inherited from WSS:

- Versioning,
- Check-in/Check-out
- Content types
- Access control
- Workflow



Steps to Create a New Page Layout

- Create shared columns
- Create content type
- Add created site columns to content type
- In the Master Page Gallery
 - Create new Page Layout file
 - Check-out file and edit in SharePoint Designer
 - Populate the file with content fields
 - Check-in and approve
- Use the new page layout file

Steps to Create a New Page Layout

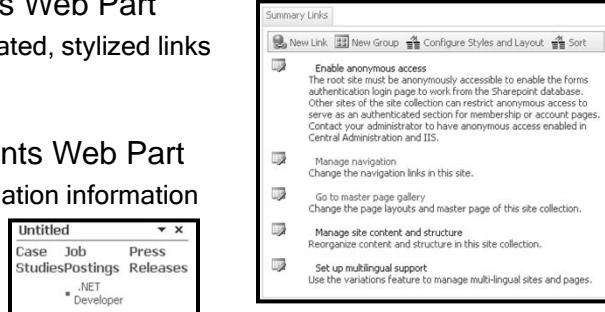
- Create site columns
- Create content type
- Add created site columns to content type
- In the Master Page Gallery
 - Create new Page Layout file
 - Check-out file and edit in SharePoint Designer
 - Populate the file with content fields
 - Check-in and approve
- Use the new page layout file

Publishing Cycle

- Workflow based on Windows Workflow Foundation
- Light-weight approval workflow is active OOB
 - Based on approval
 - Minor versions need to be approved to become major versions
 - Visitors only see the major (published) versions
- Workflow can be replaced by custom workflow
 - OOB delivered with MOSS 2007
 - Designed using SharePoint Designer 2007
 - Created using Visual Studio.NET 2005

WCM Web Parts

- Summary Links Web Part
 - Custom annotated, stylized links
- Table of Contents Web Part
 - Displays navigation information of your site
- Content Query Web Part
 - Displays a dynamic view of the content in your site

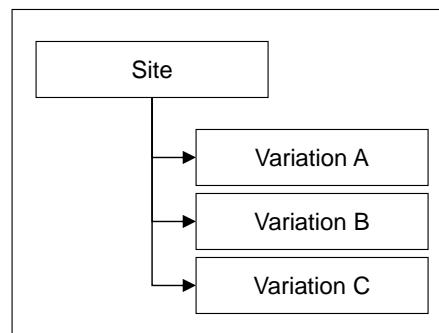


Multilingual sites

- Common pattern
 - Parallel sites in multiple languages
 - In concept, they are localized mirrors
 - In reality, there are exceptions and customizations for different regions
- Modeled as variations
 - Admin creates multiple labels
 - System creates and maintains parallel versions of containers and items
 - Exceptions are allowed
- Not just for language translations
 - Multilingual sites, multi-device sites, and multi-branded sites

Site Variations

- Allow for publishing of related sites and pages
 - Multilingual scenario
 - Device targeting



Profile Caches

Litware Publishing Site > Site Settings > Site Collection Output Cache Settings

Site Collection Output Cache Settings

Configure site-wide cache settings.

Output Cache Select the Enable output cache check box to enable output caching in this site collection.	<input checked="" type="checkbox"/> Enable output cache
Default Page Output Cache Profile A cache profile specifies how long items should be held in the cache. It also describes to the caching system how to determine whether a cached page element is in fact valid for other requests for the same element from different users. You can specify different cache profiles to use for anonymous and authenticated users. This optimizes the use of the cache based on the authentication methods allowed on the site. Page output cache profiles specifically affect portal publishing pages. Show me more information.	Anonymous Cache Profile Public Internet (Purely Anonymous)
	Optimized for public Internet facing sites or areas that are meant to serve the same content to all users. No authentication check is done and any user requesting a page receives the same page as any other user.
	Authenticated Cache Profile Disabled Caching is not enabled
Page Output Cache Policy You can allow administrators and page layout designers to choose a different page output cache profile from the profile specified above.	Publishing Sites: <input type="checkbox"/> Publishing sites can use a different page output cache profile Page Layouts: <input type="checkbox"/> Page layouts can use a different page output cache profile

Configuring Document Conversion

Central Administration > Operations > Services on Server

Services on Server: OSS1

Select the role that most closely matches how this server will be used. After selecting the role, start all highlighted services in the list below.

