

# The Great SharePoint Adventure 2007

## Student Presentations Manual

---

This course includes the following presentations:

---

1. Roadmap to SharePoint 2007 Development
2. Developing Features
3. SharePoint Architecture
4. Pages and Design
5. Master Pages and Site Branding
6. Web Part Development
7. AJAX Web Parts
8. Lighting up SharePoint with Silverlight 2
9. Lists and Content Types
10. Document Libraries
11. Creating and Testing Site Definitions
12. Forms Services with InfoPath 2007
13. Developing SharePoint Workflows with Visual Studio
14. Extending Portal and Search in SharePoint Server
15. Web Content Management
16. The Business Data Catalog (BCD)
17. Excel Services and Report Center
18. Application Security



# SharePoint 2007 Developer Roadmap

Getting Started with SharePoint Development



## Logistics

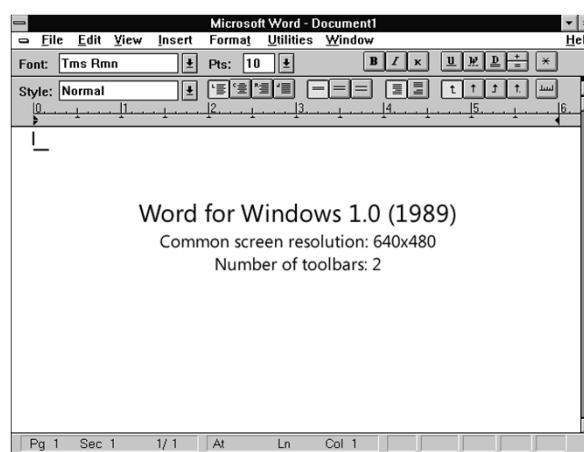
- Basic Human Needs
  - Bathrooms
  - Food and coffee
  - Meals
  - Class hours

## Agenda

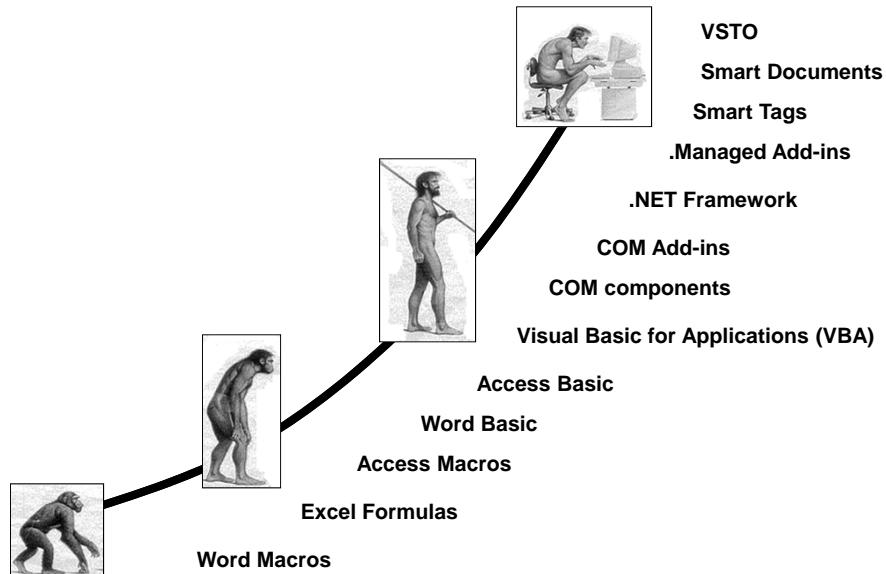
- Architectural overview of SharePoint 2007
  - Windows SharePoint Services 3.0 (WSS)
  - Microsoft Office SharePoint Server 2007 (MOSS)
- Basic WSS Terminology
- WSS as a collaboration solution
- Customizing WSS Sites
- Overview of MOSS components and services

## Microsoft Office Through the Ages

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

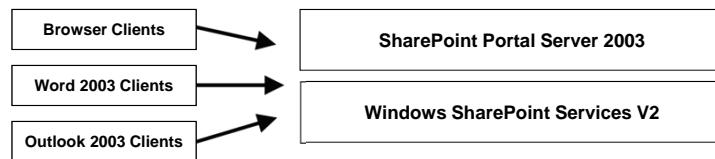


## Evolution of the Office Developer

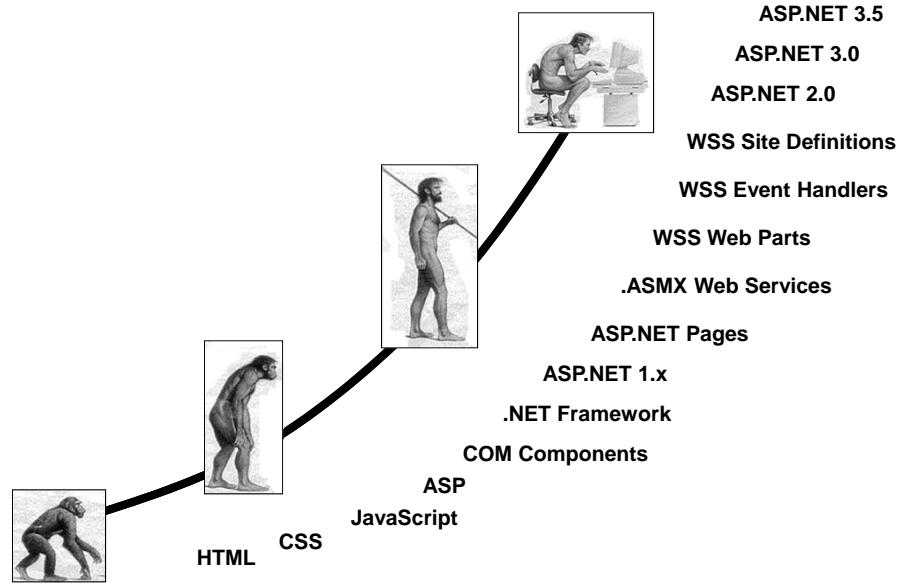


## Office 2003 Server Components

- 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

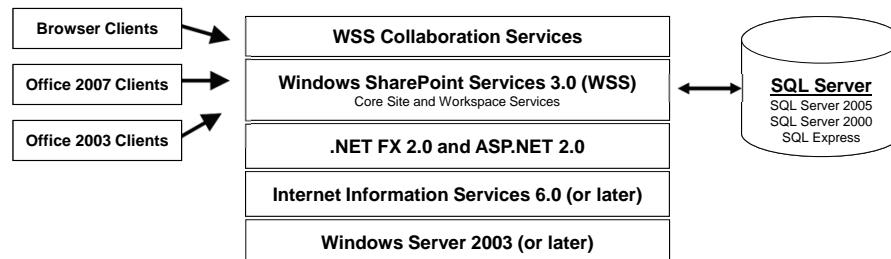
- What's Your Name?
- What Company are you with?
- How have you evolved as a Developer?
- Do you have experience with...
  - The .NET Framework and Visual Studio
  - ASP.NET (what was the latest version)
  - WSS 2.0 and SPS 2003
  - WSS 3.0 and MOSS

## 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

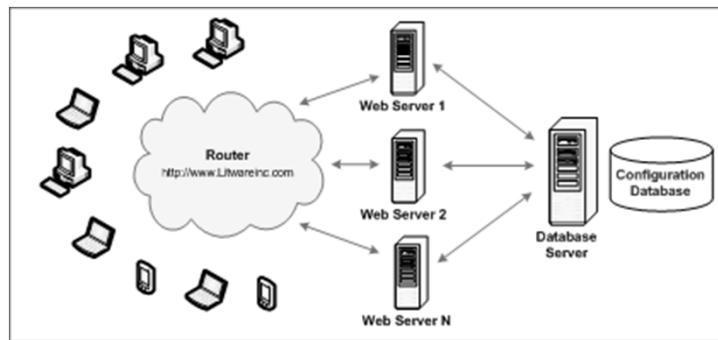
## The WSS 3.0 Server-side Platform

- 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



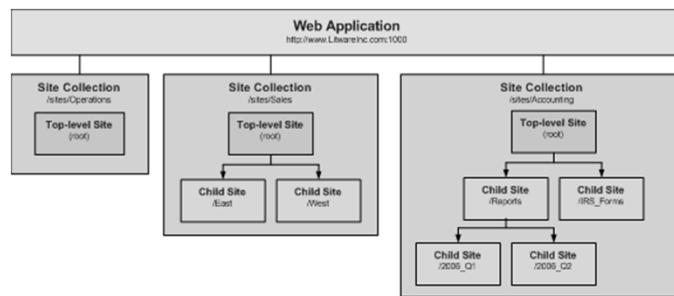
## Web Applications

- 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>]
      [-ownermail <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>]

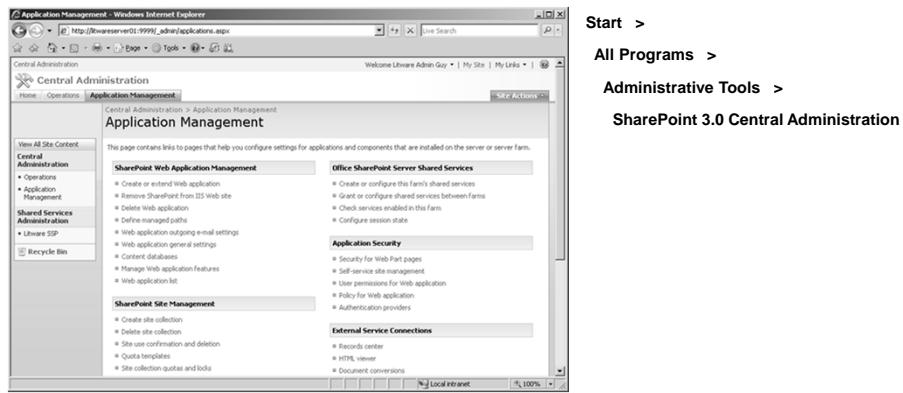
C:\>STSADM.EXE -o CreateSite -url http://LitwareInc.com/sites/Marketing2007
      -ownerlogin LITWAREINC\Administrator -ownermail administrator@litwareinc.com
      -sitetemplate STS#1
Operation completed successfully.

C:\>
  
```

A screenshot of a Windows Command Prompt window showing the execution of the STSADM.EXE command. The command 'stsadm.exe -o CreateSite' is run, followed by parameters specifying the URL, owner information, site template, and quota template. The output shows the operation completed successfully.

## 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



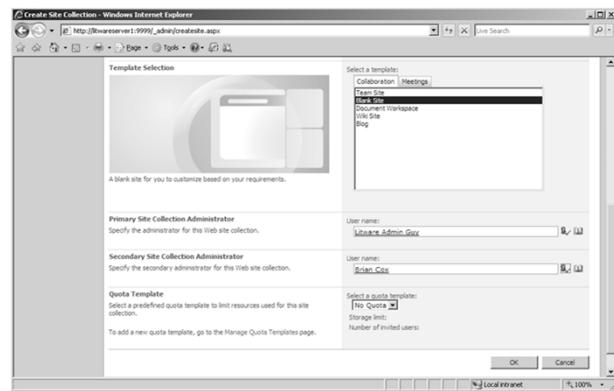
## Creating New Site Collections

- 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

The screenshot shows the 'Create Site Collection' dialog box. It has fields for 'Web Application' (set to 'http://itwareloc.com/'), 'Title' ('itware Home'), 'Description' ('A site for tracking itware sales information'), and 'URL' ('http://itwareloc.com/sites/Sales'). There are 'OK' and 'Cancel' buttons at the top right. The background shows the SharePoint Central Administration 'Application Management' page.

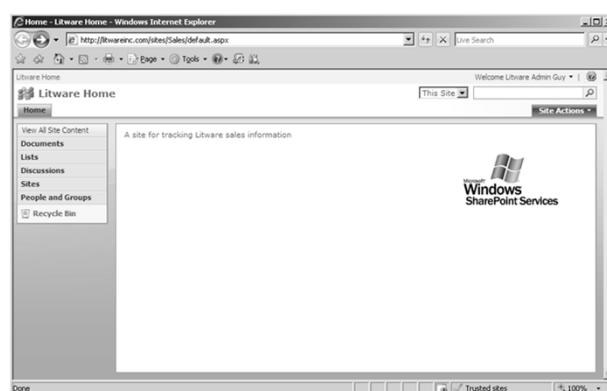
## Creating New Site Collections (Part 2)

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



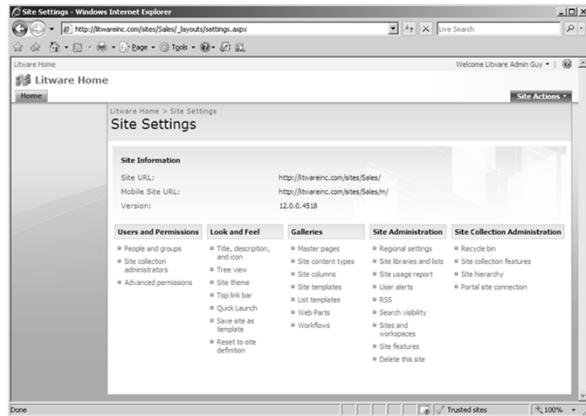
## A New WSS Site

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



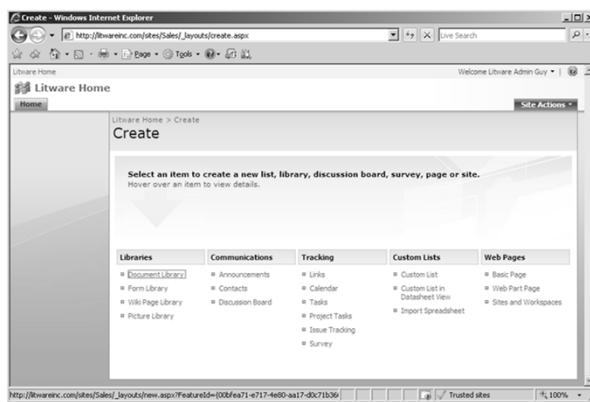
## The Site Settings Page

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



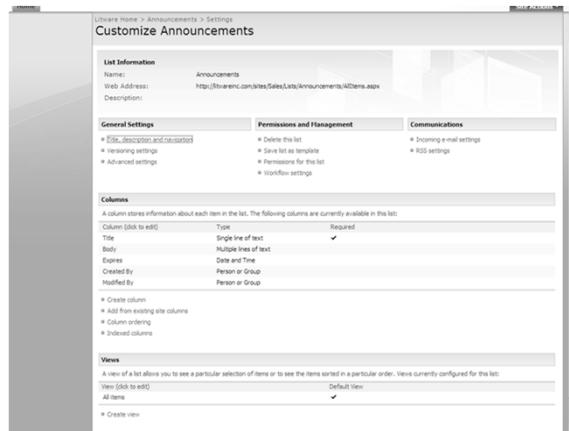
## The Create Page

- 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



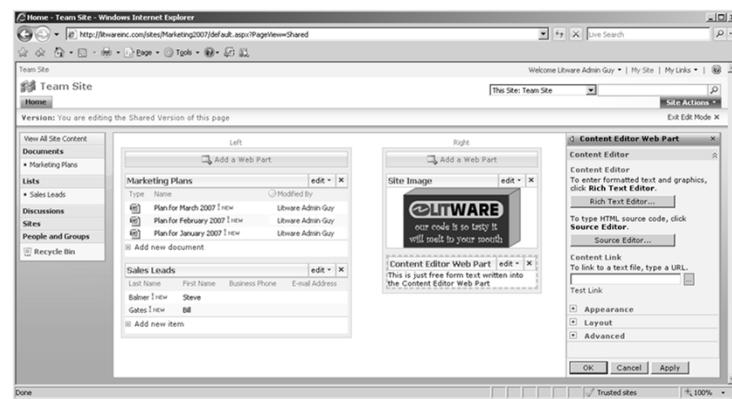
## The List Settings Page

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

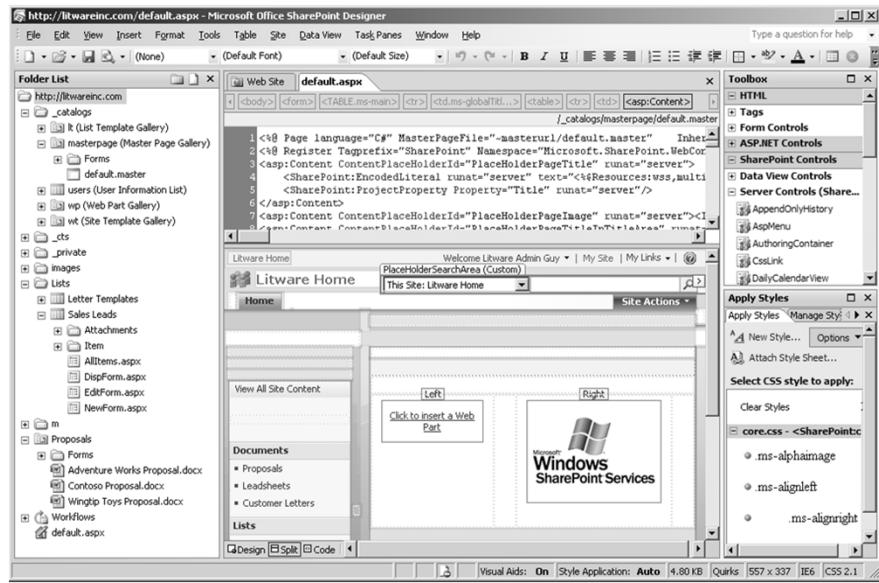


## Page Customization using Web Parts

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

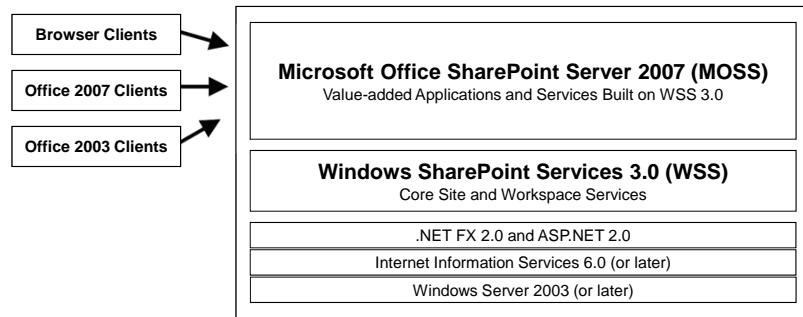


## Customization with the SharePoint Designer



## Microsoft Office SharePoint Server 2007

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

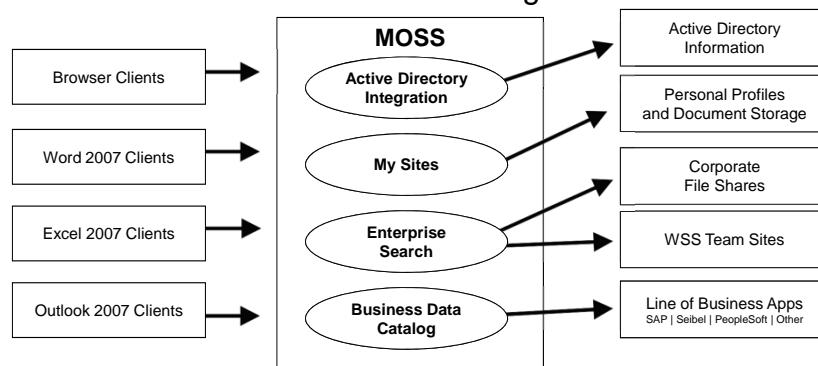


## MOSS Services and Components

- 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

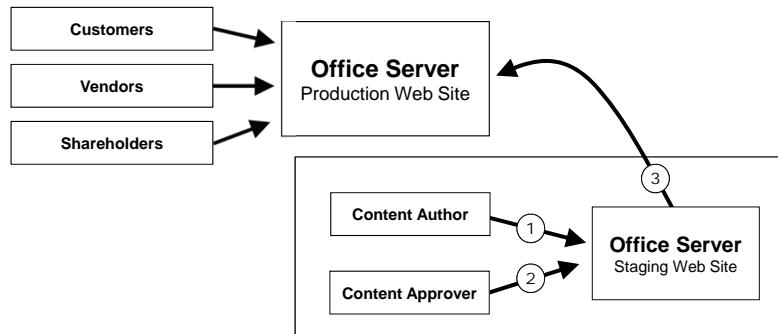
## Portal and Search

- 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

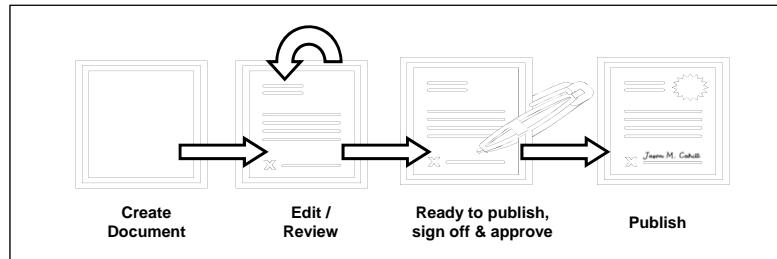


## InfoPath 2007 and Forms Services

- 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

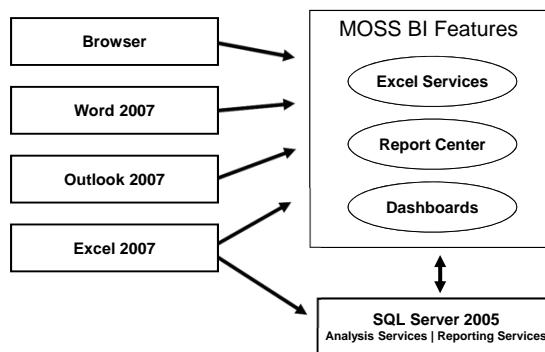
## SharePoint 2007 Workflows

- 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

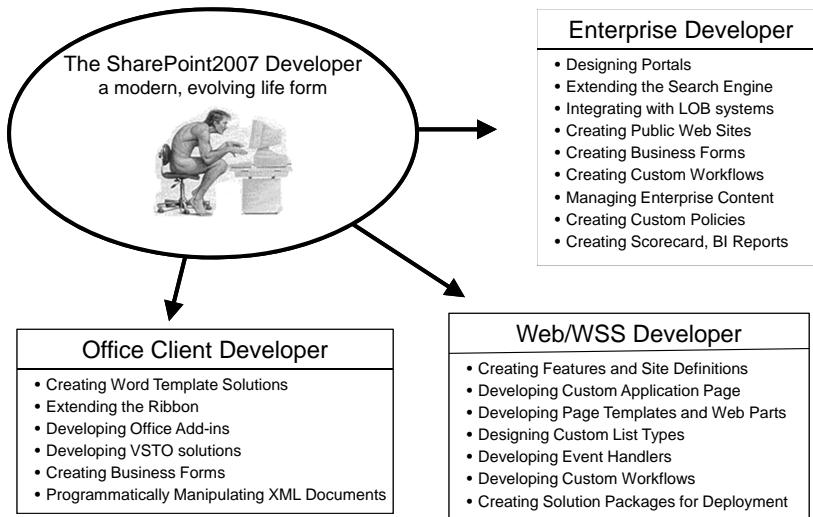


## Business Intelligence

- 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



## Developing Features for WSS

Creating SharePoint Components with Visual Studio



### Agenda

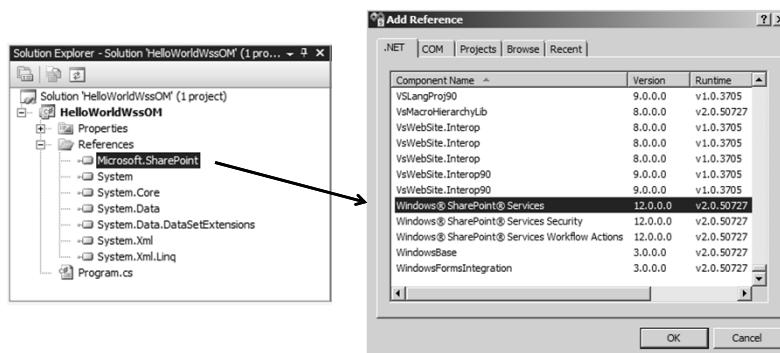
- SharePoint Customization versus Development
- The WSS system directories
- What Are Features?
- Developing a Custom Feature
- Adding Event Handlers to a Feature

## Customization Versus Development

- Site Customizations
  - Changes to one particular site
  - Done using the browser or the SharePoint Designer
  - Changes recorded in content database
  - Easy to do but hard to reuse
- WSS Development
  - Creation of reusable templates/components
  - Templates/components installed on Web server
  - Development based on Visual Studio projects
  - Project source files checked into source code control
  - Projects can be moved through staging to production

## Hello World: The WSS Object Model

- Create a simple Console Application
  - Add a reference to Microsoft.SharePoint.dll
  - Write the code to access a site and see it's lists



## Watch Out: Inconsistent Terminology

New Term	Old Term	WSS Object Model
Site Collection	Site	SPSite
Site	Web	SPWeb

## The 'Hello World' Code