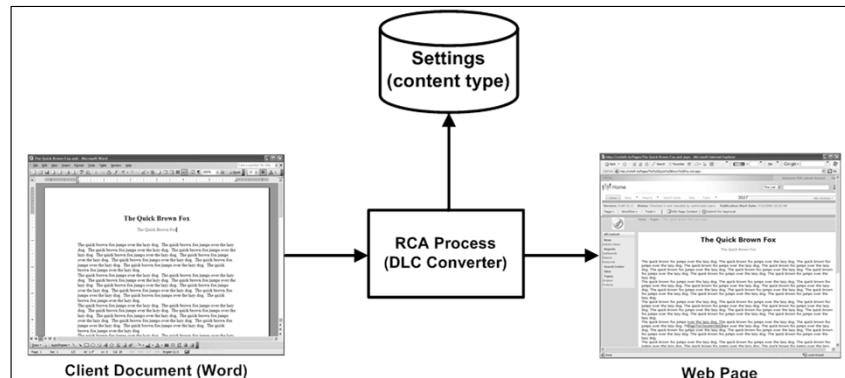
Service	Description	Status	Action
Central Administration	All services run on this server	Started	Stop
Document Conversions Launcher Service	Web Application and Search Query services run on this server	Stopped	Start
Document Conversions Load Balancer Service	Search Indexing service runs on this server	Stopped	Start
Excel Calculation Services	Excel Calculation service runs on this server	Started	Stop
Office SharePoint Server 2007 (Beta) Search Service [Index: On; Query: On]	Services you choose run on this server	Started	Stop
Windows SharePoint Services Administration	All services run on this server	Started	Stop
Windows SharePoint Services Incoming E-Mail	Web Application and Search Query services run on this server	Started	Stop
Windows SharePoint Services Search Service	Search Indexing service runs on this server	Started	Stop
Windows SharePoint Services Web Application	Excel Calculation service runs on this server	Started	Stop

Indicates required service which is not yet enabled on any server in the farm.
Indicates the required service has been started on one or more servers in the farm.

Configuring Document Conversion



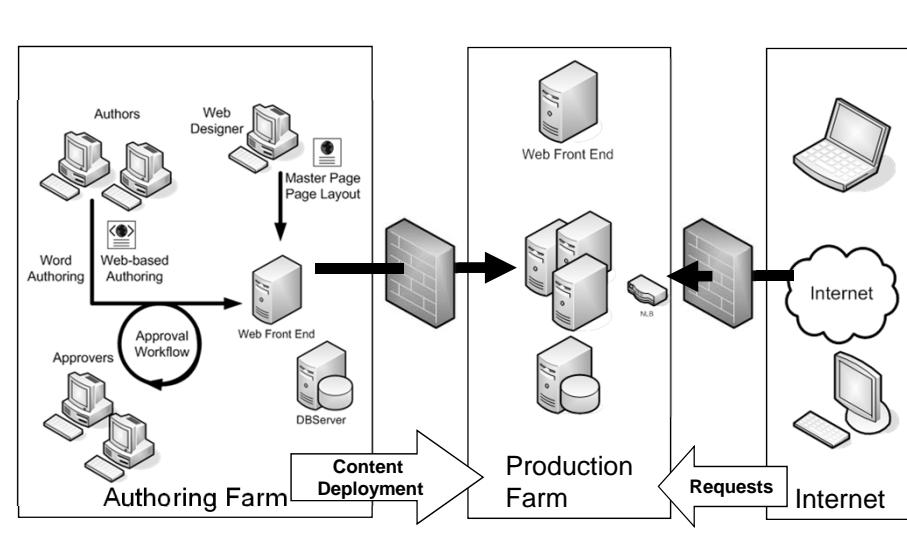
Smart Client Authoring



- Convert documents of different formats into web pages (server side)
- Linkage between document and page preserved
- Document Content Type stores info about how and where to create page
- Pluggable converter model
 - Builds on document services conversion infrastructure
 - Out-of-the-box converter for Word documents

Content Deployment

- Transfers content from one site collection to another
 - Paths define the relationship between source and destination
 - Jobs define the content to deploy and a schedule



Summary

- The Publishing Site template
- The MOSS Approval Process
- Creating custom page layouts
- Converting Office documents
- Content Translation using Variations
- Optimization through Caching Profiles



The background image shows a silhouette of a person walking up a rocky hill against a backdrop of a cloudy sky at sunset or sunrise. The Critical Path Training logo is in the top right corner.