```
using System;
using Microsoft.SharePoint;
namespace Hello_WSS_OM {
    class Program {
        static void Main(string[] args) {

            string sitePath = "http://lithwareinc.com";
            // enter object model through site collection.
            SPSite siteCollection = new SPSite(sitePath);
            // obtain reference to top-level site.
            SPWeb site = siteCollection.RootWeb;
            // enumerate through lists of site
            foreach (SPList list in site.Lists) {
                Console.WriteLine(list.Title);
            }
            // clean up by calling Dispose.
            site.Dispose();
            siteCollection.Dispose();
        }
    }
}
```

## What is a Feature?

- A building block for creating SharePoint solutions
  - A unit of design, implementation and deployment
- Features can contain elements
  - e.g. menu items, links, list types and list instances
  - Many other element types possible
- Features can contain event handlers
  - You can add any code which used WSS object model

## User's View of Features

- Features support concept of activation/deactivation

The screenshot shows the 'Site Features' page in a Windows Internet Explorer browser. The URL is http://lwwarenc.com/\_layouts/ManageFeatures.aspx. The page lists several SharePoint features with their descriptions and status buttons ('Activate' or 'Deactivate').

Name	Status
Office SharePoint Server Enterprise Site features Features such as the business data catalog, forms services, and Excel Services, included in the Office SharePoint Server Enterprise License	Activate
Office SharePoint Server Publishing Create a Web page library as well as supporting libraries to create and publish pages based on page layouts.	Activate
Office SharePoint Server Standard Site features Features such as user profiles and search, included in the Office SharePoint Server Standard License	Activate
Team Collaboration Lists Provides team collaboration capabilities for a site by making standard lists, such as document libraries and issues, available.	Deactivate
Translation Management Library Create a translation management library when you want to create documents in multiple languages and manage translation tasks. Translation management libraries include a workflow to manage the translation process and provide sub-folders, file versioning, and check-in/check-out.	Active

This is the site-level feature management page in a WSS farm where MOSS has been installed.

Much of the functionality of MOSS is enabled and disable by activating and deactivating features that have been developed by the MOSS team.

# The WSS System Directories

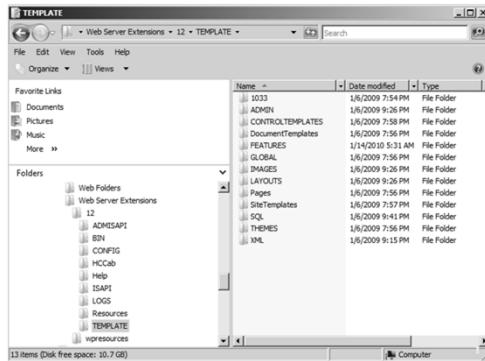
- Developers must learn WSS system directories

\12\TEMPLATE

\12\TEMPLATE\FEATURES ← This is the one we care about in this lecture

\12\TEMPLATE\IMAGES

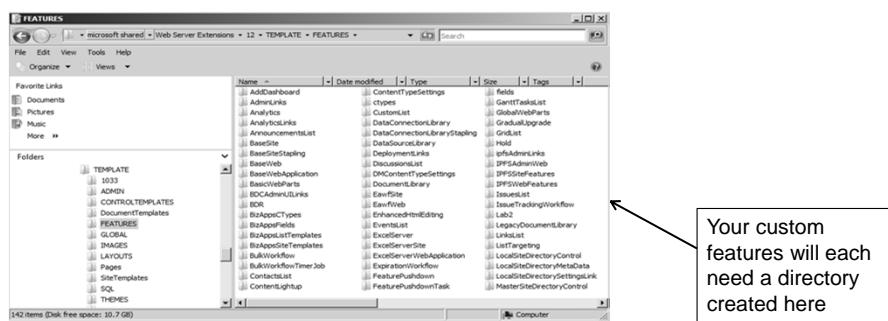
\12\TEMPLATE\LAYOUTS



## The Features Directory

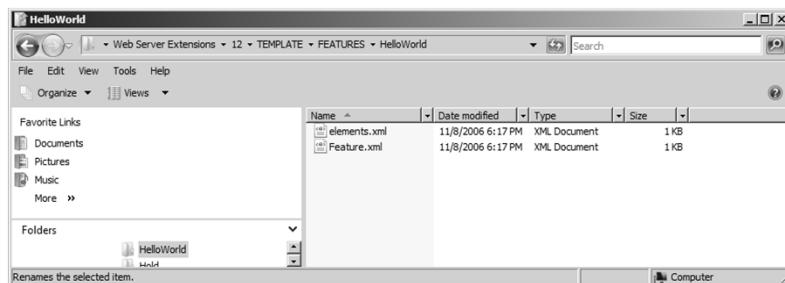
- Functionality in WSS based on Features

- Features are installed at farm level
  - Feature activation makes functionality available
  - WSS supports for different feature activation scopes
    - (1) Site
    - (2) Site Collection
    - (3) Web Application
    - (4) Farm



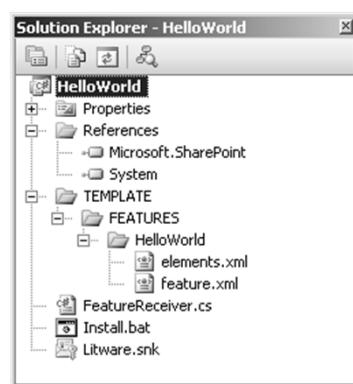
## Developer's View of a Feature

- Each feature requires its own directory
  - Directory must contain feature.xml file
  - Directory often contains other files definition elements



## Creating the 'Hello World' Feature

- Create a new Visual Studio Class Library project
  - Create XML files which define feature
  - Add a FeatureActivated event handler



## The feature.xml file

- Feature.xml file serves as feature manifest
  - Features defined in declarative fashion using CAML

CAML = Collaborative Application Markup Language

```
<Feature
  Id="B2CB42E2-4F0A-4380-AABA-1EF9CD526F20"
  Title="A Sample Feature: Hello World"
  Description="Hi mom, class is fun. I am doing great"
  Scope="Web"
  Hidden="FALSE"
  ImageUrl="TGP\HelloWorldFeature.gif"
  xmlns="http://schemas.microsoft.com/sharepoint/"></Feature>
```

## Elements.xml

- Feature includes elements defined using CAML
  - This element defines a Site Actions menu item
  - There are many other types of elements

```
<Elements xmlns="http://schemas.microsoft.com/sharepoint/"><CustomAction
  Id="SiteActionsToolBar"
  GroupId="SiteActions"
  Location="Microsoft.SharePoint.StandardMenu"
  Sequence="100"
  Title="Hello World"
  Description="A custom menu item added using a feature"
  ImageUrl="_layouts/images/crtsite.gif">
  <Url Action="Url="http://msdn.microsoft.com"/>
</CustomAction></Elements>
```

## Install.bat

- Visual Studio supports post-build events
  - Can be used to run batch file to deploy components
  - Used on development machines
  - Should not be used on staging/production machines

```

@SET TEMPLATEDIR="c:\program files\microsoft shared\web server extensions\12\template"
@SET STSADM="c:\program files\microsoft shared\web server extensions\12\bin\stsadm"
@SET GACUTIL="c:\Program Files\Microsoft SDKs\Windows\v6.0A\bin\gacutil.exe"

Echo Installing HelloWorld.dll in GAC
%GACUTIL% -if bin\debug\HelloWorld.dll

Echo Copying files to TEMPLATE directory
xcopy /e /y TEMPLATE/* %TEMPLATEDIR%

Echo Installing feature
%STSADM% -o installfeature -filename HelloWorld\feature.xml -force

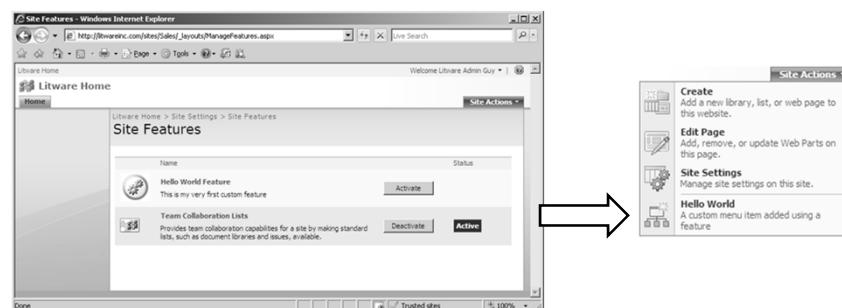
Echo Restart all IIS worker processes
IISRESET

Echo Restart just the IIS worker process for a particular Application Pool
REM cscript c:\windows\system32\iisapp.vbs /a "SharePointDefaultAppPool" /r

```

## Feature Activation

- Steps to testing Feature
  - Copy Feature files to FEATURES directory
  - Install feature with WSS
  - Activate Feature within a specific site



## Feature event handlers

- FeatureReceiver class provides 4 event handlers

```
using System;
using Microsoft.SharePoint;
namespace HelloWorld {
    public class FeatureReceiver : SPFeatureReceiver {
        // no functionality required for install/uninstall events
        public override void FeatureInstalled(SPFeatureReceiverProperties properties) { }
        public override void FeatureUninstalling(SPFeatureReceiverProperties properties) { }

        public override void FeatureActivated(SPFeatureReceiverProperties properties) {
            SPWeb site = (SPWeb)properties.Feature.Parent;
            // track original site Title using SPWeb property bag
            site.Properties["OriginalTitle"] = site.Title;
            site.Properties.Update();
            // update site title
            site.Title = "Hello World";
            site.Update();
        }

        public override void FeatureDeactivating(SPFeatureReceiverProperties properties) {
            // reset site Title back to its original value
            SPWeb site = (SPWeb)properties.Feature.Parent;
            site.Title = site.Properties["OriginalTitle"];
            site.Update();
        }
    }
}
```

## The feature.xml file revisited

- Feature.xml file serves as feature manifest

```
<Feature
  Id="B2CB42E2-4F0A-4380-AABA-1EF9CD526F20"
  Title="A Sample Feature: Hello World"
  Description="Hi mom, class is fun. I am doing great"
  Scope="Web"
  Hidden="FALSE"
  ImageUrl="TPG\WhitePlaceholder.gif"
  ReceiverAssembly="HelloWorld, [full 4-part assembly name]"
  ReceiverClass="HelloWorld.FeatureReceiver"
  xmlns="http://schemas.microsoft.com/sharepoint/">>

  <ElementManifests>
    <ElementManifest Location="elements.xml" />
  </ElementManifests>

</Feature>
```

## Agenda

- SharePoint Customization versus Development
- The WSS system directories
- What Are Features?
- Developing a Custom Feature
- Adding Event Handlers to a Feature



## SharePoint Architecture

Taking a Look Under the Hood

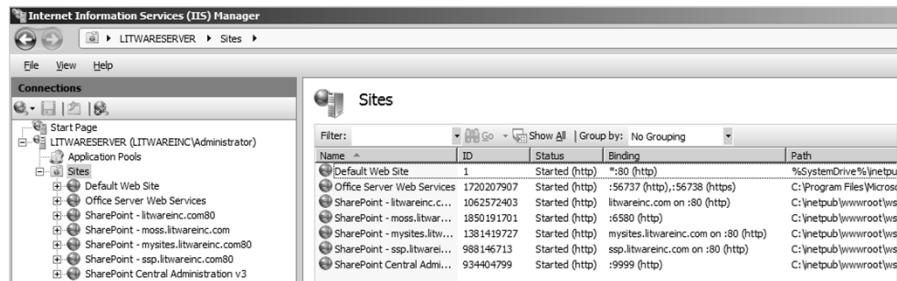


### Agenda

- 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

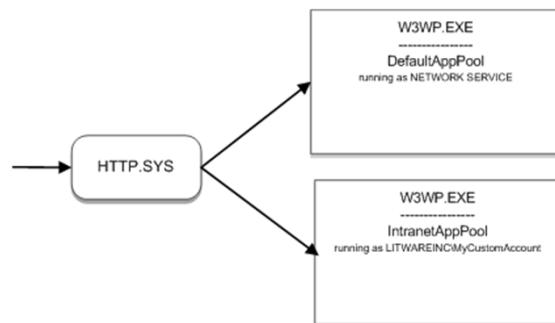
## IIS Web Sites

- 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

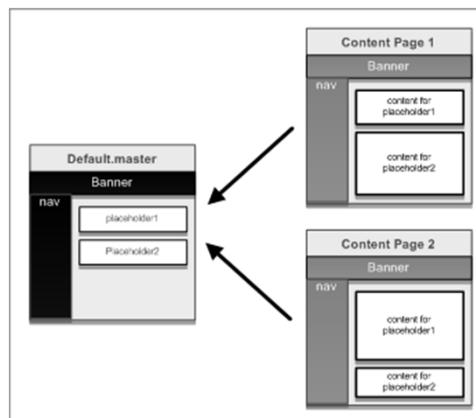
- ASP.NET development typically based on pages
  - Pages are deployed as .ASPX files to Web server
  - .ASPX files parsed and compiled on first request
  - Compiled page class inherits from `System.Web.UI.Page`

```
<%@ 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

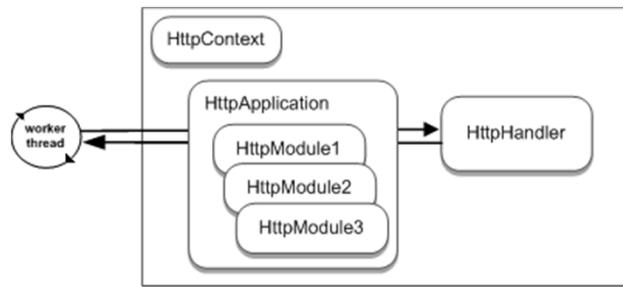
```
<!-- default.master -->
<%@ Master %>
<html><body>
  <form id="frmMain" runat="server">
    <table width="100%">
      <tr><td> <h1>Listware Inc. </h1><hr /></td></tr>
      <tr>
        <td> <!-- Display Main Body of Page -->
          <asp:ContentPlaceholder ID="PlaceholderMain" runat="server"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

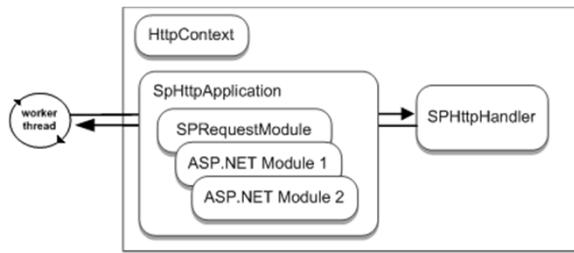
```
<!-- global.asax file at root of WSS Web Application -->
<@Application Inheritance="Microsoft.SharePoint.ApplicationRuntime.SPHttpApplication" >
```

- **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

```

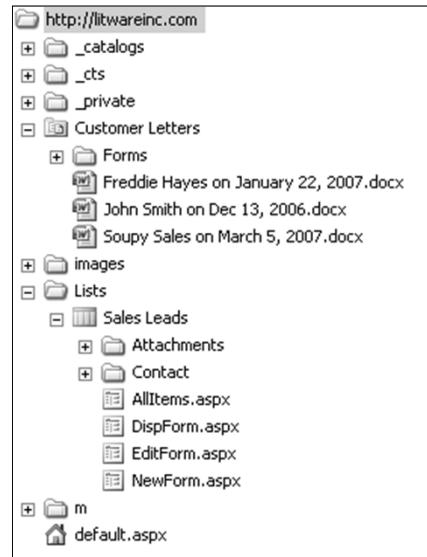
<configuration>
  <configSections>...
  <sharePoint>
    <safeMode MaxControls="200" callstack="false" DirectFileDependencies="10"
      <PageParserPaths>...
      </PageParserPaths>
    </safeMode>
    <webPartLimits MaxZoneParts="50" PropertySize="1048576" />
    <webPartCache Storage="CacheObject" />
    <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" />
    
```

set to true

set to Off

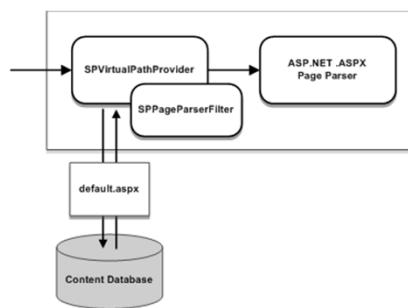
## The Virtual File System of a Site

- 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

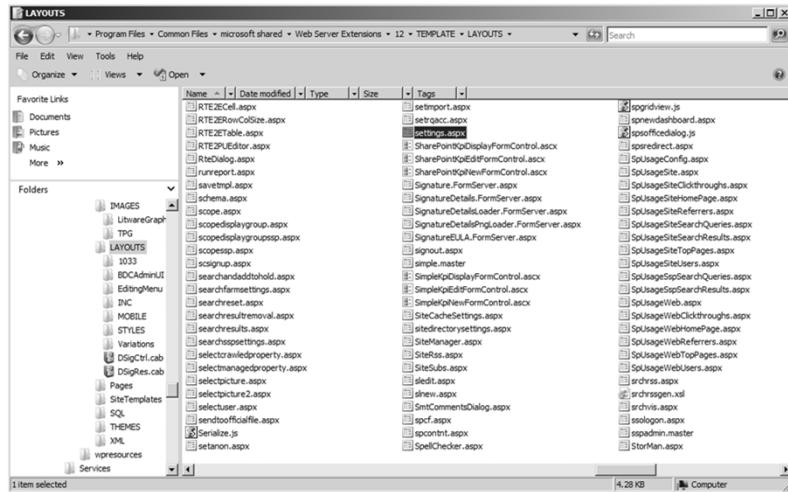


## The \_layouts Virtual Directory

- 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

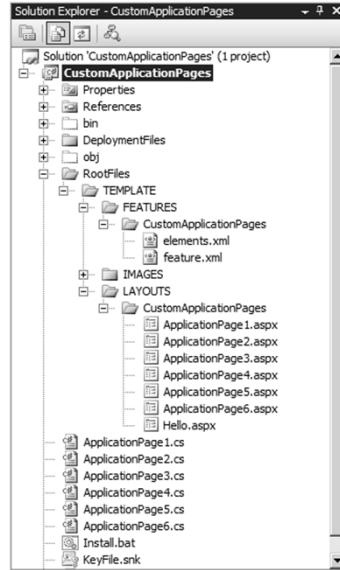
- Steps to creating a custom Application Page
  - Inherit from LayoutsPageBase
  - Link to application.master
  - Add server-side controls and code
  - Deploy to LAYOUTS directory



## '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>
```

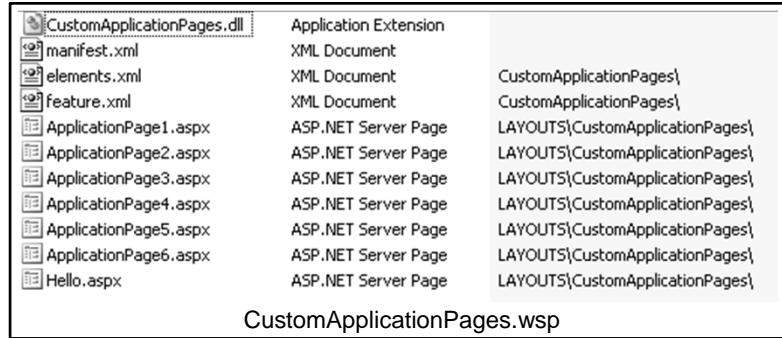


## Deployment using Solution Packages

- 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>

  <TemplateFiles>
    <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"/>
  </TemplateFiles>

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

## Solution Package: install vs. deploy

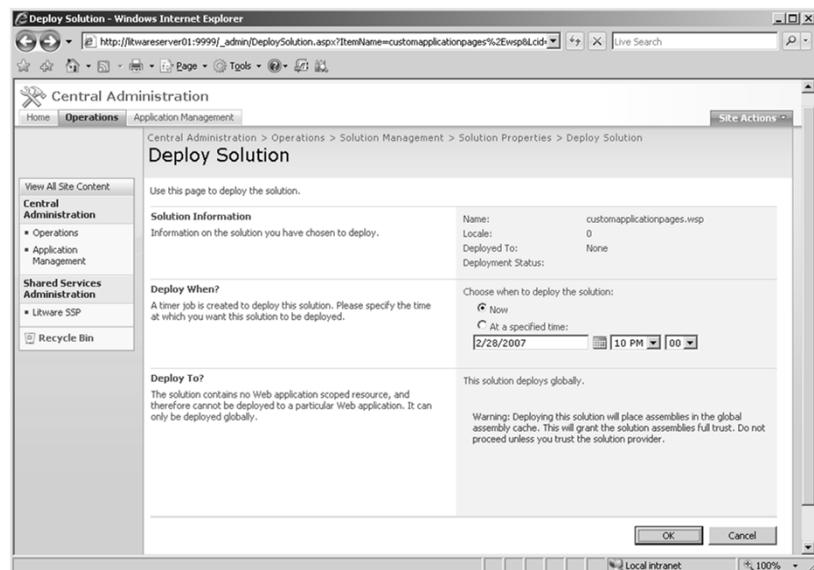
- Solution Package Installation
  - WSP file copied into configuration database
  - Done using **addsolution** operation of STSADM.EXE
- Solution Package Deployment
  - WSP files copied to each FE Web Server and deployed
  - Done using **deploysolution** operation of STSADM.EXE

```
REM – a batch file named DeploySolutionPackage.cmd from CustomApplication 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

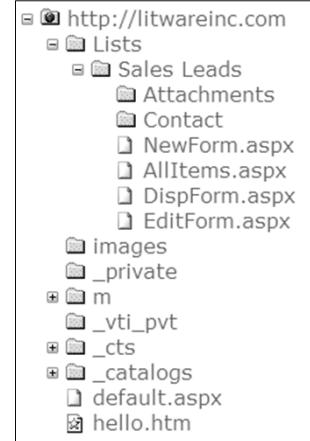


### Agenda

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

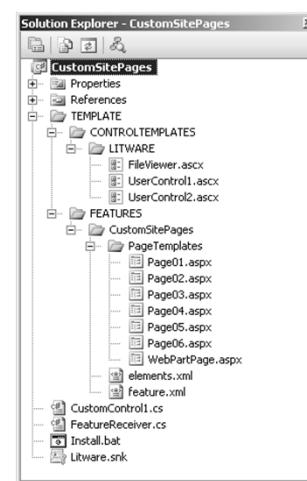
## Site Page Fundamentals

- 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

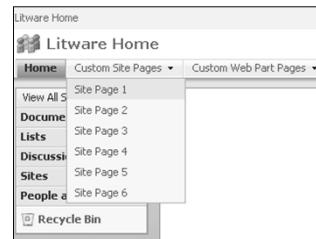
- Module element used to provision page instance
  - File element per page instance
  - Supports page ghosting



```
<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

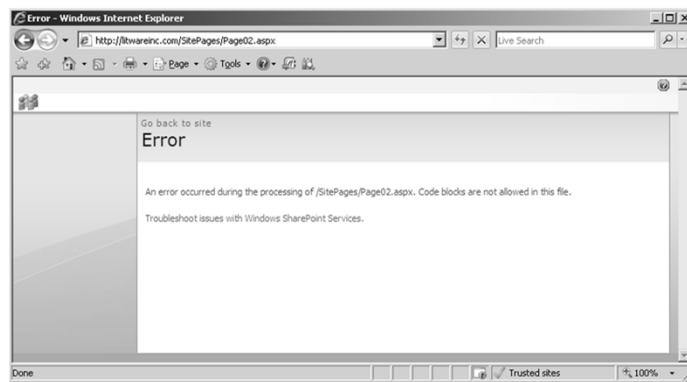
- Navigation nodes can be added
  - Can be added during feature activation
  - Can be added to top-link bar
  - Can be added to QuickLaunch
  - Nodes created as SPNavigationNode



```
public class FeatureReceiver : SPPeerFeatureReceiver {  
    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"));  
    }  
}
```

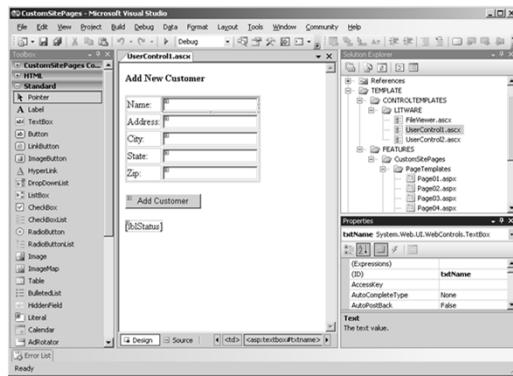
## Safe Mode Processing

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



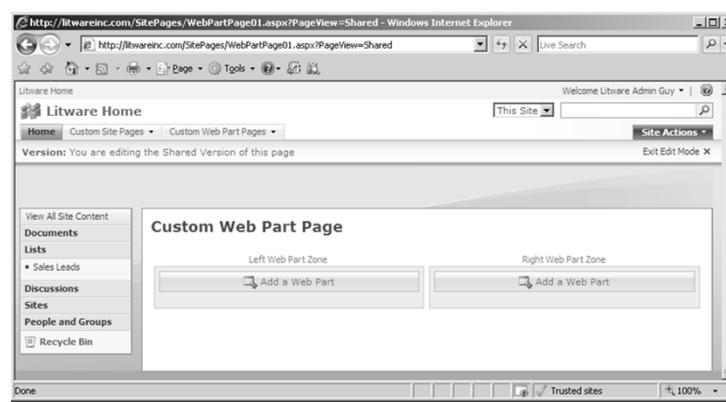
## Designing Pages with Controls

- 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



**Critical  
Path  
TRAINING**

# Master Pages and Site Branding

Working with Master Pages and CSS

## Agenda

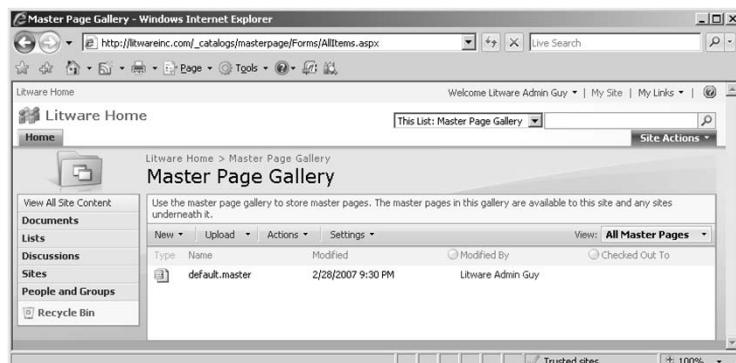
- Master Pages
- Branding a site collection with a custom feature
- Understanding and extending core.css

## Master Pages in WSS

- Application pages use application.master
  - Farm-wide master page
  - Cannot be customized on a per-site basis
- Site Pages use default.master by default
  - default.master is a page template
  - default.master instance is created in Master Page Gallery
  - default.master can be customized on a per-site basis
  - default.master can be replaced with a different template

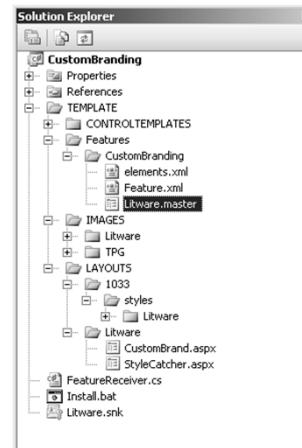
## The Master Page Gallery

- Each site has a Master Page Gallery
  - Instance of default.master automatically provisioned
  - default.master can be customized on a per-site basis



## Demo: CustomBranding

- Important Concepts
  - Custom Master Page Templates
  - Custom CSS File
  - Custom Site Logo



## Custom Master Page Templates

- Creating a Master Page Template
  - Use default.master as a starting point
  - Make changes to suit your tastes
- Master Page templates are like site page templates
  - Support ghosting and unghosting
  - Provisioned using a File element within a Module

```
<Elements ns="http://schemas.microsoft.com/sharepoint/">
  <Module Name="MasterPages" List="116" Url="_catalogs/masterpage">
    <File Url="Litware.master" Type="GhostableInLibrary" />
  </Module>
</Elements>
```

## Master Page Elements

```
<%@Master Language="C#"%
<%@ Register Tagprefix="SharePoint"
    Namespace="Microsoft.SharePoint.WebControls"
    Assembly="Microsoft.SharePoint, ... "%>

<HTML Id="HTML1" runat="server">
<HEAD Id="HEAD1" runat="server">

    <!-- SharePoint Utility Controls -->
    <SharePoint:CssLink ID="CssLink1" runat="server"/>
    <SharePoint:Theme ID="Theme1" runat="server"/>

    <!-- Named Placeholder -->
    <Title ID="onetidTitle">
        <asp:ContentPlaceholder ID=PlaceHolderPageTitle runat="server"/>
    </Title>
    <asp:ContentPlaceholder ID="PlaceHolderAdditionalPageHead" runat="server"/>

    <!-- Delegate Control -->
    <SharePoint:DelegateControl
        ID="DelegateControl1" runat="server"
        ControlID="AdditionalPageHead" AllowMultipleControls="true"/>

</HEAD>
```

## Updating the MasterUrl Property

- Update MasterUrl to redirect site pages
  - Child site can reference the Master Page in a top-level site

```
public partial class _Default : System.Web.UI.Page {
    protected void cmdApplyCustomBrand_Click(object sender, EventArgs e) {
        SPWeb site = SPContext.Current.Site.RootWeb
        string MasterUrlPath = site.ServerRelativeUrl;
        if (!MasterUrlPath.EndsWith("/"))
            MasterUrlPath += "/";
        MasterUrlPath += @"_catalogs/masterpage/Litware.master";
        ApplyCustomBrand(MasterUrlPath, site);
    }

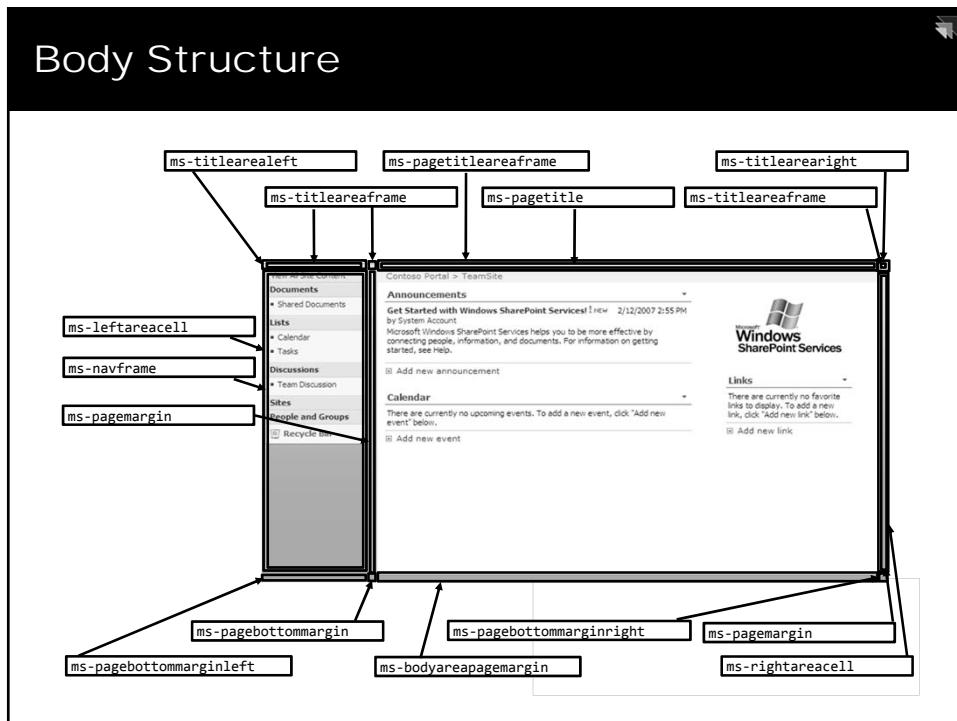
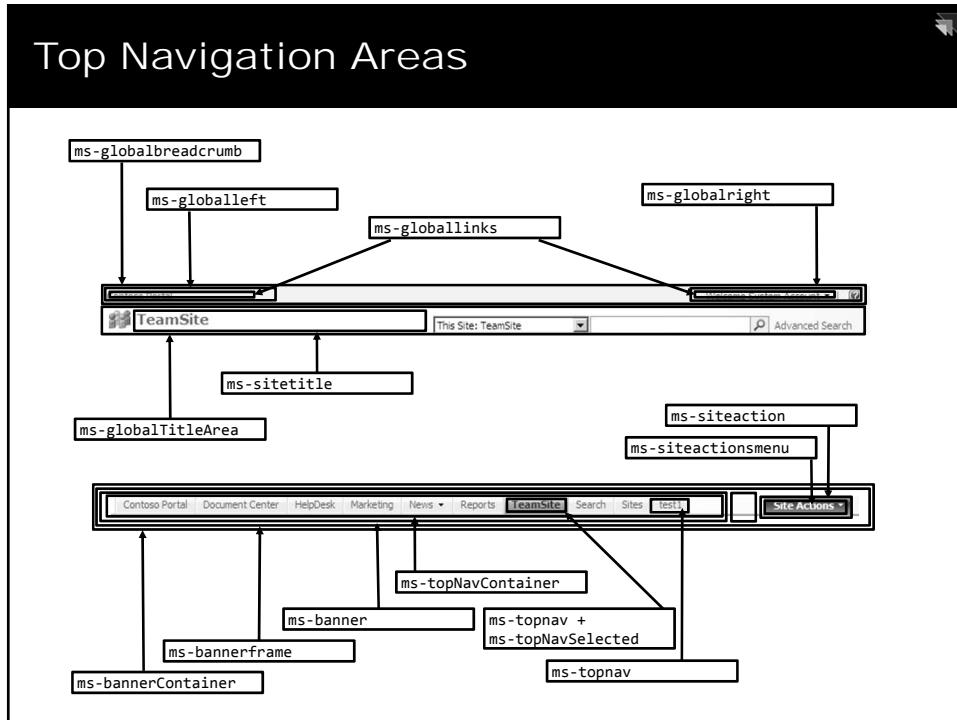
    protected void ApplyCustomBrand(string MasterUrlPath, SPWeb site) {
        site.MasterUrl = MasterUrlPath;
        site.Update();
        // use recursion to update all child sites in site collection
        foreach (SPWeb childInSite in site.Webs) {
            ApplyCustomBrand(MasterUrlPath, child);
        }
    }
}
```

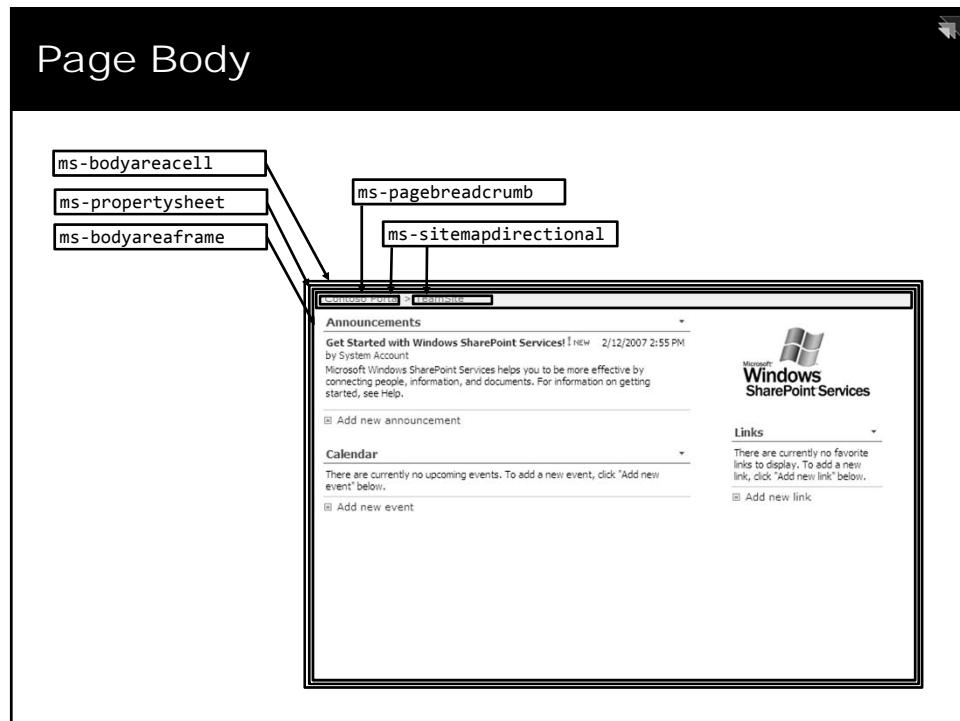
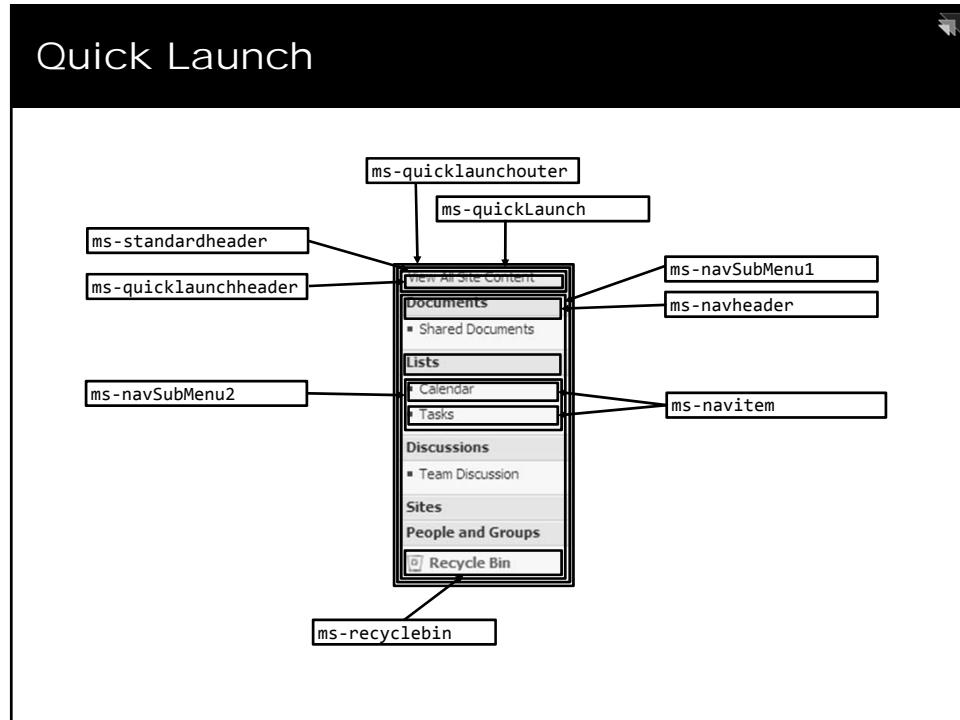
## Understanding core.css

- All styles in WSS initially defined by core.css
  - Located in \TEMPLATE\AYOUTS\1033\STYLES
  - Contains over 4000 lines of CSS class definitions
  - Classes used throughout standard WSS UI elements
- Extending core.css
  - Applying WSS styles (meant for end users)
  - Applying custom CSS files (meant for developers)

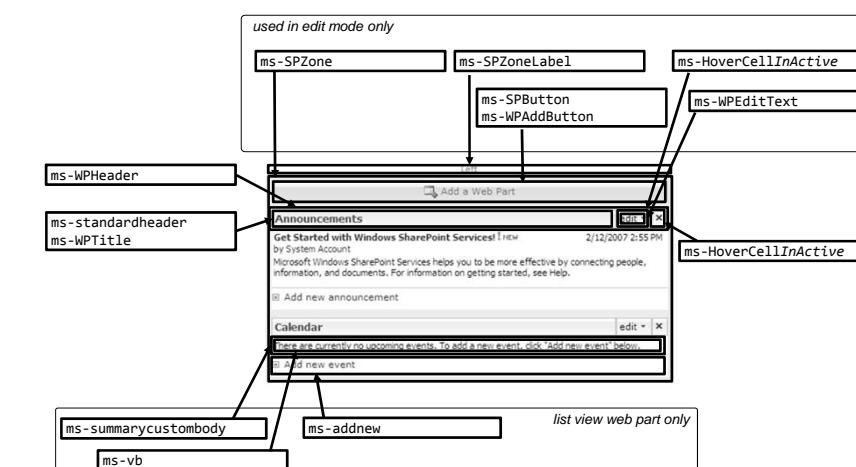
## Applying Custom Branding

```
protected void cmdApplyCustomBrand_Click(object sender, EventArgs e) {  
    SPWeb site = SPContext.Current.Web;  
  
    string MasterUrlPath = site.ServerRelativeUrl;  
    if (!MasterUrlPath.EndsWith("/"))  
        MasterUrlPath += "/";  
    MasterUrlPath += "_catalogs/masterpage/Litware.master";  
    ApplyCustomBrand(MasterUrlPath, site);  
  
    Response.Redirect(Request.RawUrl);  
}  
  
protected void ApplyCustomBrand(string MasterUrlPath, SPWeb site) {  
    site.ApplyTheme("");  
    site.MasterUrl = MasterUrlPath;  
    site.AlternateCssUrl = "/_layouts/1033/STYLES/Litware/LitwareBrand.css";  
    site.SiteLogoUrl = "/_layouts/images/Litware/LitwareFullLogo.png";  
    site.Update();  
  
    foreach (SPWeb child in site.Webs) {  
        ApplyCustomBrand(MasterUrlPath, child);  
    }  
}
```