**SharePoint
Application Security**
Securing Your SharePoint Business Solutions

Agenda

- Authenticating users in WSS and MOSS
- Configuring access control within a site collection
- Configuring access control for Web Application
- Using the MOSS single sign-on service (SSO)

Security 101

- Authentication and Identity
 - Authentication based on an identity
(i.e. Security Principal)
 - Authentication performed using credentials
 - Authentication produces some form of badge
- Authorization and Access Control
 - A subsystem is used to define security policy
 - Privileged users to configure security policy on objects
 - Subsystem enforces policy at run time

WSS Identities

- IIS Application Pool Identity
 - Configured with IIS or WSS administration tools
 - Authenticated when IIS worker process is launched
- WSS System Identity
 - New concept introduced with WSS 3.0
 - Used by WSS to hide application pool identity
- User Identity
 - Used for authorization and auditing
 - Authenticated by Windows or Forms Auth Provider

Application Pool Identity

- WSS runtime is hosted by IIS Application Pools
 - Each WSS Web Application runs in a IIS Web site
 - Each IIS Web site runs with in a specific IIS application pool
 - Application pool identity configured with local or domain account
 - Domain account recommended in farms of two or more servers

The screenshot shows the 'Application Pools' section of the IIS Manager. A table lists several application pools:

Name	Status	.NET Frame...	Managed Pipe...	Identity
Classic .NET AppPool	Started	v2.0	Classic	NetworkService
DefaultAppPool	Started	v2.0	Integrated	NetworkService
Ltware SSP	Started	v2.0	Classic	LTWAREINC\sys-s...
OfficeServerApplicationPool	Started	v2.0	Classic	NetworkService
SharePoint - Ltware SSP	Started	v2.0	Classic	ltwareinc\sys-spSh...
SharePoint Central Administration v3	Started	v2.0	Classic	ltwareinc\sys-share...
SharePoint Default App Pool	Started	v2.0	Classic	ltwareinc\sys-spWo...

The 'SharePoint Default App Pool' row is highlighted. An 'Advanced Settings' dialog is open over the table, showing the following configuration:

- General**: .NET Framework Version: v2.0, Managed Pipeline Mode: Classic, Name: SharePoint Default App Pool, Queue Length: 1000, Start Automatically: True.
- CPU**: Limit: 0, Limit Action: NoAction, Limit Interval (minutes): 5, Processor Affinity Enabled: False, Processor Affinity Mask: 4294967295.
- Process Model**: Identity: ltwareinc\sys-spWorkerP, Idle Time-out (minutes): 0, Load User Profile: False, Maximum Worker Processes: 1.

WSS Authentication with SQL Server

- WSS system code must access SQL Server
 - WSS must create and access configuration database
 - WSS must create and access content databases
- WSS must authenticate against SQL Server
 - Option #1: Integrated Windows Authentication (recommended)
 - Option #2: Standard SQL authentication

Accessing SQL Server

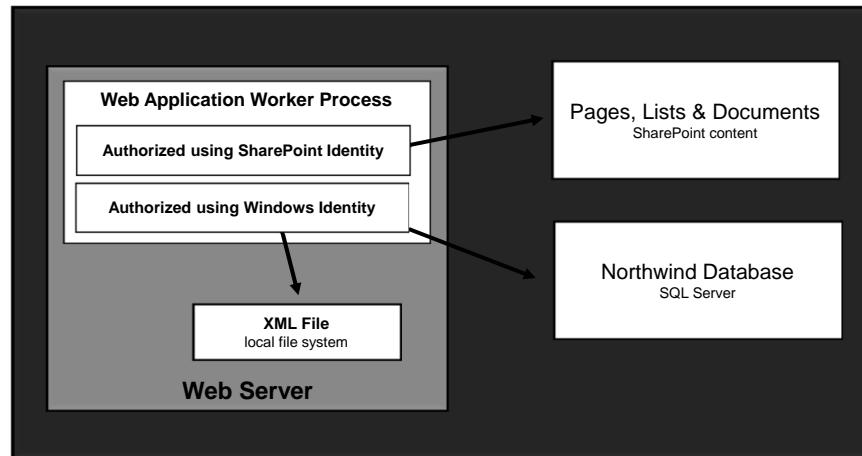
- What Identity is used to access SQL Server?
 - WSS system code accesses configuration database and content databases with Application Pool Identity
 - Custom code (e.g. Web Parts) run with impersonation and access SQL Server database with user identity