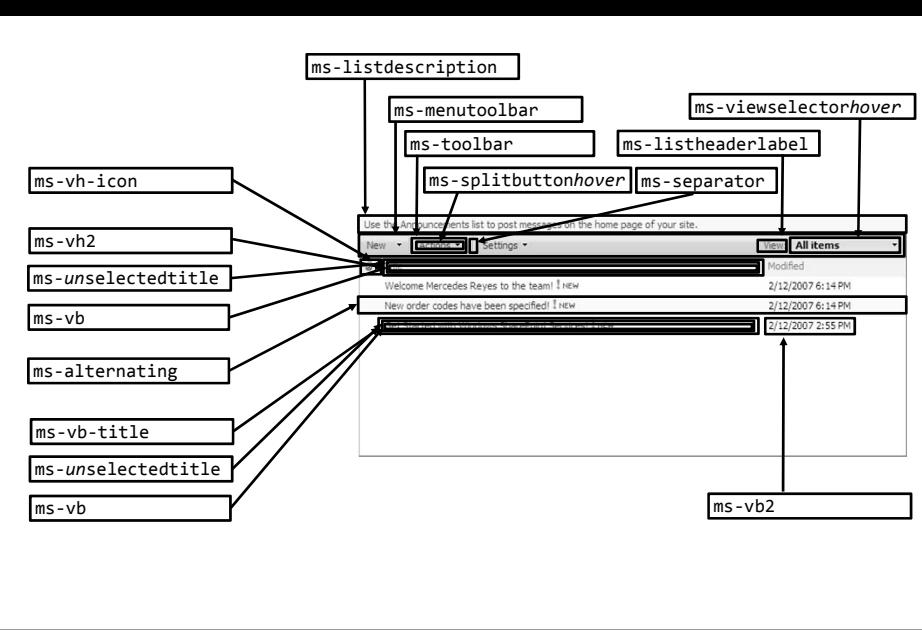


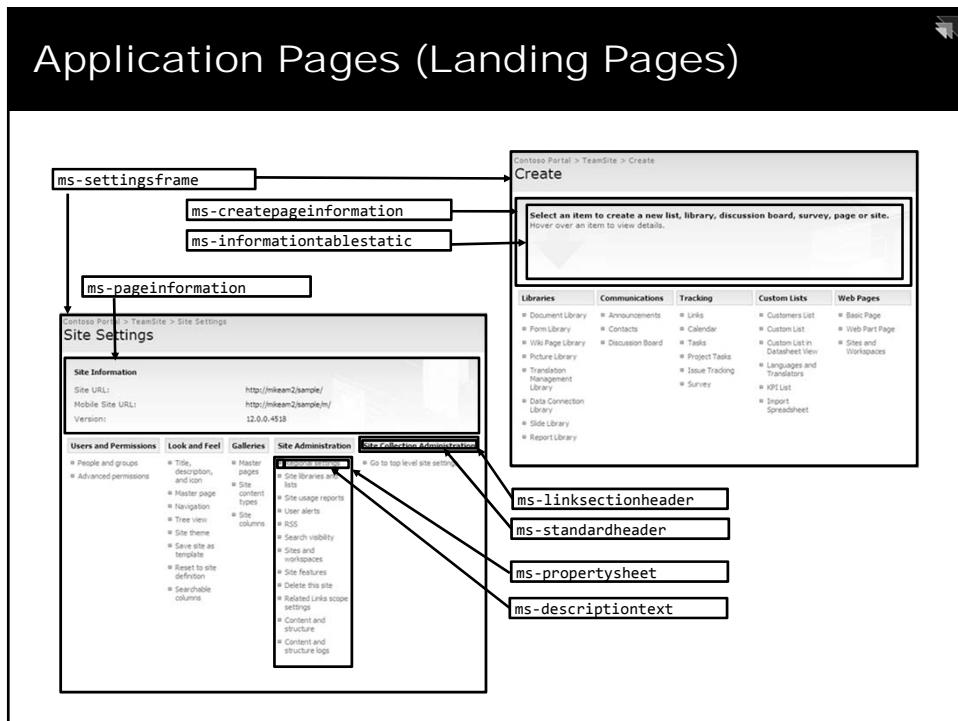
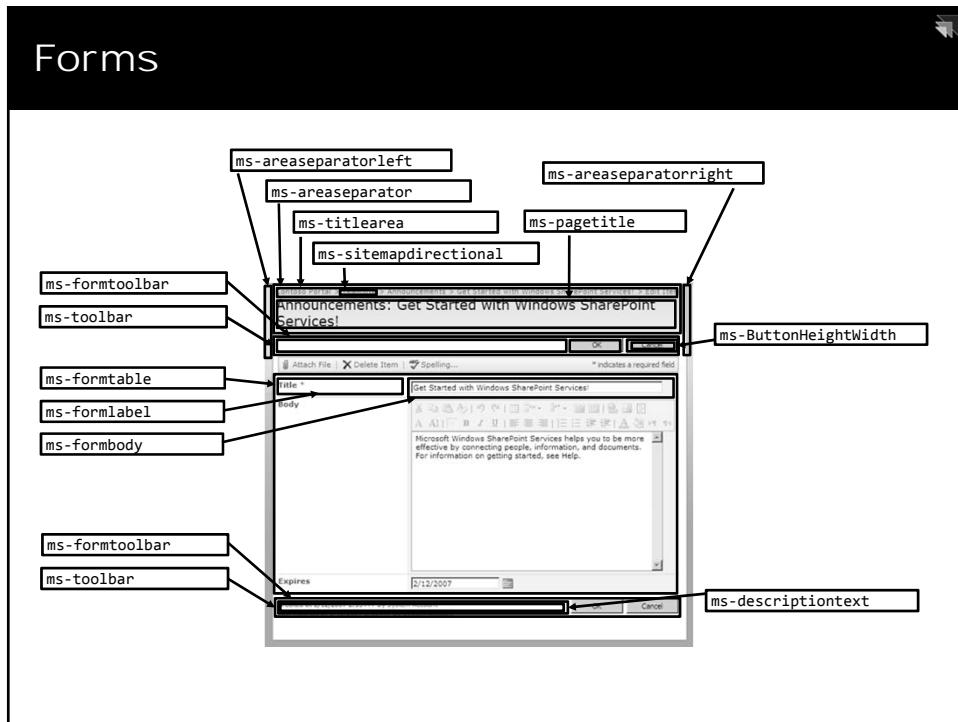


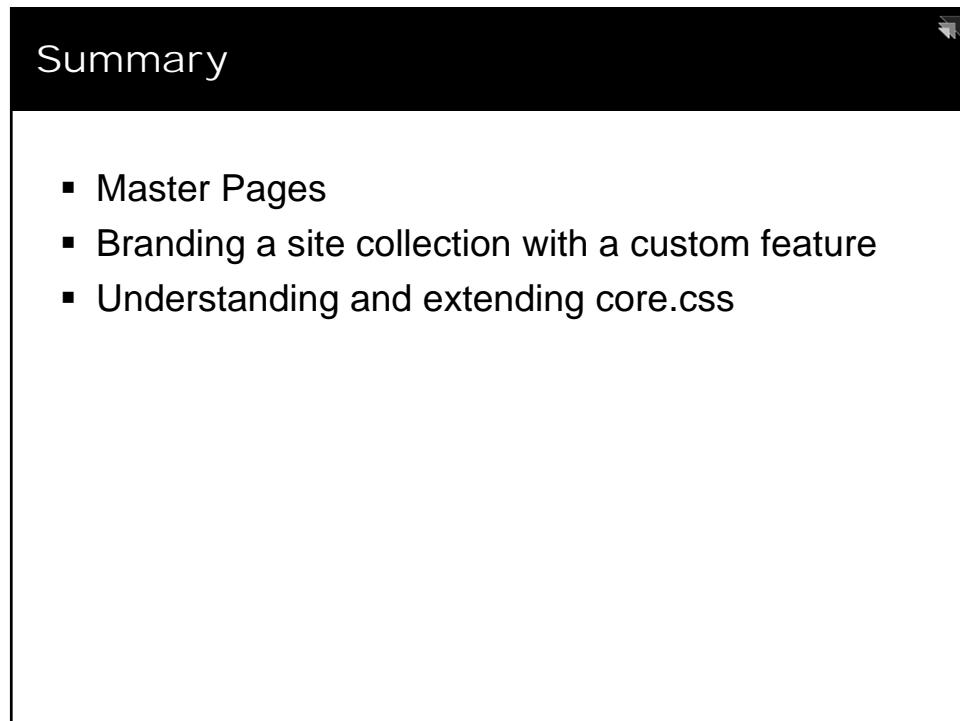
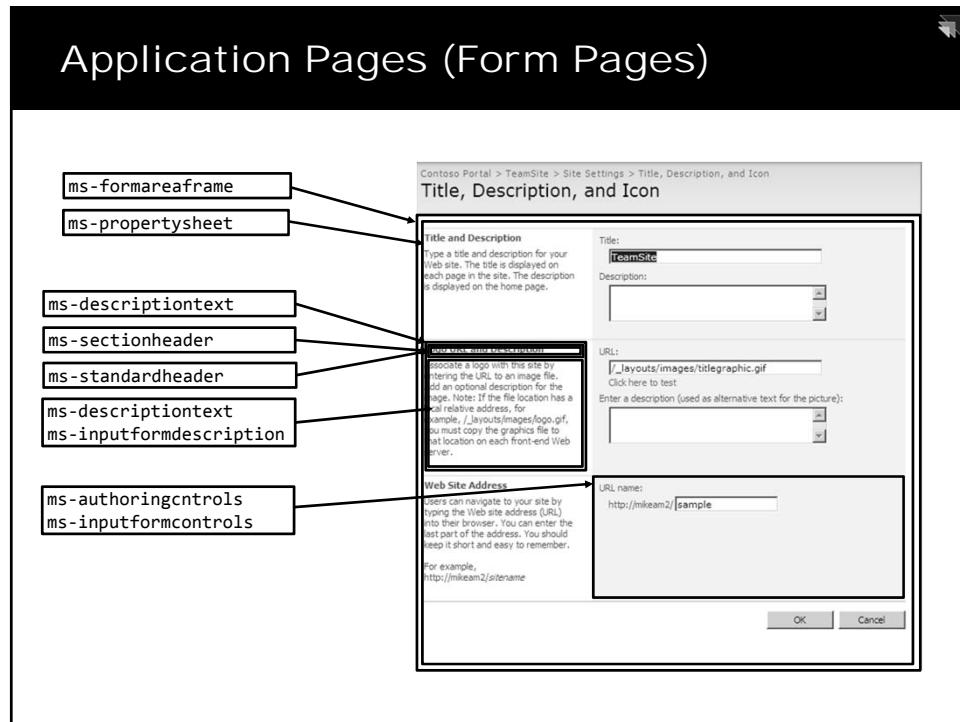
## Web Parts



## List Views









## Developing Web Parts

Creating User Interface Components that  
Support Customization and Personalization

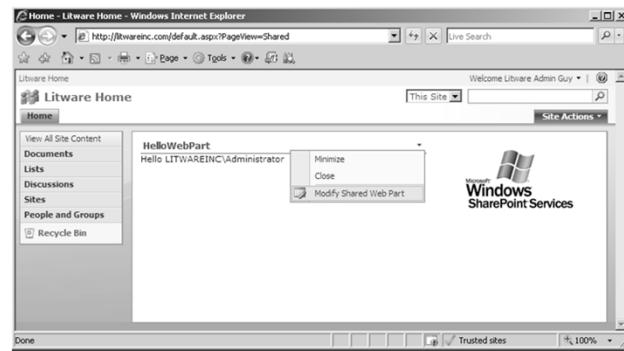


## Agenda

- 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

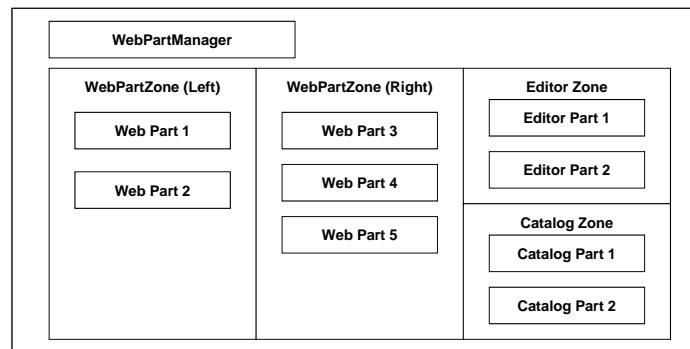


## Web Part History

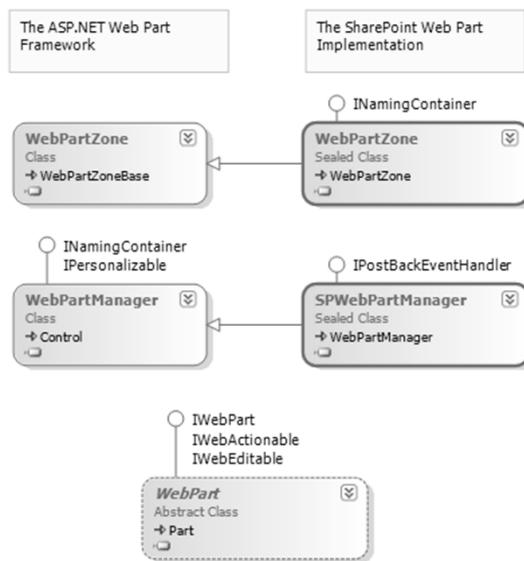
- Windows SharePoint Services 2.0 (WSS V2)
  - Designed with its own Web Part infrastructure
  - WSS serializes/stores/retrieves personalization data
- ASP.NET 2.0
  - Designed with a newer universal Web Part infrastructure
  - Serializes/stores/retrieves personalization data
  - More flexible and more 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

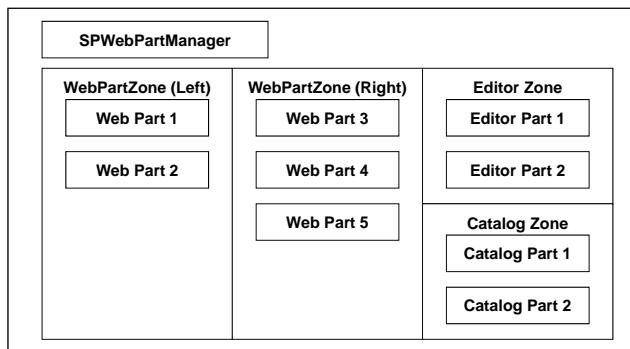


## SharePoint's Web Part Implementation



## 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 LftwareWebParts {

    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 LftwareWebParts {
    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

Add Web Parts - Webpage Dialog

Add Web Parts to Left

Lists and Libraries

- Sales Leads
  - A list used to track sales leads

All Web Parts

- Hello World Web Part**
  - Use to create classic Hello World excitement
- Litware Feed List Web Part**
  - Use to provide list of WSS lists to feed RSS View Web Part
- Litware RSS View Web Part**
  - Use to create viewable RSS feed for any source

Miscellaneous

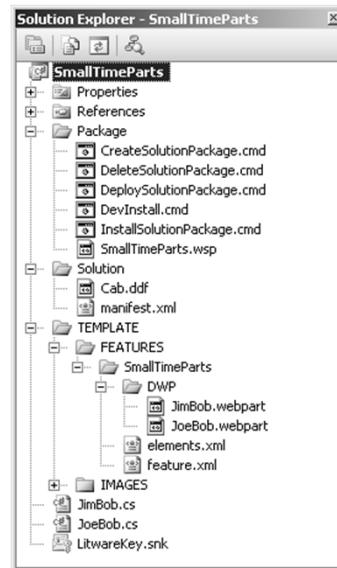
- Content Editor Web Part**
  - Use for formatted text, tables, and images.
- Form Web Part**
  - Use to connect simple form controls to other Web Parts.
- Image Web Part**
  - Use to display pictures and photos.

Advanced Web Part gallery and options

Add Cancel

## 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>
```

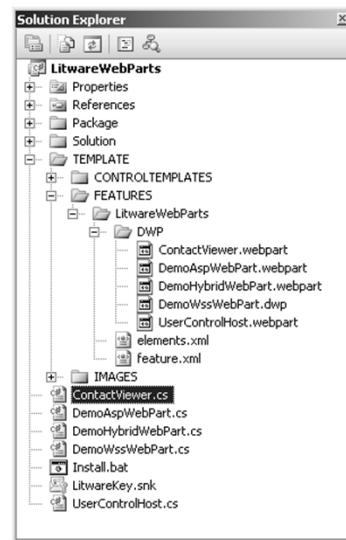
## Solution Manifest for WP Deployment

```
<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>
```

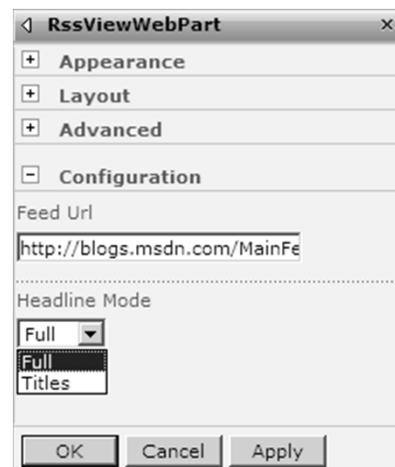
## Demo: LitwareWebParts

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



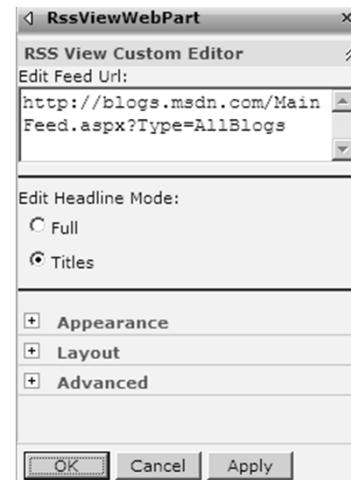
## Standard Editor Parts

- WSS provides standard editor parts



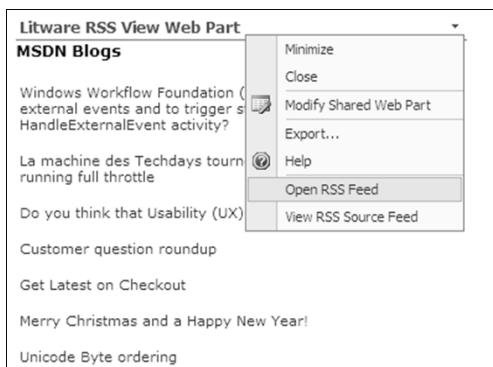
## A Custom Editor Part

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



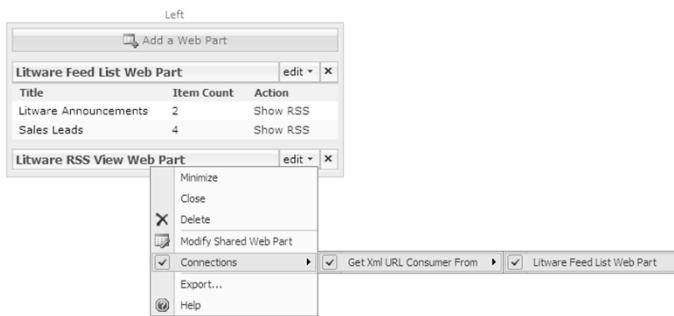
## Web Part Verbs

- 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 on a rocky path, possibly a hiker, against a backdrop of a cloudy sky. The Critical Path Training logo is in the top right corner.

**Building Web Parts with  
User Controls and AJAX**

Creating the Web 2.0 Experience

## Agenda

- Hosting User Controls in a Web Part
- Why Do we need AJAX?
- AJAX integration with SharePoint
- Using Ajax and User Controls together
- Object-oriented JavaScript
- The AJAX Control Toolkit

## Hosting User Controls

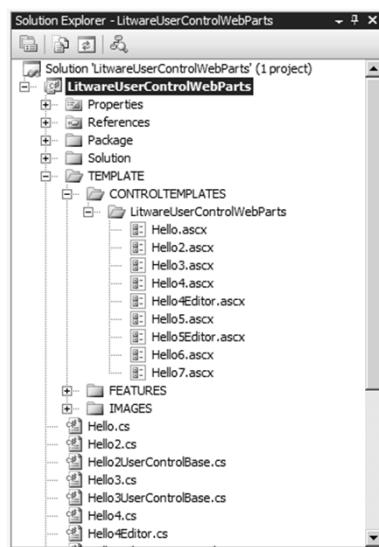
```
using System;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.webParts;

namespace LitwareUserControlWebParts {
    public class Hello : WebPart {

        // field to hold onto UserControl instance
        protected UserControl userControl;
        protected string path = @"/_controltemplates/MyDir/MyUserControl.ascx";

        // load .ascx file and create UserControl instance
        protected override void CreateChildControls() {
            try {
                this.Controls.Clear();
                userControl = (UserControl)this.Page.LoadControl(path);
                this.Controls.Add(userControl);
            }
            catch (Exception ex) {
                string errMessage = ex.GetType().ToString() + ": " + ex.Message;
                this.Controls.Add(new LiteralControl(errMessage));
            }
        }
    }
}
```

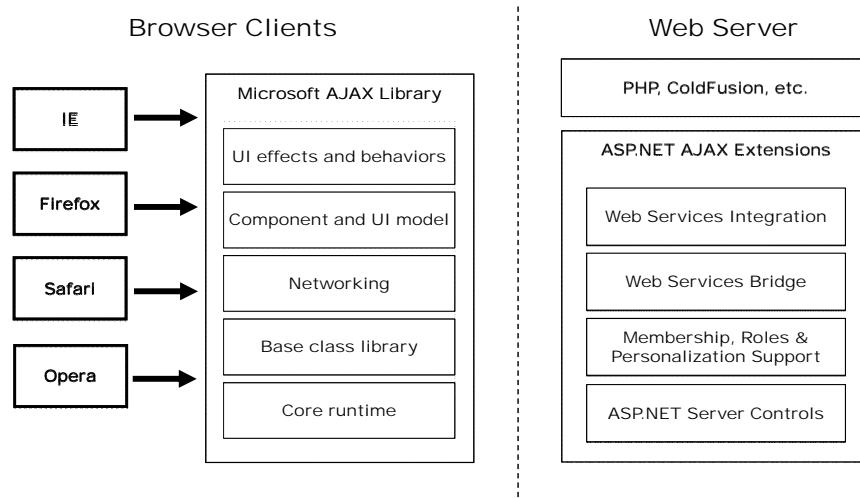
## Demo: LitwareUserControls



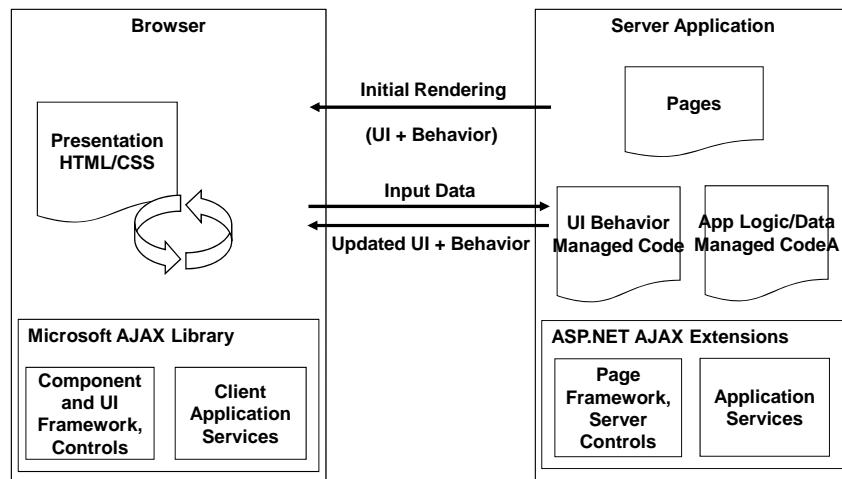
## What's in ASP.NET AJAX

- AJAX = Asynchronous JavaScript + XML
- Dynamic client-side interaction
- Asynchronous data retrieval
- Partial rendering

## What's in ASP.NET AJAX

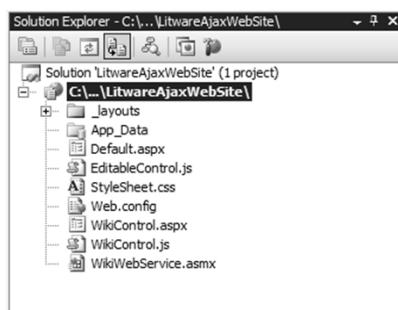


## Choosing an AJAX Application Model



## Demo: LitwareAjaxWebSite

- AJAX-enabled ASP.NET Web Site
  - Demonstrates client-side JavaScript programming
  - Demonstrates client callback to .asmx Web Service



## Object-oriented JavaScript

- Namespaces