SHAREPOINT\System Account

- WSS V2 had issues with Application Pool Identity
- WSS 3.0 introduces SHAREPOINT\system account
 - Hides IIS Application Pool Identity from users
 - Runs as God within WSS authorization system
 - Removes need to treat Application Pool Identity as site user
- System account details for the super geek
 - SID as S-1-0-0 (Null SID)
 - User id as 1073741823 (0x3FFFFFFF)
 - Account is a internal WSS identity and NOT a Windows identity

WSS Identity versus Windows Identity

- It's important to understand the difference



Elevation of Privileges

- Code typically runs under identity of user
 - Authorization works as expected in SharePoint
 - Sometime code must do things which user cannot do
- Custom code elevate privilege
 - Advantage: elevated code can do what it wants
 - Disadvantage: elevated code can do what it wants

```
public void MyCustomWebPartCode() {
    // this code runs as Bob the user
    SPSecurity.RunWithElevatedPrivileges(delegate() {
        // this code runs as SYSTEM\SHAREPOINT
        // this code uses Application Pool identity not user identity
    });
}
```

Accessing Sites with Elevated Privileges

- Accessing sites with WSS object is tricky
 - Must create new SPSite object after elevating

```
public void MyCustomWebPartCode() {
    // objects in SPContext created under Bob's identity
    SPSite siteCollection = SPContext.Current.Site;
    SPWeb site = SPContext.Current.Web;

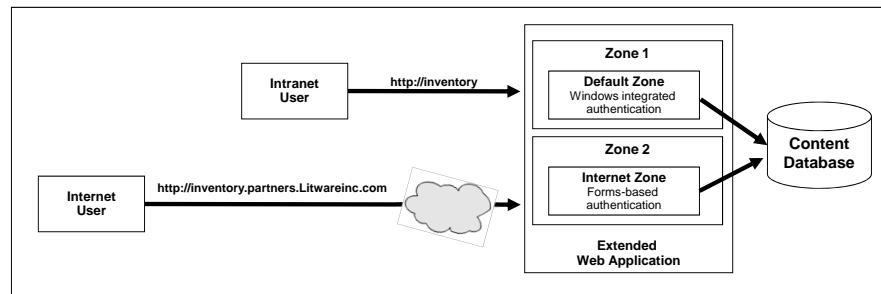
    SPSecurity.RunWithElevatedPrivileges(delegate() {
        // create new object while running with SYSTEM\SHAREPOINT identity
        using (SPSite ElevatedsiteCollection = new SPSite(siteCollection.ID)) {
            using (SPWeb Elevatedsite = ElevatedsiteCollection.OpenWeb(site.ID)) {
                string s1 = Elevatedsite.Owner.Name;
                string s2 = Elevatedsite.Usage.Visits.ToString();
                string s3 = Elevatedsite.RootFolder.Audit.GetEntries().Count.ToString();
            }
        });
    });
}
```

WSS Authentication Providers

- Windows Authentication
 - IIS performs authentication with client
 - Users authenticated to Windows account (AD or local)
 - Only type supported in WSS V2 and SPS 2003
- ASP.NET Forms Authentication
 - Based on ASP.NET 2.0 authentication provider FX
 - IIS configured for anonymous access
- Web Single Sign On
 - Based on Federation

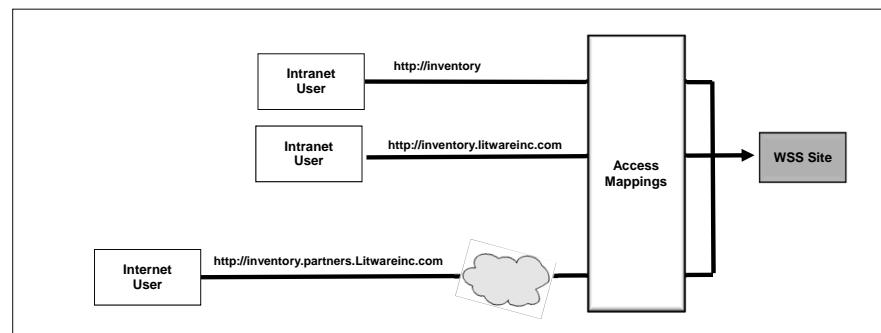
Authentication And WSS Zones

- WSS authentication configured in terms of zones
 - There is one zone per IIS Web site
 - Each zone has its own web.config file
 - Each zone has exactly one authentication provider
 - Web Application can be extended with multiple zones



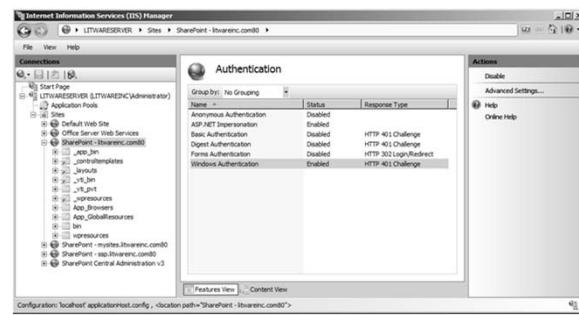
Alternate Access Mapping

- Ensures internal and public URL mappings work correctly
 - Main Web Application URL is mapped by default
 - Web Application and zones can be extended with additional URLs.
 - Alternate URLs can be mapped to one physical path



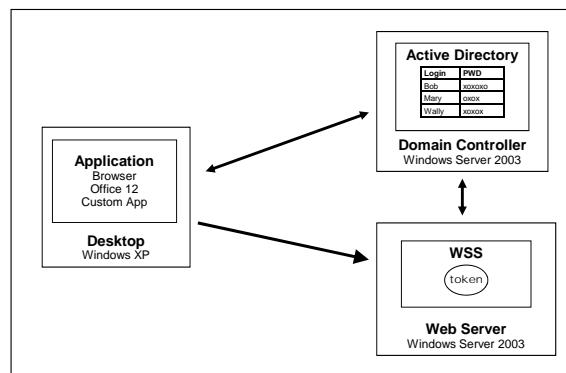
Configuring Windows Authentication

- Authentication performed against Windows accounts
 - Local Accounts can be used in single-server configurations
 - Active Directory accounts are usually much better choice
- Most popular Authentication types
 - Windows Integrated Authentication
 - Basic Authentication



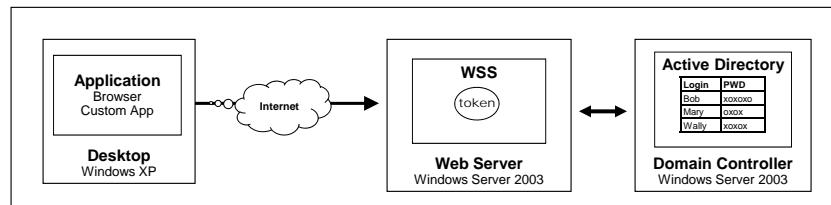
Integrated Windows Authentication

- Authentication using Windows protocols
 - Enhancements to WSS V3 enable Kerberos protocol
 - WSS V3 still uses NTLM protocol when necessary
 - Authentication results in creation of Windows security token



Basic Authentication

- Commonly used in Internet scenarios
 - Industry-standard, cross-browser, firewall-friendly protocol
 - No need for client to access Windows domain controller
 - Authenticates to Windows account and creates security token
 - User name and password passed in clear text
 - You must use HTTPS for any reliable level of security