```
if (typeof(Litware) == 'undefined')
    Type.registerNamespace('Litware');
```

- Functions as objects

```
if (typeof(Litware) == 'undefined')
    Type.registerNamespace('Litware');

Litware.EditableControl = function(element){
    element.editControl = this;
    this.element = element;
}
```

## Prototypes as Class Definitions

```
if (typeof(Litware) == 'undefined')
    Type.registerNamespace('Litware');

Litware.EditableControl = function(element){
    element.editControl = this;
    this.element = element;
}

Litware.EditableControl.prototype = {
    element : null,
    editElement : null,

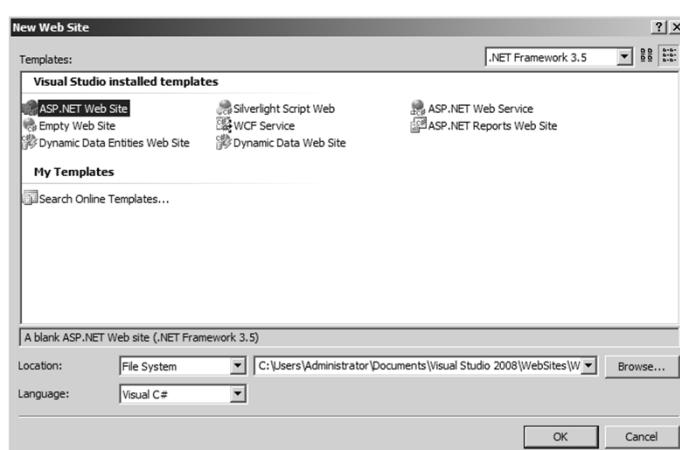
    initialize : function(text){
        this.editElement = document.createElement('DIV');
        this.editElement.style.border = '1px solid';
        this.editElement.style.padding='3px';
        this.editElement.innerHTML = text;
        this.element.appendChild(this.editElement);
    }
}
```

## Wiring Up Event Handlers

```
Litware.EditableControl.prototype = {  
    element : null,  
    editElement : null,  
  
    initialize : function(text){  
        // create an initialize control using DOM  
        this.editElement = document.createElement('DIV');  
        this.editElement.style.border = '1px solid';  
        this.editElement.style.padding='3px';  
        this.editElement.innerHTML = text;  
        this.element.appendChild(this.editElement);  
        // wire up event handler  
        $addHandler(this.editElement,'dblclick',  
                    Litware.EditableControl.MakeEditable);  
    },  
  
    makeEditable : function(){  
        // code for event handler here  
    }  
}
```

## Creating a JavaScript Component for AJAX

- Create a new ASP.NET-enabled Web Site



## The ASP.NET AJAX ScriptManager



## Initializing an ASP.NET Page

```
<body>
    <form id="form1">...</form>
    <script type="text/javascript" language="javascript">
        if (typeof(Litware) == 'undefined')
            Type.registerNamespace('Litware');

        Litware.PageLoad = function(){
            var control = $get('MainPlaceHolder');
            window.wiki = new Litware.EditableControl(control);
            window.wiki.initialize('When SharePoint Met AJAX');
        }

        // Add the OnLoad to the AJAX Runtime On Load event
        Sys.Application.add_load(Litware.PageLoad);

        document.body.style.padding='3px';
    </script>
</body>
```

## Adding a Web Service

```
<%@ WebService Language="C#" Class="WikiWebService" %>

using System;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;

[WebService(Namespace = "http://HelloAjax.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
[ScriptService]
public class WikiWebService : WebService {

    private const string DefaultContent = "Living on the server-side!";
    private const string keyPrefix = @"litwiki_";

    [WebMethod]
    public string GetContent(string wikiID) {
        return ((string)Application[keyPrefix + wikiID] ?? 
            DefaultContent);
    }

    [WebMethod]
    public void SetContent(string wikiID, string wikiContent) {
        Application[keyPrefix + wikiID] = wikiContent;
    }
}
```

## Adding a Service to ScriptManager

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
    <Scripts>
        <asp:ScriptReference Path="Scripts\EditableControl.js" />
        <asp:ScriptReference Path="Scripts\WikiControl.js" />
    </Scripts>
    <Services>
        <asp:ServiceReference Path="WikiWebService.asmx" />
    </Services>
</asp:ScriptManager>
```

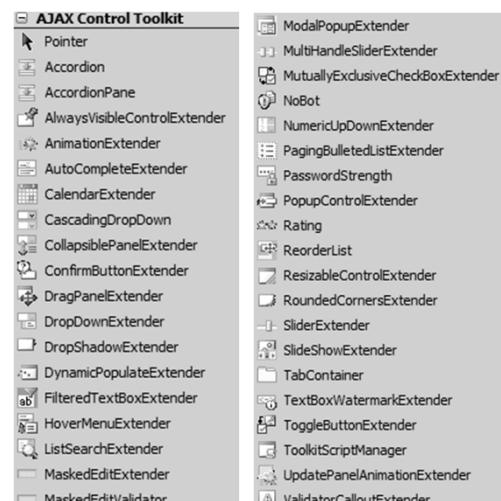
## Using UpdatePanel

- Requires a ScriptManager
- Enables partial page updates
- Converts postbacks

```
<div>
    <asp:ScriptManager ID="ScriptManager" runat="server" />
    <asp:UpdatePanel ID="UpdatePanel1" UpdateMode="Conditional" runat="server">
        <ContentTemplate>
            <fieldset>
                <legend>UpdatePanel content</legend>
                <!-- Other content in the panel. -->
                <%=DateTime.Now.ToString() %>
                <br />
                <asp:Button ID="Button1" Text="Refresh Panel" runat="server" />
            </fieldset>
        </ContentTemplate>
    </asp:UpdatePanel>
</div>
```

## The AJAX Control Toolkit

- Contains about 40 controls:
  - Autocomplete
  - Calendar
  - CascadingDropdown
  - CollapsiblePanel
  - MaskedEdit
  - MultiHandleSlider
  - PopupControl
  - ResizableControl
  - RoundedCorners
  - PasswordStrength
  - Tabs

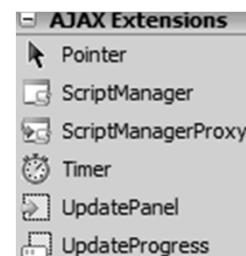


## What You Now Know About ASP.NET AJAX

- Summarizing the core architecture of AJAX
  - Library of XML APIs on server
  - JavaScript components on client
  - Maintain user interface
  - Partial rendering
  - Issue XML data requests asynchronously
  - Call Web Service operations asynchronously
  - AJAX Control Toolkit

## Integrating AJAX in SharePoint

- Install .NET Framework 3.5
- ASP.NET AJAX Support in Visual Studio 2008
- Extend web.config of IIS Web Application with ASP.NET Exttensions
  - Existing Web Parts can easily be wrapped to serve as AJAX-enabled Web Parts
  - Use controls from the AJAX Control Toolkit
  - Write your own controls



## Using AJAX in SharePoint

- Not officially support until service pack 1
  - Requires entries to web.config

```
<!-- entries for using AJAX in SharePoint Web Application -->
<configuration>
  <system.web>

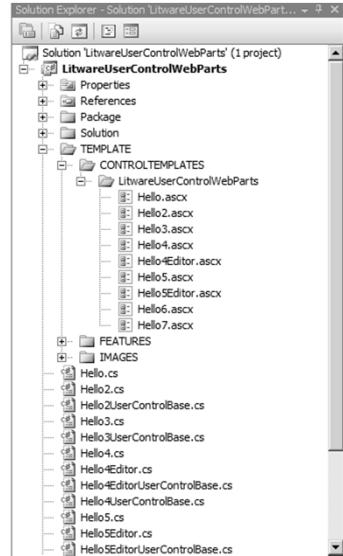
    <!-- three httpHandlers entry -->
    <httpHandlers>
      <add verb="*" path="*.asmx" validate="false"
          type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
      <add verb="*" path="*_AppService.axd"
          type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
      <add verb="GET,HEAD" path="ScriptResource.axd"
          type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
    </httpHandlers>

    <!-- one httpModules entry -->
    <httpModules>
      <add name="ScriptModule"
          type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
    </httpModules>
  </system.web>
</configuration>
```

## Demo: Ajaxification

- Using Visual Studio 2008 to upgrade the web.config

## Demo: LitwareUserControls



## Summary

- Hosting User Controls in a Web Part
- Why Do we need AJAX?
- AJAX integration with SharePoint
- Using Ajax and User Controls together
- Object-oriented JavaScript
- UpdatePanel
- AJAX Control Toolkit



# Silverlight

Lighting Up SharePoint with Silverlight



## Agenda

- What is Silverlight?
- SharePoint Configuration
- Silverlight 3 and SharePoint
  - Get Started
  - Deployment
  - Data Transfer between Silverlight and SharePoint
  - Data binding



## What is Silverlight?

- Rich User Presentation (XAML + Code-Behind)
- Silverlight 1.0 is 100% JavaScript-driven
- Silverlight 2 uses managed code (small .NET Framework footprint)
- Silverlight 3 supports .NET RIA Services
- Applications are packaged as ZIP files (.XAP extension)



## Getting Started with Silverlight

- Silverlight 3 Runtime
- Silverlight 3 SDK
- Visual Studio 2008
  - Microsoft Silverlight Tools Visual Studio 2008
    - Silverlight 3 Application Project Template
    - Silverlight 3 Class Library Project Template
    - Silverlight 3 Navigation Application Template
    - XAML IntelliSense
    - Limited Design Experience
- Expression Blend 3
  - Rich Design Experience
  - Sync options with project in Visual Studio 2008



## Silverlight and SharePoint

- Sample scenarios
  - Complex and dynamic interaction with data in dashboard pages (charting, reporting, ...)
  - Visualize multimedia data stored in SharePoint
  - Rich navigation controls
  - Interactive field types, Web parts and pages
  - Off-load more work to the clients



## SharePoint Server Configuration

- WSS 3.0 Service Pack 1 adds support for .NET Framework 3.5
  - .NET Framework 3.5 needed for Silverlight 3
  - .NET Framework 2.0 is fine for Silverlight 1.0
- MIME type registration of the .XAP file extension for IIS Web application
  - Format: application/x-silverlight-app
  - Automatically included as of IIS 7



## XAML Essentials

### ▪ App.xaml

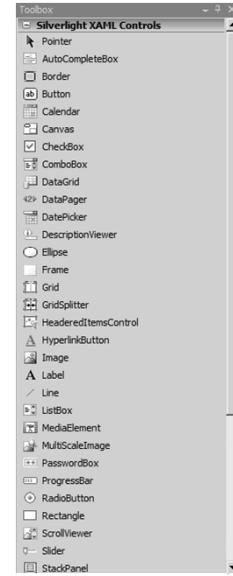
- Entry Point of the Silverlight application
- Can contain Application Resources such as styles

### ▪ MainPage.xaml

- UserControl
- Canvas vs Grid
- TextBlock
- StoryBoard

### ▪ MyControl.xaml

- Re-usable control



## XAML Code Behind

### ▪ App.xaml.cs

- Application\_Startup is place where incoming parameters are processed and MainPage object is created

### ▪ MainPage.xaml.cs

- Handle events
- Dynamically create XAML elements
- Connect to browser DOM
- Connect to external data (e.g. Web Services)

### ▪ MyControl.xaml.cs

- Code-behind for control

## Where to deploy the XAP?

- **Scope = Site**
  - Document Library in Site
- **Scope = IIS Web Application**
  - ClientBin folder in IIS Web App folder
- **Scope = Server**
  - Layouts folder
  - ControlTemplates folder
- **Scope = Web Part**
  - Embedded Resource



## Hosting Silverlight in Web Part - 1

- **Silverlight.js**
  - Comes with the Silverlight 3 SDK
  - Location:  
C:\Program Files\Microsoft SDKs\Silverlight\v3.0\Tools
  - Must be deployed in the 12 Hive
  - Must be loaded in OnPreRender event

```
protected override void OnPreRenderEventArgs e)
{
    base.OnPreRender(e);
    // Include the required javascript file.
    ClientScriptManager cs = Page.ClientScript;
    if (!cs.IsClientScriptIncludeRegistered("sl_javascript"))
        cs.RegisterClientScriptInclude(this.GetType(), "sl_javascript", "/_LAYOUTS/Silverlight.js");
}
```

- **Heavy Silverlight usage in SharePoint:**
  - Deploy as ControlTemplate to AdditionalPageHead delegate control



## Hosting Silverlight in Web Part - 2

- Add Silverlight object in Render method

```
protected override void Render(HtmlTextWriter writer)
{
    base.Render(writer);

    writer.WriteLine("<object id='SilverlightHost'>");
    writer.WriteLine(" width='" + width + "'");
    writer.WriteLine(" height='" + height + "'");
    writer.WriteLine(" data='data:application/x-silverlight-2,'");
    writer.WriteLine(" type='application/x-silverlight-2' >");
    writer.WriteLine(" <param name='source' value='" + source + "'/>");
    writer.WriteLine(" <param name='initParams' value='" + initParams + "'/>");
    writer.WriteLine(" <a href='http://go.microsoft.com/fwlink/?LinkId=149156' >");
    writer.WriteLine(" <img alt='Get Microsoft Silverlight!' style='text-decoration: none;' >");
    writer.WriteLine(" </a> ");
    writer.WriteLine(" </object>");
}
```

## Demo: Silverlight Enabled Web Part

- Important Concepts
  - Developing a Silverlight enabled Web Part
  - Packaging the Web Part
  - Deploying the Web Part



## Communication with Silverlight

- **Silverlight.InitParameters**
  - Small amount of initialization data
  - Pass as comma delimited string  
Key1=val ue1, key2=val ue2, ...
  - Process as IDictionary in Application\_StartUp
- **Hidden INPUT Field (two-way)**
  - Pass client ID with InitParameters
  - Cross-Browser compatible
- **XML Data Island**
  - Pass ID with InitParameters
  - Works only in IE but useful for XML data



## Consuming Web Services

- **OOB SharePoint Web Services**
  - Use System.Net.HttpWebRequest (Asynchronously)
- **Custom SharePoint Web Services**
  - Enable for AJAX Scripting
  - Deploy in ISAPI (run in context of SharePoint)
  - AJAX Client Side Web Service Proxies are possible
- **Custom SharePoint WCF Services**
  - Does not run in context of SharePoint
  - Service proxy in Silverlight application



## Data Binding in XAML

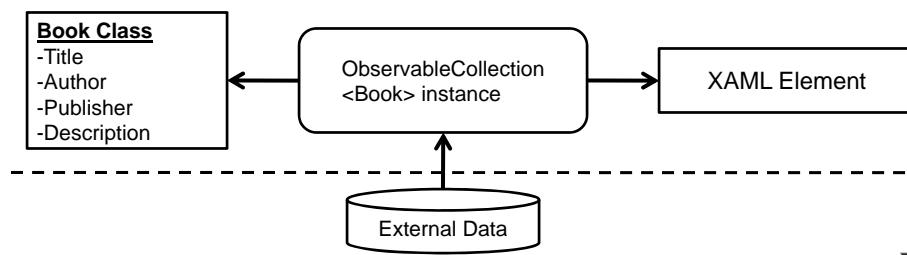
- Grid and StackPanel elements
- Define {Binding} in XAML

```
<ListBox x:Name="BooksList" Style="{StaticResource BooksList}"  
        SelectionChanged="BooksList_SelectionChanged">  
    <ListBox.ItemTemplate>  
        <DataTemplate>  
            <StackPanel Orientation="Horizontal">  
                <!-- Thumbnail -->  
                <Image Source="{Binding ThumbNail}" Style="{StaticResource ThumbNailPreview}" />  
                <!-- Title -->  
                <TextBlock Text="{Binding Title}" Margin="5" Style="{StaticResource TitleBlock}"  
                          FontWeight="Bold" />  
                <!-- Author -->  
                <TextBlock Text="{Binding Author}" Margin="5" Style="{StaticResource AuthorBlock}" />  
            </StackPanel>  
        </DataTemplate>  
    </ListBox.ItemTemplate>  
</ListBox>
```



## Data Binding in Code Behind

- Data is represented by class
- Objects of class are collected in **ObservableCollection<class>**
- Controls expose **ItemsSource** property for binding



## Updating data

- In XAML:
  - Set BindingMode property
    - OneWay
    - OneTime
    - TwoWay
- In Book class:
  - Implement INotifyPropertyChanged
  - Raise NotifyPropertyChanged



## Demo: Data binding to SharePoint

- Important Concepts
  - InitParams
  - HttpWebRequest
  - ItemsSource
  - DataContext

The screenshot shows a Silverlight application interface. At the top, there is a 'Solution Explorer' window titled 'Solution 'SL.XAML.BookViewer'' (1 project) showing files like SL.XAML.BookViewer.cs, Properties, References, App.xaml, Book.cs, BookDetails.xaml, Page.xaml, and Rotator.xaml. Below it is a 'Silverlight Application Wrapper Web Part' titled 'My Favorite Books'. This part displays a list of three books with their covers, titles, and authors:

Book Cover	Title	Author
	Professional SharePoint 2007 Web Content Management Development	Andrew Connell
	Inside Microsoft Office Server 2007	Patrick Tisseghem
	Inside Microsoft Windows SharePoint Services 3.0	Ted Pattison



## Downloading Media

- WebClient object
  - Streams
  - Asynchronous
  - OpenReadCompleted
- Pass Incoming Stream to
  - Pictures go into an Image Element
  - Videos go into a MediaElement



## Data Across Domains

- By default not allowed
- If required, add client access policy to the root of the site via SharePoint Designer or Feature

```
<?xml version="1.0" encoding="utf-8" ?>
- <access-policy>
  - <cross-domain-access>
    - <policy>
      - <allow-from>
        <domain uri="*" />
      </allow-from>
      - <grant-to>
        <resource path="/" include-subpaths="true" />
      </grant-to>
    </policy>
  </cross-domain-access>
</access-policy>
```



## Summary

- Why do we need Silverlight
- Silverlight Integrating with SharePoint
- Where to deploy the Silverlight application
- Communication between Silverlight and SharePoint
- Data binding
- Downloading Media





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 logo for "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

Category	Column Name	Type	Scope
Base Columns	Appended-Only Comments	Multiple lines of text	Team Site
	Comments	Multiple lines of text	Team Site
	Date/Time	Date and Time	Team Site
	Expires	Date and Time	Team Site
	Language	Choice	Team Site
	Start Date	Date and Time	Team Site
	Title	Single line of text	Team Site
	URL	Hyperlink or Picture	Team Site
Custom Columns	Category	Lookup	Team Site
	Published	Date and Time	Team Site

## 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

Select New Columns  
Select from the list of available column templates to add new columns to this list.

Select available templates from: All Groups

Available Column Templates:

- % Complete
- Address
- Assigned To
- Author
- Body
- Business Phone
- Category
- City
- Comments

New fields to add to the content type:

Add Remove

## 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
<b>Base Columns</b>		
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
<b>Core Contact and Calendar Columns</b>		
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

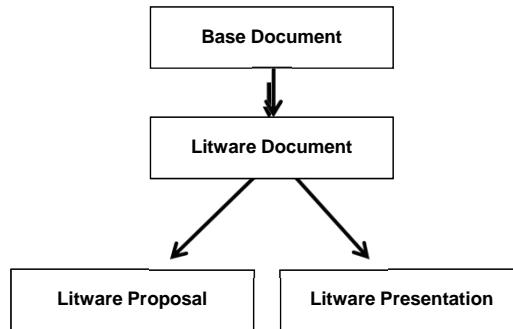
## Supporting Multiple Content Types

- 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

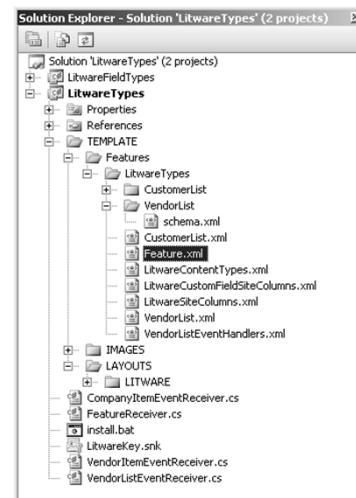
Site Content Type Gallery

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
<b>Document Content Types</b>		
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
<b>Folder Content Types</b>		
Discussion	Folder	Litware Inc
Folder	Item	Litware Inc
<b>List Content Types</b>		
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

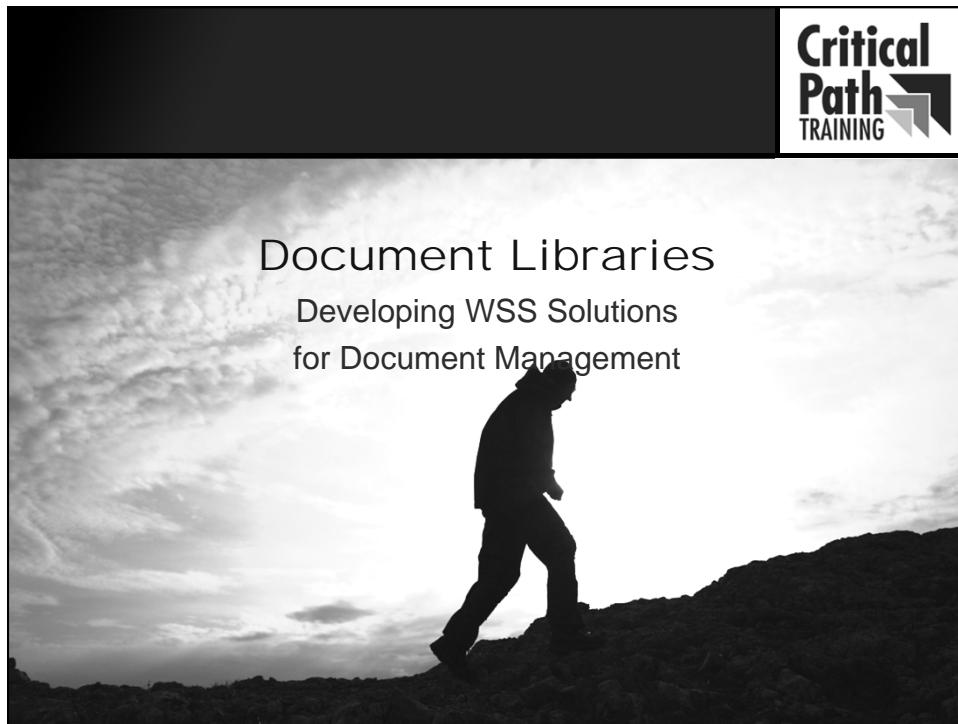
```
namespace Litware {
    public class TimesheetEventReceiver : Microsoft.SharePoint.SPItemEventReceiver {
        public override void ItemUpdating(SPItemEventProperties properties) {
            SPWeb web = properties.OpenWeb();
            SPListItem timesheet = web.Lists[properties.ListId].GetItemById(properties.ListItemId);
            // 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



## Agenda

- Programming with the SPDocumentLibrary class
- Documents as SPListItem and SPFile objects
- Libraries with custom document templates
- InfoPath and Forms Libraries
- The Office Open XML File Formats

## Demo Feature: Document Manager

The screenshot shows a SharePoint 'Customers' list and a 'DocumentManager' solution in Visual Studio. The list contains three items: Decker, Barbara; Fitzgerald, Charles; and Garcia, Cesar. The Solution Explorer shows files for the DocumentManager feature, including TestData, Features, Images, Layouts, and various .cs files.

## SPDocumentLibrary Class

- Document Libraries are specialized lists
  - SPDocumentLibrary inherits from SPList