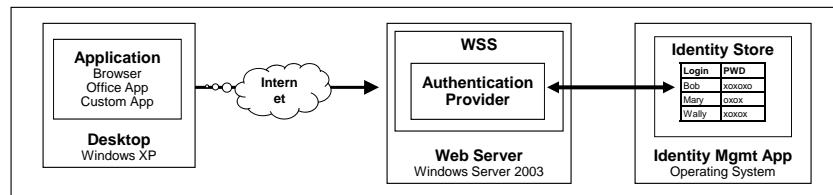
Windows Authentication Zones

- web.config file configures basic ASP.NET settings
 - Authentication specifies resolving to Windows accounts
 - Impersonation is set to true
- WSS is the entity that adds authorization
 - ASP.NET configured to allow all user access to everything

```
<!-- selected snippets from web.config for integrated Windows auth -->
<configuration>
  <system.web>
    <!-- use Integrated Windows Authentication -->
    <authentication mode="Windows" />
    <!-- Impersonate Windows user -->
    <identity impersonate="true" />
    <!-- configure ASP.NET to grant all access to resources -->
    <authorization>
      <allow users="*" />
    </authorization>
  </system.web>
</configuration>
```

Forms-based Authentication (FBA)

- WSS 3.0 supports FBA introduced in ASP.NET 2.0
 - Decouples SharePoint from Active Directory
 - Based on pluggable authentication providers
 - Providers available out-of-box with ASP.NET 2.0
 - Companies can create their own providers as well



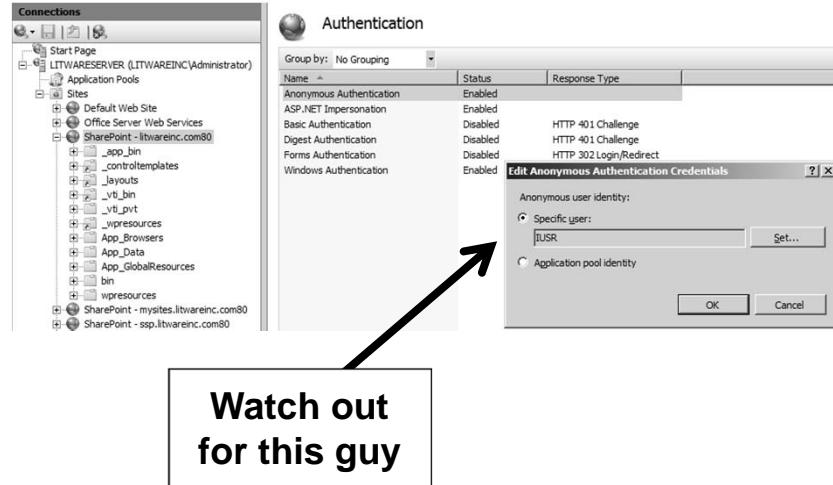
Forms Authentication Zones

- web.config file configures basic ASP.NET settings
 - Authentication configured for Forms
 - Membership provider is configured
 - Impersonation is set to true (e.g. impersonated IUSR_XXX)

```

<!-- selected snippets from web.config for Forms auth -->
<configuration>
  <system.web>
    <!-- use Forms Authentication -->
    <authentication mode="Forms">
      <forms loginUrl="/_layouts/login.aspx" />
    </authentication>
    <!-- configure membership provider -->
    <membership defaultProvider="AspNetSqlMembershipProvider" />
    <!-- impersonate anonymous access user -->
    <identity impersonate="true" />
    <!-- configure ASP.NET to grant all access to resources -->
    <authorization>
      <allow users="*" />
    </authorization>
  </system.web>
</configuration>
  
```

FBA and Windows Identity



Assignment of User Rights

- This is done at several different levels
 - Site Collection
 - Web Application
 - Farm

Site Collection-level Security

- Site Collection Owner
- People and Groups
- Permissions and Permissions Levels
- Securable objects

Permission Levels

- WSS rights managed through permission levels
 - Each permission level consists of a set of rights
 - Permission level defines rights required by business roles
 - Defined on a per site basis
 - Permissions assigned to people and groups

The screenshot shows the 'Permission Levels' page in SharePoint. At the top, it says 'Sales Site > Site Settings > Permission Levels'. Below that, it says 'Permission Levels' and 'This Web site has unique permissions.' There is a button to 'Add a Permission Level' and another to 'Delete Selected Permission Levels'. A table lists the permission levels:

Permission Level	Description
<input checked="" type="checkbox"/> Full Control	Has full control.
<input type="checkbox"/> Design	Can edit lists, document libraries, and pages in the Web site.
<input type="checkbox"/> Contribute	Can view pages and edit list items and documents.
<input type="checkbox"/> Read	Can view pages, list items, and documents.
<input checked="" type="checkbox"/> Limited Access	Can view specific lists, document libraries, list items, folders, or documents when given permissions.
<input type="checkbox"/> My Custom Permission Level	A set of permissions for some specific role of users