```
SPWeb site = SPContext.Current.Web;

foreach (SPList list in site.Lists) {
    if (list is SPDocumentLibrary && !list.Hidden) {
        SPDocumentLibrary docLib = (SPDocumentLibrary)list;
        SPFolder folder = docLib.RootFolder;
        TreeNode docLibNode = new TreeNode(docLib.Title,
                                            docLib.DefaultViewUrl,
                                            DOCLIB_IMG);
        LoadFolderNodes(folder, docLibNode);
        treeSitesFiles.Nodes.Add(docLibNode);
    }
}
```

## Inspecting Documents within a Site

```

protected void LoadFolderNodes(SPFolder folder, TreeNode folderNode) {
    foreach (SPFolder childFolder in folder.SubFolders) {
        if (childFolder.Name != "Forms") {
            TreeNode childFolderNode = new TreeNode(childFolder.Name,
                childFolder.Name,
                FOLDER_IMG);
            LoadFolderNodes(childFolder, childFolderNode);
            folderNode.ChildNodes.Add(childFolderNode);
        }
    }

    foreach (SPFile file in folder.Files) {
        TreeNode fileNode;
        fileNode = new TreeNode(file.Name, file.Name, FILE_IMG);
        folderNode.ChildNodes.Add(fileNode);
    }
}

```

## Adding Files to a Document Library

```

protected void cmdCreateDocument_Click(object sender, EventArgs e) {
    SPSite siteCollection = SPContext.Current.Site;
    SPWeb site = SPContext.Current.Web;
    Guid libraryID = new Guid(lstTargetLibrary.SelectedValue);
    SPDocumentLibrary library = (SPDocumentLibrary)site.Lists[libraryID];

    string documentName = txtFileName.Text;
    string libraryRelativePath = library.RootFolder.ServerRelativeUrl;
    string libraryPath = siteCollection.MakeFullUrl(libraryRelativePath);
    string documentPath = libraryPath + "/" + documentName;

    Stream documentStream = new MemoryStream();
    StreamWriter writer = new StreamWriter(documentStream);
    writer.Write(txtDocumentBody.Text);
    writer.Flush();

    Hashtable docProperties = new Hashtable();
    docProperties["vti_title"] = "This is a test title";
    docProperties["Color"] = "Green";
    site.Files.Add(documentPath, documentStream, docProperties, true);

    Response.Redirect(libraryPath);
}

```

## Provisioning A Document Library

```
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">

<ListInstance
  FeatureId="00BFEA71-E717-4E80-AA17-D0C71B360101"
  TemplateType="101"
  Id="Proposals"
  Description="Document Library for proposals"
  OnQuickLaunch="True"
  Title="Proposals"
  Url="Proposals" />

<Module Name="TestData" List="101" Path="TestData" Url="Proposals" >
  <File Url="Adventure Works Proposal.docx" Type="GhostableInLibrary" />
  <File Url="Contoso Proposal.docx" Type="GhostableInLibrary" />
  <File Url="Wingtip Toys Proposal.docx" Type="GhostableInLibrary" />
</Module>

<Module Name="WordTemplate" List="101" Url="Proposals/Forms">
  <File Url="ProposalTemplate.dotx" Type="GhostableInLibrary" />
</Module>

</Elements>
```

## Configuring a Document Template

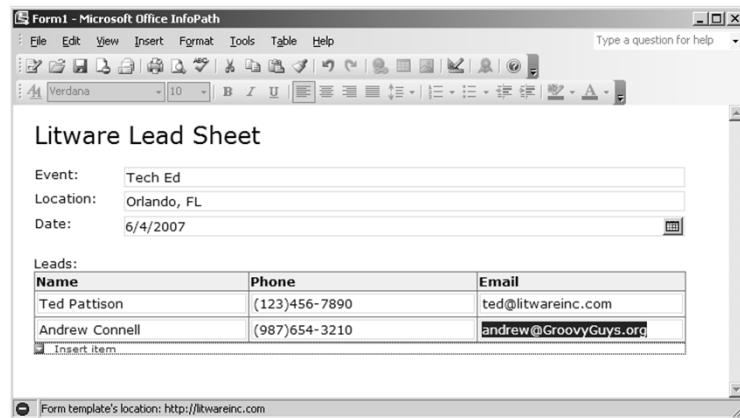
```
namespace DocumentManager {
  public class FeatureReceiver : SPFeatureReceiver {

    public override void FeatureActivated(SPFeatureReceiverProperties properties) {
      SPWeb site = (SPWeb)properties.Feature.Parent;

      SPDocumentLibrary libProposals = (SPDocumentLibrary)site.Lists["Proposals"];
      string templateUrl = @"Proposals/Forms/ProposalTemplate.dotx";
      libProposals.DocumentTemplateUrl = templateUrl;
      libProposals.Update();
    }

    public override void FeatureDeactivating(SPFeatureReceiverProperties properties)
    public override void FeatureInstalled(SPFeatureReceiverProperties properties)
    public override void FeatureUninstalling(SPFeatureReceiverProperties properties)
  }
}
```

## InfoPath and Forms Libraries

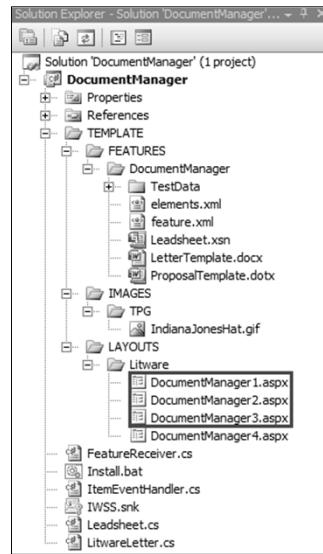


## Provisioning A Forms Library

```
<ListInstance  
FeatureId="00BFEA71-1E1D-4562-B56A-F05371BB0115"  
Id="Leadsheets"  
Description="Leadsheets for tracking sales leads"  
TemplateType="115"  
OnQuickLaunch="True"  
Title="Leadsheets"  
Url="Leadsheets" />  
  
<Module Name="LeadsheetTemplate" List="115" Url="Leadsheets/Forms">  
    <File Url="Leadsheet.xsn" Type="GhostableInLibrary" />  
</Module>
```

```
SPDocumentLibrary libLeadsheets;  
libLeadsheets = (SPDocumentLibrary)site.Lists["Leadsheets"];  
libLeadsheets.DocumentTemplateUrl = @"Leadsheets/Forms/Leadsheet.xsn";  
libLeadsheets.EnableVersioning = true;  
libLeadsheets.EnableFolderCreation = false;  
libLeadsheets.ForceCheckout = true;  
libLeadsheets.Update();
```

## Demo: Document Manager



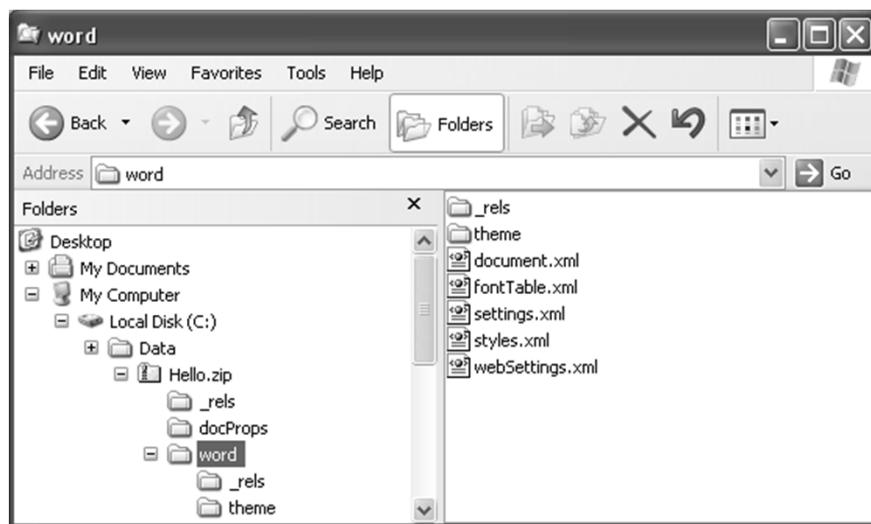
## Office Open XML File Formats

Macro-Free		Macro-Enabled	
Document	Template	Document	Template
docx	dotx	docm	dotm
pptx	potx	pptm	potm
xlsx	xltx	xlsm	xltm
Open Packaging Convention			

## Basic Open XML File Components

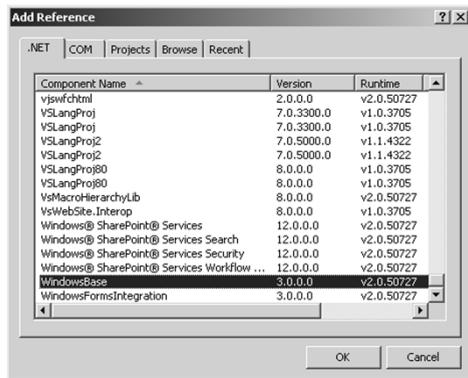
- Package
  - ZIP Container itself (e.g. Proposal.docx)
- Part
  - The “files” inside the package holding the content
  - Most parts are XML but could also be binary
- Content Types
  - Each part has a content type that is enforced on open
- Relationships
  - Associations between package and top-level parts
  - Associations between parent parts and child parts

## Looking Inside a Package

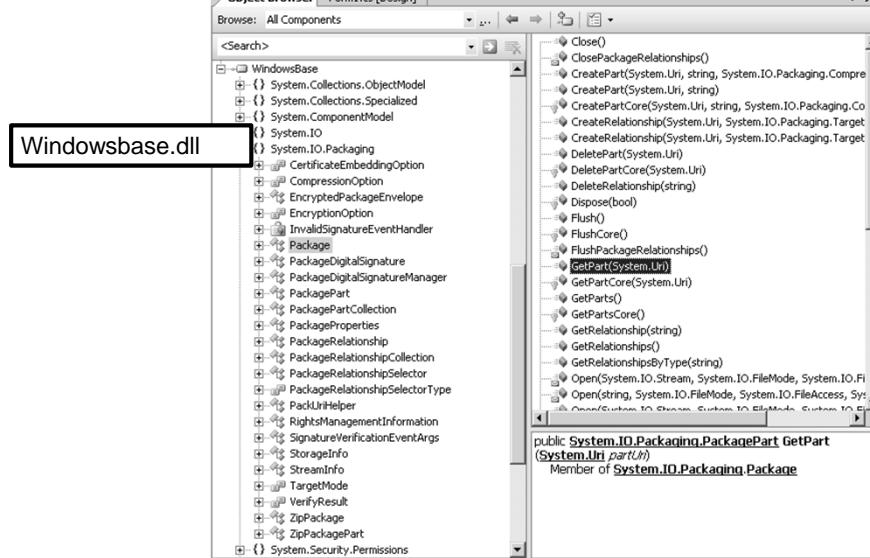


## Programming with Open XML formats

- Use classes in `System.IO.Packaging` namespace
  - Part of the `WindowsBase.dll`
  - .NET Framework 3.0

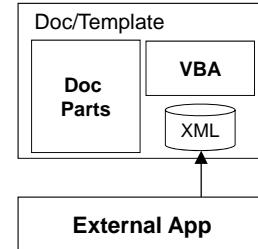


## System.IO.Packaging

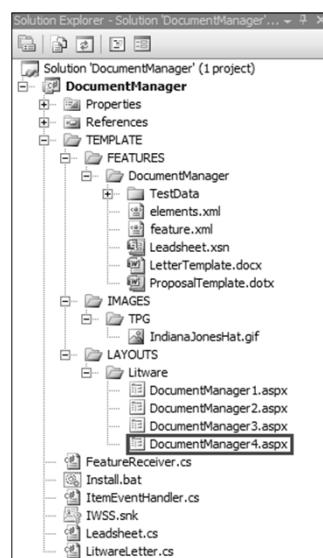


## Office XML Data Store

- Customer-defined XML stored separately rest of document parts
- Any XML can be stored
  - Document properties
  - WSS meta-data
  - Custom XML (with or without XML schema)
- XML data is available as an editable tree (using familiar DOM) within Word
- External applications (client/server) can process the store or populate the store



## Demo: Document Manager



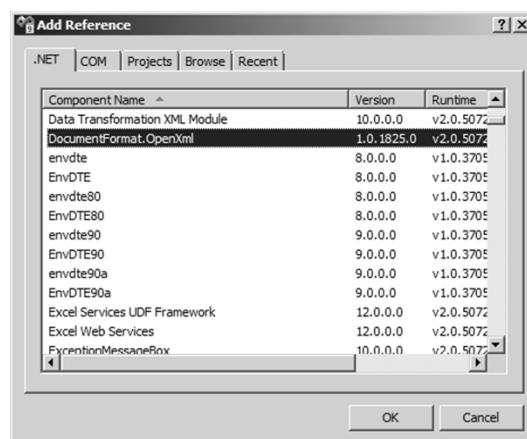
## Open XML Format SDK 1.0

- Simplifies the manipulation of OpenXML packages
- OpenXML API
  - Search
  - Document manipulation
  - Validation
  - Data update
- Download

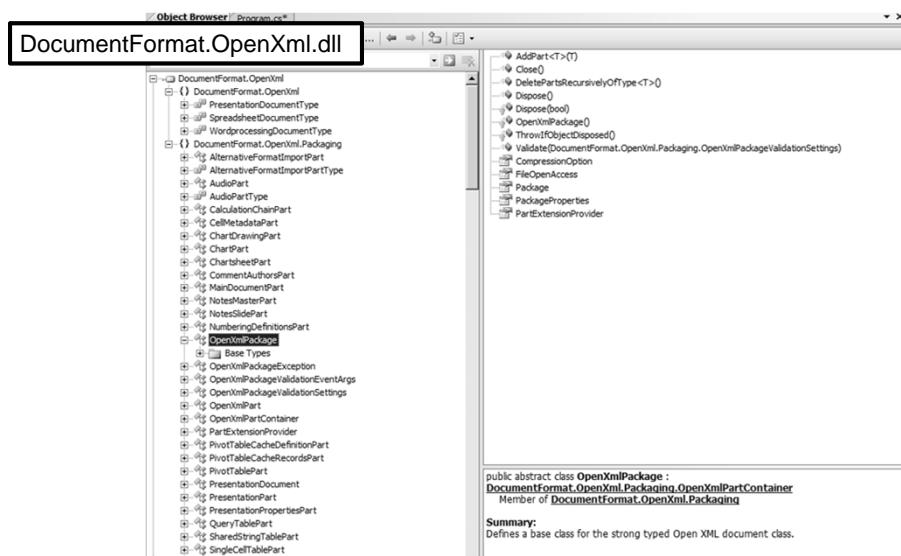
<http://www.microsoft.com/downloads/details.aspx?FamilyId=AD0B72FB-4A1D-4C52-BDB5-7DD7E816D046&displaylang=en>

## Programming with Open XML SDK

- Use DocumentFormat.OpenXml.dll
  - Classes inside Document.Format.OpenXml.Packaging namespace



## DocumentFormat.OpenXml.Packing

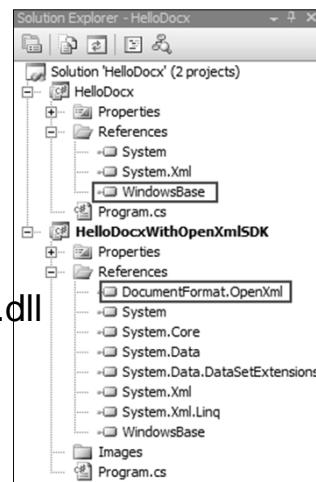


## Validation of a Document

```
// How to validate the contents of a document part against
// a collection of schemas.
public static void ValidDocumentContent(string document,
                                         List<string> schemaList)
{
    XmlSchemaSet schemas = new XmlSchemaSet();
    foreach (string schemaUri in schemaList)
    {
        schemas.Add(null, schemaUri);
    }
    using (WordprocessingDocument wordDoc =
        WordprocessingDocument.Open(document, false))
    {
        wordDoc.MainDocumentPart.ValidateXml(schemas, null);
    }
}
```

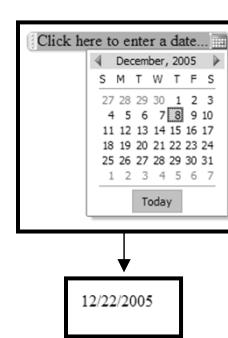
## Demo: HelloDocx

- HelloDocx:
  - WindowsBase.dll
- HelloDocxWithOpenXmlISDK:
  - OpenXml Format SDK
  - DocumentFormat.OpenXml.dll



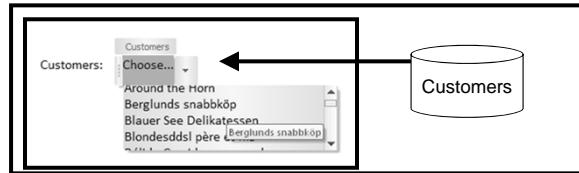
## Content Controls in Word 2007

- Evolution of customer-defined XML
  - No XML schema required
- Provide friendly end user exposure of structured content
- Each provides unique content restrictions
- Controls do not affect layout
- Controls can be grouped to lock them down as a unit
- Controls can be mapped to XML data
- Can be used by document parts
- Can insert controls using the Developer Tab
- Specialized options for each type of control
- Can set placeholder text for when it's empty



## XML Mapping

- Link content controls to nodes in the XML data store
- Mappings are created using standard XPath expressions
- Out of the box support for mapping to Office properties



## Summary

- Programming with the SPDocumentLibrary class
- Documents as SPListItem and SPFile objects
- Libraries with custom documents templates
- InfoPath and Forms Libraries
- The Office Open XML File Formats



## Site Definitions

The Origin of Life for a WSS Site



## Agenda

- What is a site definition?
- Built-in site definitions with WSS and MOSS
- The Global Site Definition
- Creating a site definition

## What is a Site Definition?

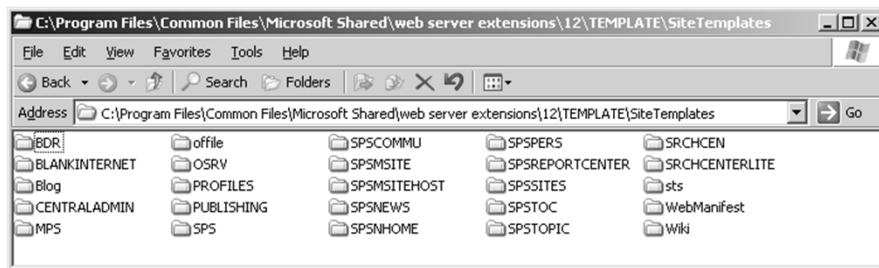
- Top-level component for site provisioning
  - Used to create site templates
  - Aggregates smaller, more modular definitions
- Every site is provisioned from a site definition
  - The association remains in place for lifetime of site
  - Each site definition has one or more configurations
  - Each configuration can serve as creatable site template

## What files make up a Site Definition?

- ONET.XML
  - Acts as manifest for site definition
- WEBTEMP.XML
  - Language-specific file used to activate site definition
- Page template files
  - Used to provision home page and secondary pages

## Standard SharePoint Site Definitions

- Out-of-box with WSS
  - STS, MPS, CENTRALADMIN, Blog, Wiki
  - All others in picture provided by MOSS



## Create A Site From A Site Definition

- Create a new site from Team Site template
  - Using default configuration for STS site definition

```
STSADM. EXE -o createsite -url http://ITI twarei nc. com  
-ownerI ogi n LI TWAREI NC\Administrator  
-owneremail I administrator@ITI twarei nc. com  
-site template STS
```

- Create a new site from Blank Site template
  - Using explicit configuration from STS site definition

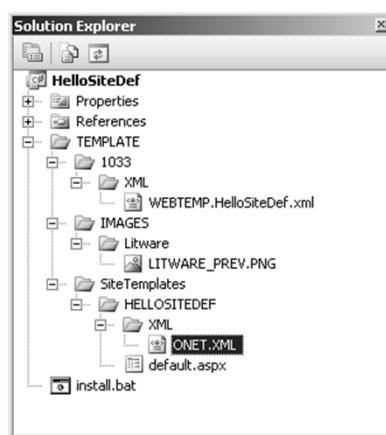
```
STSADM. EXE -o createsite -url http://ITI twarei nc. com  
-ownerI ogi n LI TWAREI NC\Administrator  
-owneremail I administrator@ITI twarei nc. com  
-site template STS#1
```

## The Global Site Definition

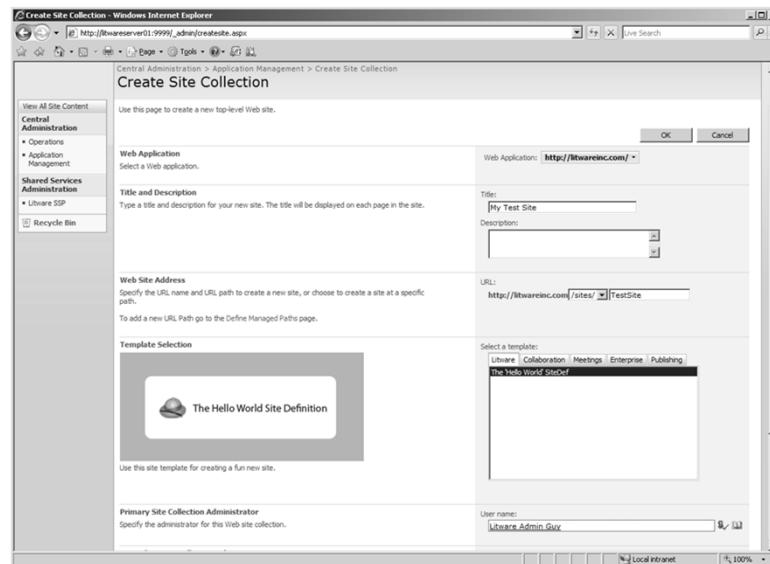
- Contains global provisioning instructions
  - What's inside affects provisioning for every site in farm
  - Eliminates redundant CAML required in WSS v2
- Global Site Definition defines and provisions
  - Master Page Gallery and default.master (all sites)
  - Web Part Gallery (top-level sites only)
  - List Template Gallery (top-level sites only)
  - Site Template Gallery (top-level sites only)
  - User Information Profile list (top-level sites only)

## Developing Site Definitions

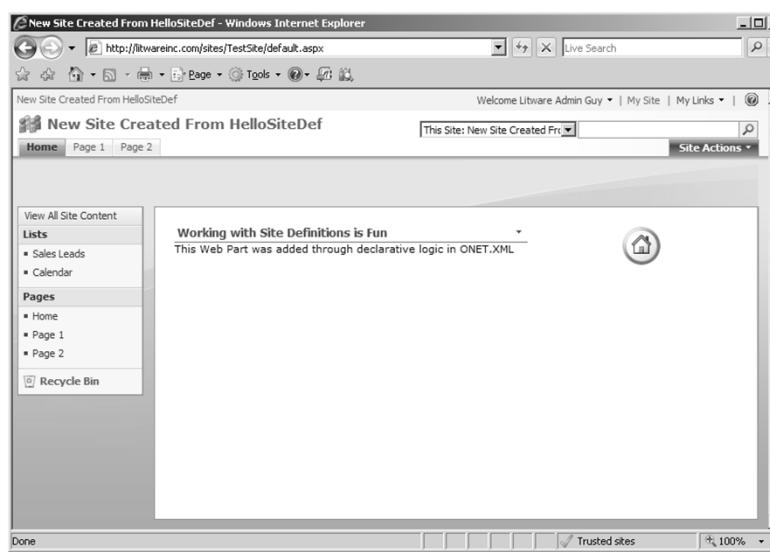
- Sample Visual Studio project: HelloSiteDef
  - Just build the project and test it out



## Testing HelloSiteDef

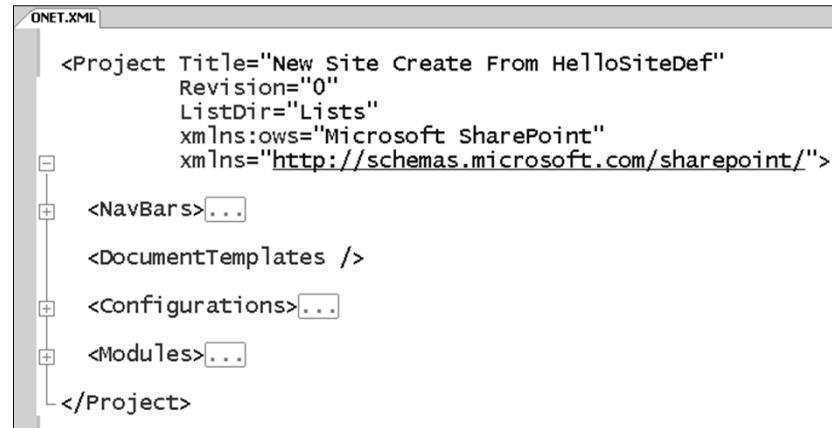


## A Freshly Created Test Site



## ONET.XML

- Provides top-level provisioning instructions



```
<Project Title="New Site Create From HelloSiteDef"
    Revision="0"
    ListDir="Lists"
    xmlns:ows="Microsoft SharePoint"
    xmlns="http://schemas.microsoft.com/sharepoint/">
    <NavBars>...
    <DocumentTemplates />
    <Configurations>...
    <Modules>...
</Project>
```

## Navbars

```
<NavBars>
    <NavBar Name="Top link bar"
        Separator="&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;" 
        Body="&lt;a ID='onettopnavbar#LABEL_ID#' href='#URL#'
            accesskey='J'&gt;#LABEL#&lt;/a&gt;" 
        ID="1002">
        <NavBarLink Name="Home" Url="default.aspx" />
        <NavBarLink Name="Page 1" Url="Page1.aspx" />
        <NavBarLink Name="Page 2" Url="Page2.aspx" />
    </NavBar>

    <NavBar Name="Pages"
        Prefix="&lt;table border=0 cellpadding=4 cellspacing=0&gt;"
        Body="&lt;tr&gt;&lt;td&gt;&lt;table border=0 cellpadding=0 cellspacing=0&gt;
            &lt;tr&gt;&lt;td&gt;&lt;td valign=top&gt;&lt;a
                ID='onetleftnavbar#LABEL_ID#'
                href="#URL#"&gt;#LABEL#&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/td&gt;&lt;/tr&gt;" 
        ID="100" alt="" 
        border=0&gt;&nbsp;&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;" 
        Suffix="&lt;/td&gt;" 
        ID="1028">
        <NavBarLink Name="Home" Url="default.aspx" />
        <NavBarLink Name="Page 1" Url="Page1.aspx" />
        <NavBarLink Name="Page 2" Url="Page2.aspx" />
    </NavBar>
```

## Configurations

```
<Configuration ID="0" Name="Default">
  <Lists>
    <!-- provision list instance form Contact list type -->
    <List Title="Sales Leads" Url="Sales Leads" QuickLaunchUrl="Sales Leads"
      FeatureId="00BFEA71-7E6D-4186-9BA8-C047AC750105" Type="105" />
    <!-- provision list instance form Contact list type -->
    <List Title="Calendar" Url="Calendar" QuickLaunchUrl="calendar"
      FeatureId="00BFEA71-EC85-4903-972D-EBE475780106" Type="106" />
  </Lists>

  <Modules>
    <Module Name="Default" />
  </Modules>

  <SiteFeatures>
    <!-- basic WSS web parts-->
    <Feature ID="00BFEA71-1c5E-4A24-B310-BA51C3EB7A57" />
  </SiteFeatures>

  <WebFeatures>
    <!-- contacts list -->
    <Feature ID="00BFEA71-7E6D-4186-9BA8-C047AC750105" />
    <!-- event list -->
    <Feature ID="00BFEA71-EC85-4903-972D-EBE475780106" />
    <!-- document library -->
    <Feature ID="00BFEA71-E717-4E80-AA17-D0C71B360101" />
  </WebFeatures>
</Configuration>
```

## Modules

```
<Modules>
  <Module Name="Default" Url="" >
    <File Url="default.aspx" Type="Ghostable">
      <!-- Add a Web Part to left zone -->
      <AllUsersWebPart WebPartZoneID="Left" WebPartOrder="0">...
      <!-- Add a Web Part to right zone -->
      <AllUsersWebPart WebPartZoneID="Right" WebPartOrder="0">...
    </File>

    <File Url="default.aspx" Name="Page1.aspx" Type="Ghostable">
      <AllUsersWebPart WebPartZoneID="Left" WebPartOrder="0">...
      <!-- Add a Web Part to right zone -->
      <AllUsersWebPart WebPartZoneID="Right" WebPartOrder="0">...
    </File>

    <File Url="default.aspx" Name="Page2.aspx" Type="Ghostable">
      <AllUsersWebPart WebPartZoneID="Left" WebPartOrder="0">...
      <AllUsersWebPart WebPartZoneID="Right" WebPartOrder="0">...
    </File>
  </Module>
</Modules>
```

## File and AllUsersWebPart

```
<File Url="default.aspx" Name="Page1.aspx" Type="Ghostable">
  <AllUsersWebPart WebPartZoneID="Left" WebPartOrder="0">
    <![CDATA[
      <WebPart xmlns="http://schemas.microsoft.com/WebPart/v2"
                xmlns:cewp="http://schemas.microsoft.com/WebPart/v2/ContentEditor">
        <Assembly>Microsoft.SharePoint, [full 4-part name] </Assembly>
        <TypeName>Microsoft.SharePoint.WebPartPages.ContentEditorWebPart</TypeName>
        <Title>Welcome to Page 1</Title>
        <FrameType>TitleBarOnly</FrameType>
        <cewp:Content>
          This is a page where you can just stare and watch your gears spin away
        </cewp:Content>
      </WebPart>
    ]]>
  </AllUsersWebPart>
</File>
```

## WEBTEMP\*.XML

```
<Templates xmlns:ows="Microsoft SharePoint" >
  <Template Name="HELLOSITEDEF" ID="10001">
    <Configuration ID="0"
      Title="The 'Hello world' siteDef"
      Hidden="FALSE"
      ImageUrl="/_layouts/images/TPG/HelloSiteDef_Preview.gif"
      Description="Use this site template for creating a fun new site."
      DisplayCategory="Litware"
      RootWebOnly="false"
      SubWebOnly="false" />
  </Template>
</Templates>
```

## Summary

- What is a site definition?
- Built-in site definitions with WSS and MOSS
- The Global Site Definition
- Creating a site definition



## Forms Services and InfoPath 2007

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

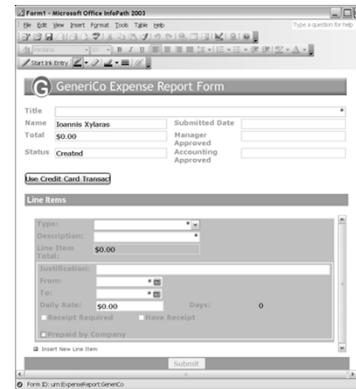


### Agenda

- 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



## Security – Trust Levels



### Restricted

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



### Domain

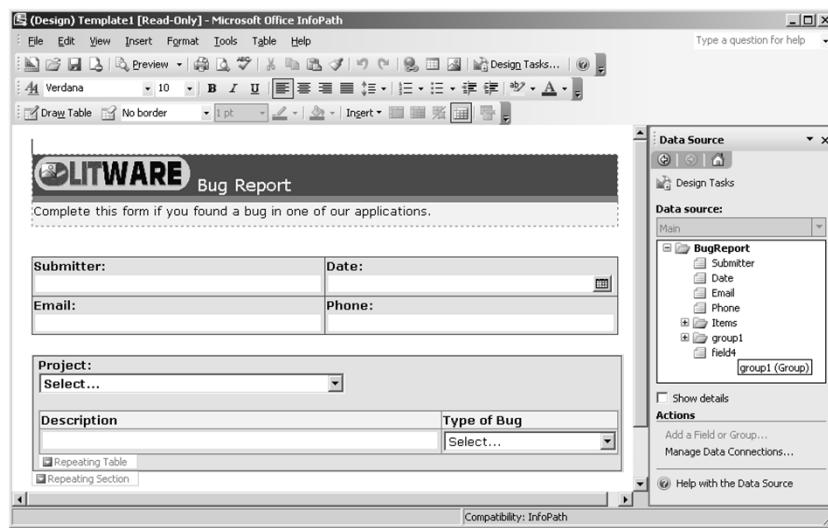
- Deployed to SharePoint library, browser forms
- Connect to own server only, no code for browser forms
- Use trusted Data Connection Library for cross-domain



### 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

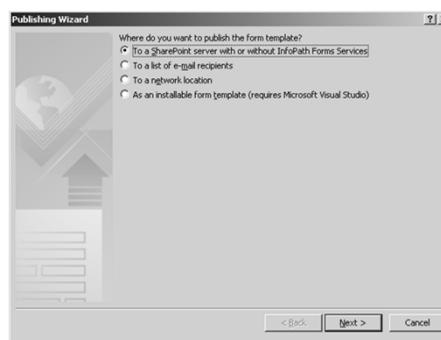
## Demo: The InfoPath Forms Designer



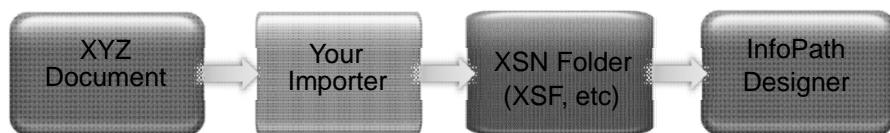
## InfoPath Integration with WSS

### ▪ 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

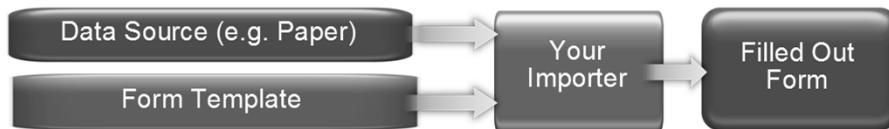


## Template Importing



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

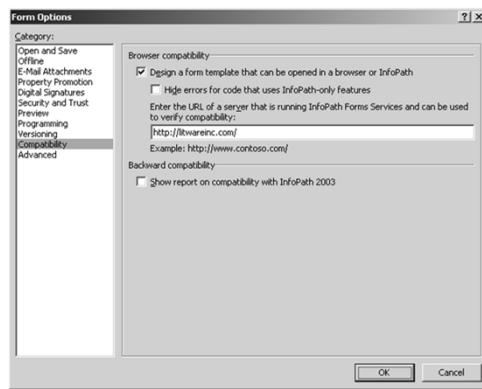
## Data Importing



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

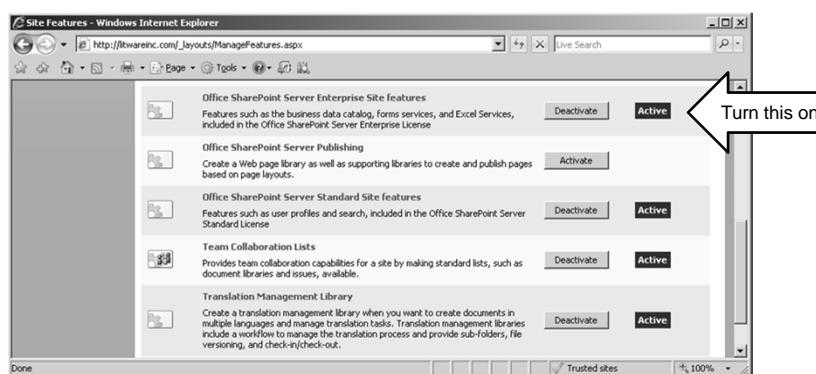
## Browser-based Forms

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

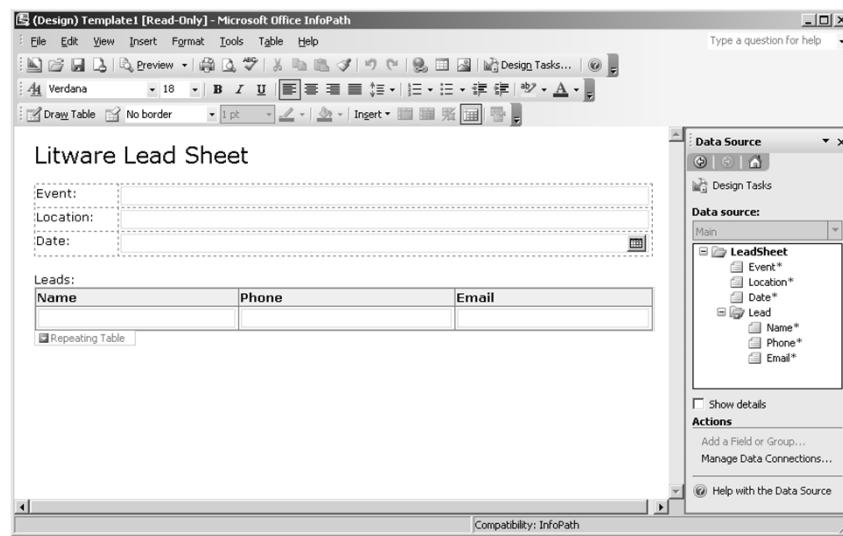


## Sites and Browser-based Publishing

- 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**

<b>Content Types</b> 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.	<b>Allow management of content types?</b> <input type="radio"/> Yes <input checked="" type="radio"/> No
<b>Document Template</b> 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.	<b>Template URL:</b> Leadsheets/Forms/template.xsn (Edit Template)
<b>Browser-enabled Documents</b> 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.	<b>Opening browser-enabled documents</b> <input type="radio"/> Open in the client application <input checked="" type="radio"/> Display as a Web page

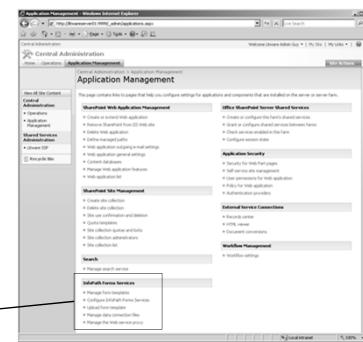
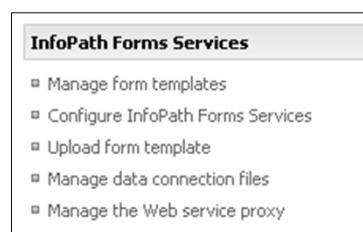
## Browser-based Rendering

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

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

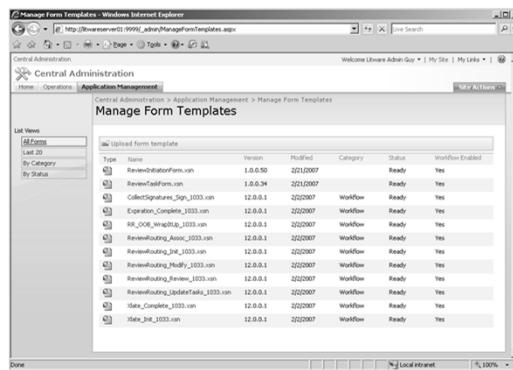
## Forms Services Administration

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



## 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

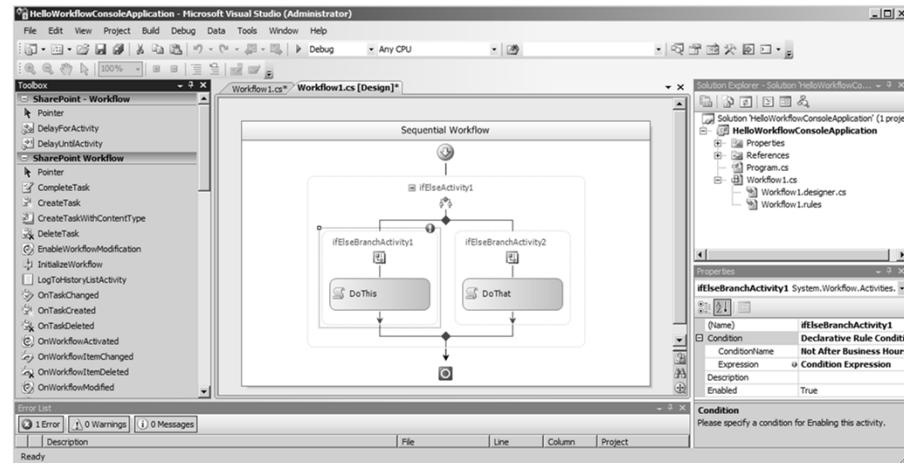
## Reactive Programs

- 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

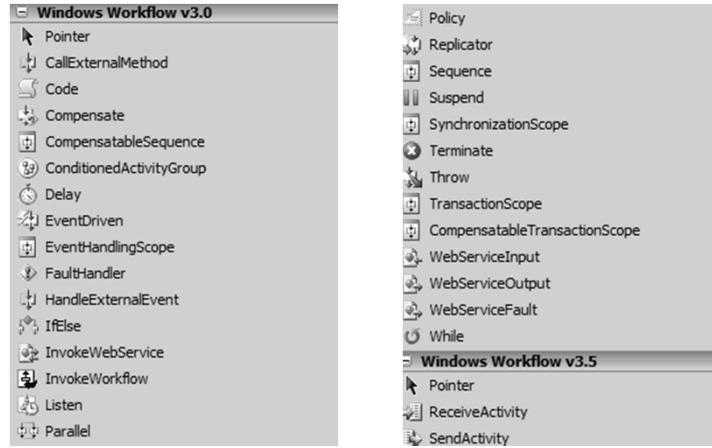


## Activities

- 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

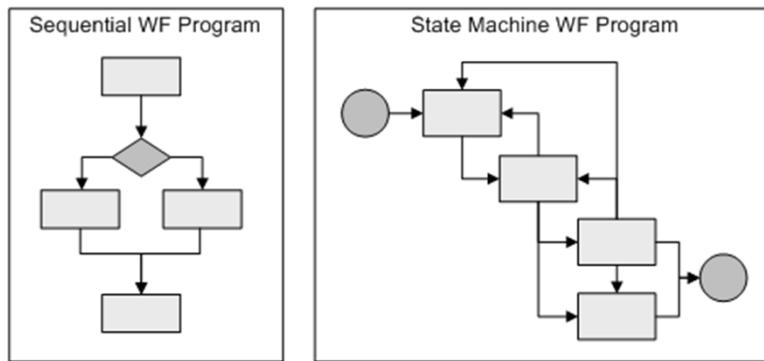


## Composite Activities

- 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;

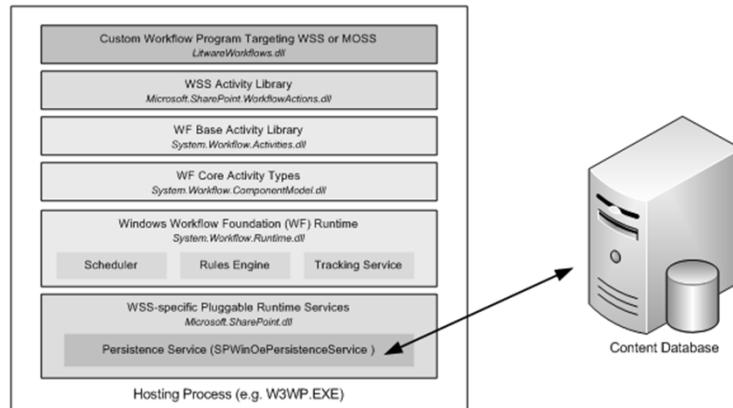
namespace HelloWorkflowConsoleApplication {
    class Program {
        static void Main() {
            // start WF runtime
            using(WorkflowRuntime workflowRuntime = new WorkflowRuntime()) {
                AutoResetEvent waitHandle = new AutoResetEvent(false);
                workflowRuntime.WorkflowCompleted +=
                    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

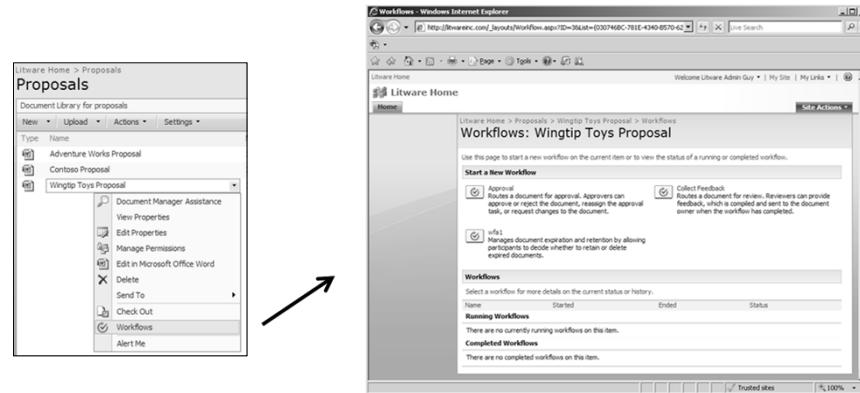
Use this page to set up a workflow for this document library.

<b>Workflow</b> 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.	<b>Select a workflow template:</b> Collect Feedback Collect Signatures Disposition Approval Three-state	<b>Description:</b> Use this workflow to track items in a list.
<b>Name</b> 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	
<b>Task List</b> 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.	<b>Select a task list:</b> Tasks	<b>Description:</b> Task list for workflow.
<b>History List</b> 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.	<b>Select a history list:</b> Workflow History	<b>Description:</b> History list for workflow.
<b>Start Options</b> 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**

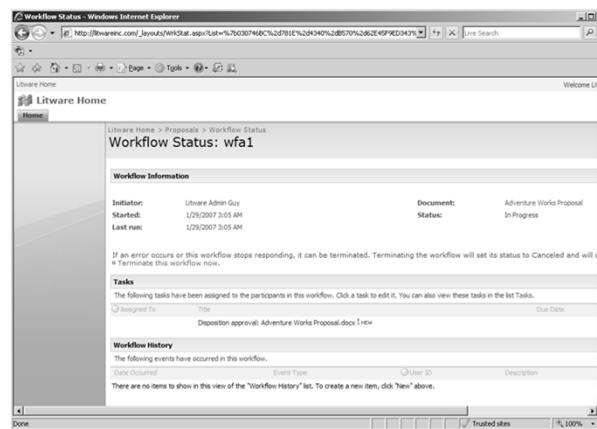
## Starting a Workflow Instance

- Users can manually start workflows



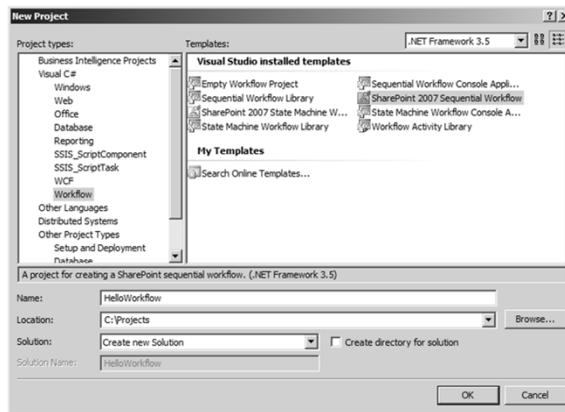
## The Workflow Status Page

- Any user can see the status of a workflow instance



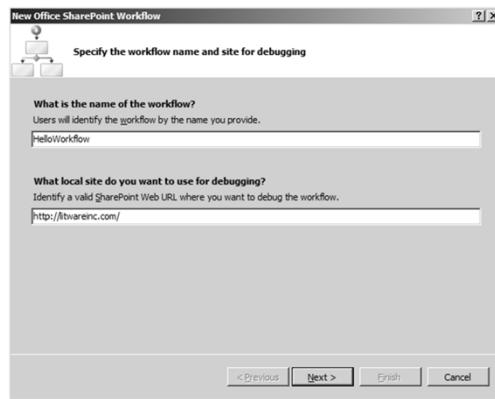
## Creating a Workflow Template Project

- Creating a SharePoint Workflow Project in Visual Studio 2008



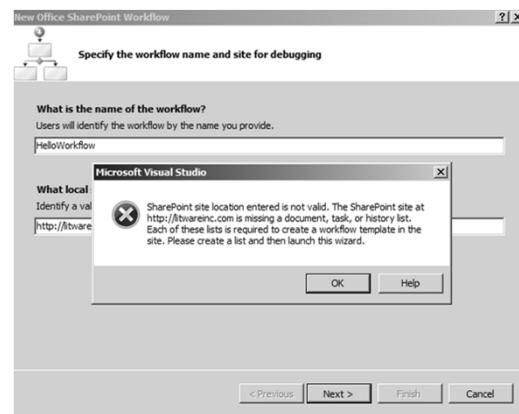
## Complete the Wizard

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



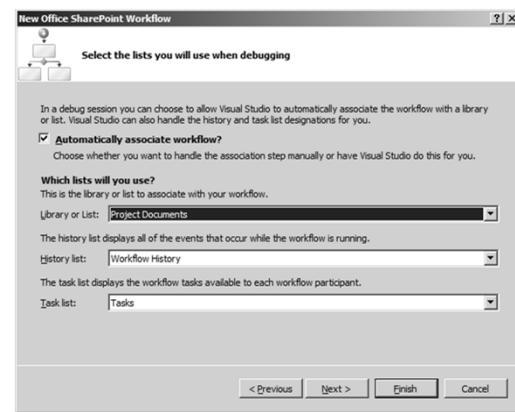
## Complete the Wizard

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



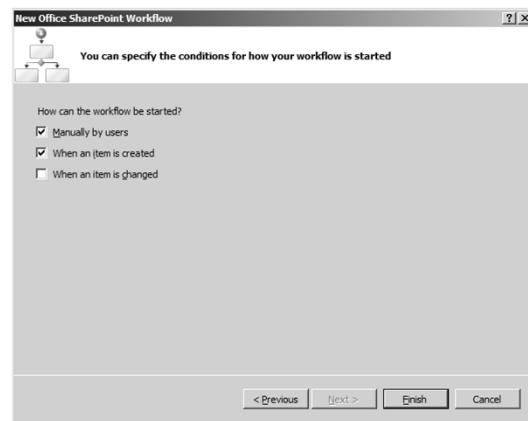
## Complete the Wizard

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



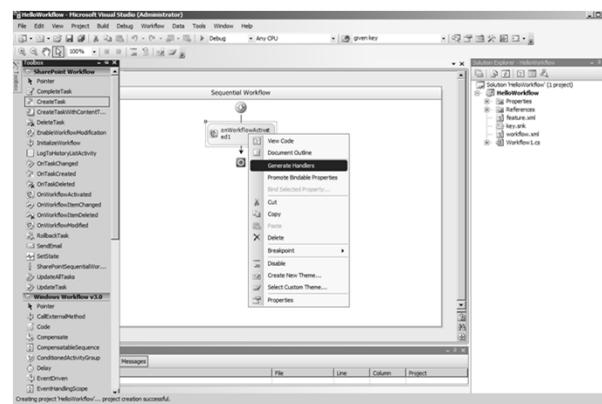
## Complete the Wizard

- 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

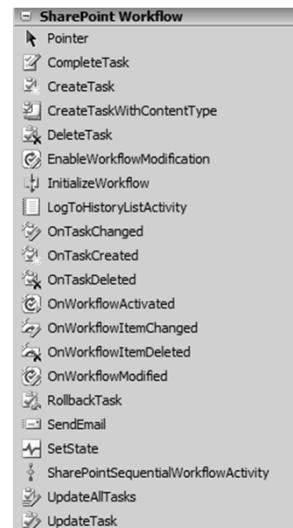
```
using Microsoft.SharePoint.Workflow;
namespace HelloWorkflow {
    public partial class Workflow1 : SharePointSequentialWorkflowActivity {
        // code to call wizard-generate code
        public Workflow1() {
            InitializeComponent();
        }