Permissions Managed Using Rights

Site Rights	
<input checked="" type="checkbox"/>	Manage Permissions - Create and change permission levels on the Web site and assign permissions to users and groups.
<input checked="" type="checkbox"/>	View Usage Data - View reports on Web site usage.
<input checked="" type="checkbox"/>	Create Subsites - Create subsites such as team sites, Meeting Workspace sites, and Document Workspace sites.
<input checked="" type="checkbox"/>	Manage Web Site - Grants the ability to perform all administration tasks for the Web site as well as manage content and permissions.
<input checked="" type="checkbox"/>	Add and Customize Pages - Add, change, or delete HTML pages or Web Part Pages, and edit the Web site using a Windows SharePoint Services-compatible editor.
<input checked="" type="checkbox"/>	Apply Themes and Borders - Apply a theme or borders to the entire Web site.
<input checked="" type="checkbox"/>	Apply Style Sheets - Apply a style sheet (.CSS file) to the Web site.
<input checked="" type="checkbox"/>	Create Groups - Create a group of users that can be used anywhere within the site collection.
<input checked="" type="checkbox"/>	Browse Directories - Enumerate files and folders in a Web site using FrontPage and Web DAV interfaces.
<input checked="" type="checkbox"/>	Use Self-Service Site Creation - Create a Web site using Self-Service Site Creation.
<input checked="" type="checkbox"/>	View Pages - View pages in a Web site.
<input checked="" type="checkbox"/>	Enumerate Permissions - Enumerate permissions on the Web site, list, folder, document, or list item.
<input checked="" type="checkbox"/>	Browse User Information - View information about users of the Web site.
<input checked="" type="checkbox"/>	Manage Alerts - Manage Alerts for all users of the Web site.
<input checked="" type="checkbox"/>	Use Remote Interfaces - Use SOAP, Web DAV, or FrontPage interfaces to access the web site.
<input checked="" type="checkbox"/>	Open - Allows users to open a web site, list, or folder in order to access items inside that container.
<input checked="" type="checkbox"/>	Edit Own UserInfo - Edit user's own profile

List Rights	
<input checked="" type="checkbox"/>	Manage Lists - Add or remove columns in a list, and add or remove public views of a list.
<input checked="" type="checkbox"/>	Cancel Check-Out - Check in a document without saving the current changes.
<input checked="" type="checkbox"/>	Add Items - Add items to lists, add documents to document libraries, add Web discussion comments.
<input checked="" type="checkbox"/>	Edit Items - Edit items in lists, edit documents in document libraries, edit Web discussion comments in documents, and customize Web Part Pages in document libraries.
<input checked="" type="checkbox"/>	Delete Items - Deletes items from a list, documents from a document library, and Web discussion comments in documents.
<input checked="" type="checkbox"/>	View Items - View items in lists, documents in document libraries, and view Web discussion comments.
<input checked="" type="checkbox"/>	Approve Items - Approve a minor version of a list item or document.
<input checked="" type="checkbox"/>	Open Items - View the source of documents with server-side file handlers.
<input checked="" type="checkbox"/>	View Versions - View past versions of a list item or document.
<input checked="" type="checkbox"/>	Delete Versions - Delete past versions of a list item or document.
<input checked="" type="checkbox"/>	Create Alerts - Create e-mail alerts.
<input checked="" type="checkbox"/>	View Document Pages - View the documents and views in a list or document library.

Personal Rights	
<input checked="" type="checkbox"/>	Manage Personal Views - Create, change, and delete personal views of lists.
<input checked="" type="checkbox"/>	Add/Remove Private Web Parts - Add or remove private Web Parts on a Web Part Page.
<input checked="" type="checkbox"/>	Update Personal Web Parts - Update Web Parts to display personalized information.

Securable Objects

site collection

- top-level site
- list1
 - item1
 - item2
- documentlibrary1
 - document1
 - document2
- childsite1
 - list1
 - item1
 - item2

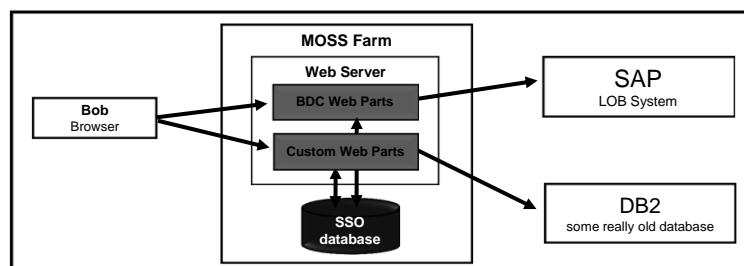
Web Application Security Policy

- New with WSS 3.0
 - Allows farm administrator to grant or deny access
 - Web application policy overrides site collection

The screenshot shows the 'Central Administration > Application Management > Policy for Web Application > Add Users' page. The 'Web Application' dropdown is set to 'http://litwareinc.com/'. The 'Zone' dropdown is set to '(All zones)'. In the 'Choose Users' section, 'Brian Cox' is listed. Under 'Choose Permissions', the 'Full Read - Has full read-only access' checkbox is selected. Other options like 'Full Control', 'Deny Write', and 'Deny All' are available but not checked.

MOSS Single Sign-On Service

- Provides credential mapping
 - Maps identities between identity management systems e.g. map authenticated Windows user to SAP credentials
 - Stores credentials in encrypted form in SSO database
- Where is it used?
 - Custom Web Parts, BDC, Excel Services, etc.



Programming SSO

```
// assumes - using Microsoft.SharePoint.Portal;
// assumes - using Microsoft.SharePoint.Portal.SingleSignon;
protected override void RenderContents(System.Web.UI.HtmlTextWriter writer) {
    try {
        string user = null;
        string password = null;
        string[] credentials = null;
        // get user credentials
        Credentials.GetCredentials(1, AppName, ref credentials);
        user = credentials[0];
        password = credentials[1];
        // do something with user credentials
        writer.Write("User: " + user + "<br/>");
        writer.Write("Password: " + password + "<br/>");

    }
    catch (SingleSignonCredsNotFoundException x) {
        // redirect user to MOSS-supplied page for entering credentials
        writer.Write("<a href=\"" +
                    SingleSignonLocator.GetCredentialEntryUrl(AppName) +
                    "\">>Please Sign In</a>");
    }
    catch (SingleSignonException x) {
        // deal with other SSO-specific type of exception
    }
    catch (Exception x) {
        // deal with other any other type of exception
    }
}
```

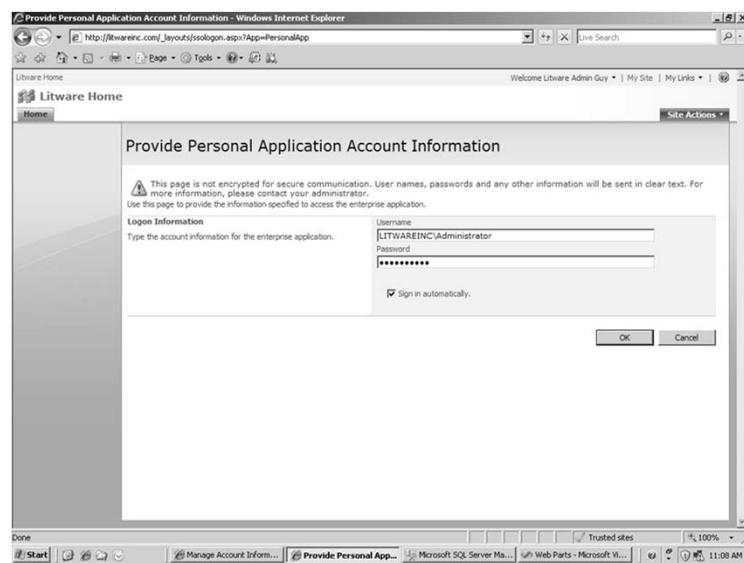
Group SSO Login

User: SSOLogin
Password: pass@word1

Individual SSO Login

Please Sign In

SSO Credential Entry Page



Summary

- Authenticating users in WSS and MOSS
- Configuring access control within a site collection
- Configuring access control for Web Application
- Using the MOSS single sign-on service (SSO)