        // 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
    }
}
```

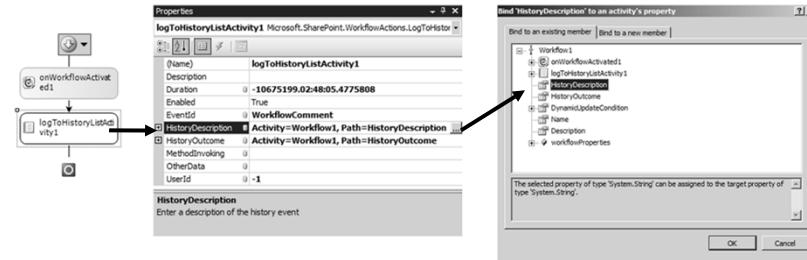
## SharePoint Activity Library

- 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

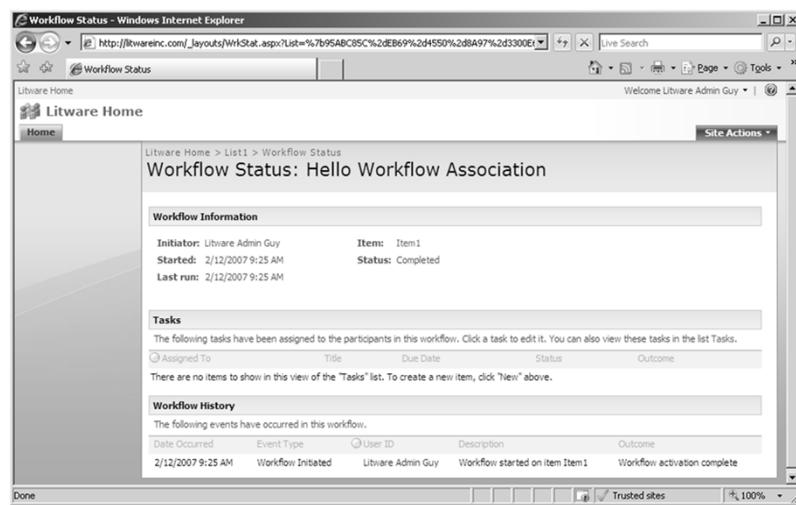
```
<Feature  
  Id="0CEED7AE-D327-41ad-BC33-B3F3F8A4DAD2"  
  Title="Hello World Workflow Template Feature"  
  Description="This feature installs our Hello World Workflow Template"  
  Version="12.0.0.0"  
  Scope="Site"  
  xmlns="http://schemas.microsoft.com/sharepoint/">  
  
  <ElementManifests>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

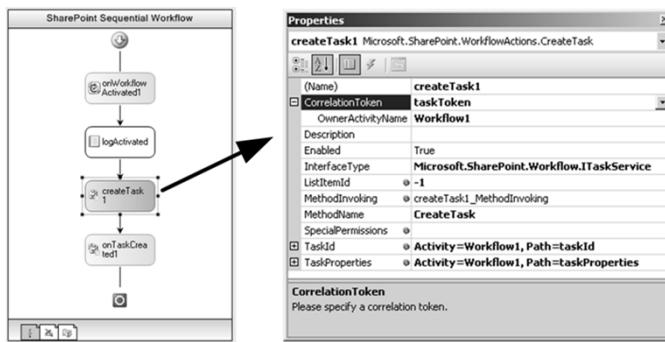


## Creating and Waiting on Tasks

- 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

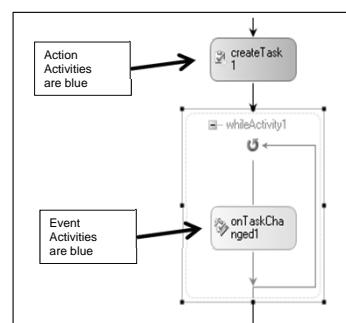
## Task GUIDs and Correlation Tokens

- 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

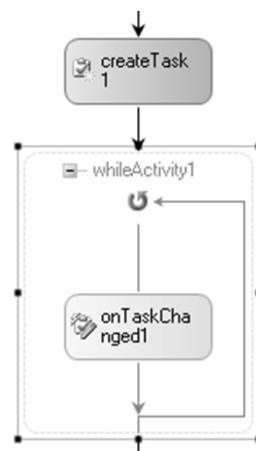
- Add event handler behind CreateTask activity
  - This event handlers fires before task creation
  - Gives you a chance to initialize task properties

```
// 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 xml ns="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>
```

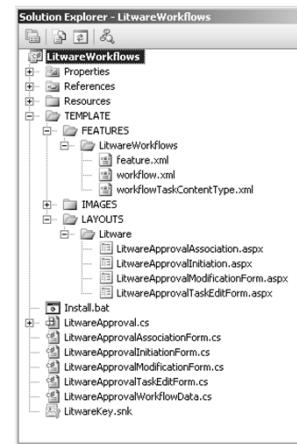
## Integrating Workflow Input Forms

### ▪ Workflow Input Form Types

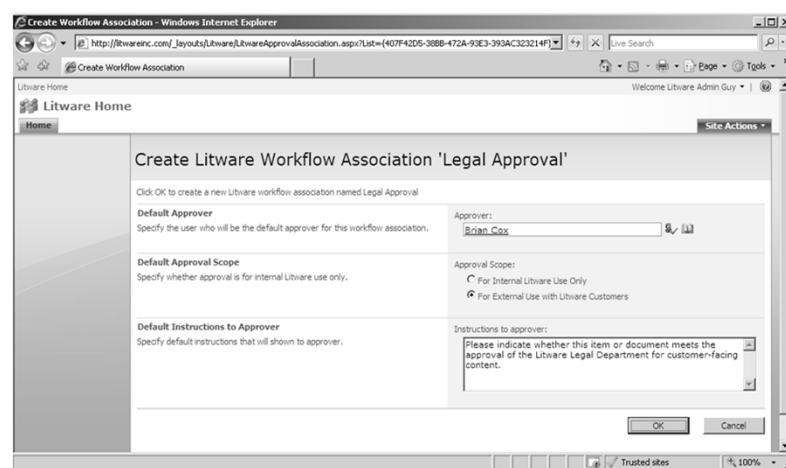
- Association Form
- Initiation Forms
- Modification Forms
- Task Edit Form

### ▪ Sample Project

- LitwareWorkflows



## The Association Form



## The Initiation Form

Start New 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 content. Make sure to check and then double check all copyright notices.

OK Cancel

## Invoking the Modification Form

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

There is one link per modification

## The Task Edit Form

The screenshot shows a SharePoint task edit form titled "Approve or Reject Item1". The page header includes "Litware Home" and "Welcome Brian Cox". The main content area has several sections: "Item Requiring Approval" (with links to "Item1", "List1", and "Site: http://litwareinc.com"), "Instructions to Approver" (with a note about legal approval), and "Approvers Comments" (with a text area 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"
    CodeBaseAssembly="LitwareWorkflow1InfoPath.Workflow1"
    CodeBaseAssemblyName="full assembly name"
    TaskListContentTypeId="0x01080100C9C9515DE4F24001905074F980F93160"
    AssociationUrl="_layouts/CstWrkflIP.aspx"
    InstantiationUrl="_layouts/IniWrkflIP.aspx"
    ModificationUrl=_layouts/ModWrkflIP.aspx">

    <MetaData>
      <Instantiation_FormURN>[INFOPATH FORM ID]</Instantiation_FormURN>
      <Association_FormURN>[INFOPATH FORM ID]</Association_FormURN>
      <Task0_FormURN>[INFOPATH FORM ID]</Task0_FormURN>
    </MetaData>

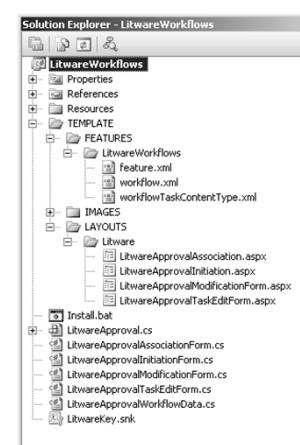
    <Categories/>

  </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



## Agenda

- Collaboration Portals
- Shared Service Provider (SSP) Architecture
- Creating corporate portal sites
- User Profiles
- Audience Targeting
- MySites
- Extending Search

## SharePoint Portal Server 2003 (SPS)

- SharePoint Portal Server 2003 Features
  - Areas and listings
  - User profiles and audience targeting
  - Search
  - My Sites
  - Shared Services
  - Single sign on

## Problems with SPS 2003

- Problems with SharePoint Portal Server 2003
  - A strange semi-undocumented layer on top of WSS V2
  - Limited to one SPS portal site per IIS Web application
  - Portal user interface hard to customize and extend
  - Shared Services architecture hard to deploy and administrate
  - Limited and clunky integration with Content Management Server

## Collaboration Portal Site Template

- Used to create SPS-like Portal sites
  - Supplies same basic features as SPS
  - Except that Areas and Listing have been eliminated



## Collaboration Portal Architecture

- A collaboration portal is a hierarchy of WSS sites
  - Created as a WSS site collection
  - Created using standard Site Definitions and Features



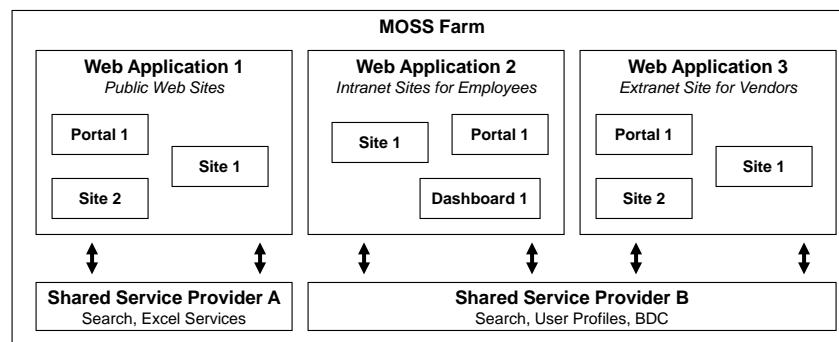
- Architectural improvements over SPS portals
  - Not limited to one portal per IIS Web site
  - Portals extended using standard WSS V3 development

## Sharing Resources and Services

- Need to share resources/services across sites
  - Content crawling and index creation
  - Querying for search results
  - Allocating and managing storage for My Sites
- SPS used 1st generation shared services model
  - One portal site is selected as "Master" portal
  - Other portals use shared services of master portal
- MOSS takes a different and better approach
  - Enter the Shared Service Provider (SSP)

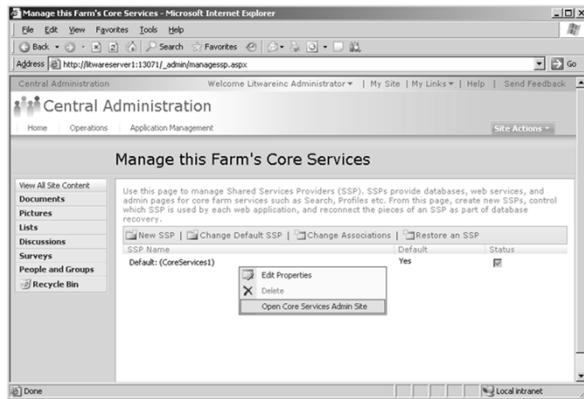
## SSP Architecture

- Shared Service Provider (SSP)
  - Configured independently of any portal or site
  - SSP resources/services accessible to all types of sites
  - Each Web Application maps to exactly one SSP



## Creating An SSP

- Central Administration -> Application Management -> Create or Configure this Farm's Core Services



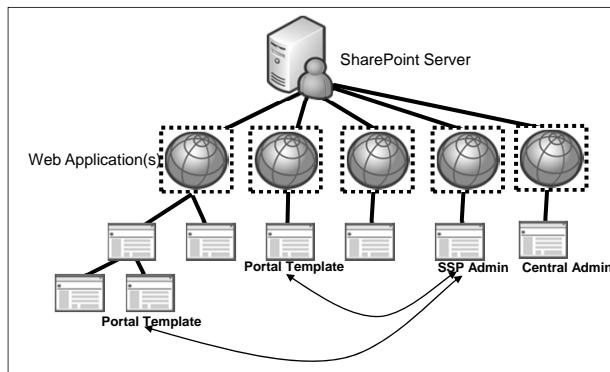
## SSP Administration

- Functionality of SSP Broken out into Sections

The screenshot shows the 'Shared Services Administration: Litware SSP' page in Microsoft Internet Explorer. The URL is <http://lwserver01:13099/ssp/admin/default.aspx>. The page title is 'Shared Services Administration: Litware SSP'. A callout box points to the 'Business Data Catalog' section, which is labeled 'MOSS Enterprise Edition Only'.

## So what is a Portal, Really?

- A Site Collection that...
  - has the correct features activated
  - is consuming services from an underlying SSP
  - might or might not have been created from a portal site template



## User Profiles

- User profiles track user information for social networking
  - Information used to describe users to each other
  - Information users to target information to interested parties
- MOSS maintains user profiles in SQL Server
  - Profile data can be imported from Active Directory or LDAP
  - Profile data can be extended with custom properties
  - Profiles can be extended with BDC data source
- What MOSS features rely on user profiles?
  - My Site infrastructure
  - Audience targeting
  - People Search

## People and Personalization

- Custom components can access...
  - Privacy policies
  - Audience targeting
- User profile store: reading/writing:
  - Users
  - Profiles
  - Memberships
  - Colleagues
  - Audiences

## Configuring User Profiles

**Administrative control for adding, viewing, managing and importing profiles**

**You can see and modify user profile properties**

**You can add application-specific properties**

The screenshot shows the 'Manage Profile Database' page. On the left, there are three callout boxes with the following text:

- Administrative control for adding, viewing, managing and importing profiles**
- You can see and modify user profile properties**
- You can add application-specific properties**

The main content area has two sections:

- Profile and Import Settings**: This section displays various status and configuration details:
  - Number of user profiles: 7
  - Source of user set: Current domain (litwareinc)
  - Import status: Enumerating
  - Membership Import status: Idle
  - Import time: Started full import at 12/13/2005 4:50 PM
  - Import schedule (full): Disabled
  - Import schedule (incremental): Disabled
  - Last log entry: No data source is defined.
  - Last import errors: 0A list of links is provided for managing profiles:
  - Refresh
  - Add user profile
  - View user profiles
  - Configure profile import
  - Stop import
  - Manage connections
  - View import log
- User Profile Properties**: This section shows the number of user profile properties and mapped properties:
  - Number of user profile properties: 42
  - Number of properties mapped: 18A list of links is provided for managing properties:
  - Add profile property
  - View profile properties

## Viewing User Profiles

- User profiles can be added and deleted

Account name	Preferred name	E-mail address
LITWAREINC\Administrator	Litwareinc Administrator	Administrator@litwareinc.com
LITWAREINC\Bart	Bart Wessels	Bart@litwareinc.com
LITWAREINC\Carol	Carol Phillips	Carol@litwareinc.com
LitwareInc\Mike	Mike Fitzmaurice	Mike@litwareinc.com
LITWAREINC\Sean	Sean Purcell	Sean@litwareinc.com
LITWAREINC\tedp	Ted Pattison	TedP@litwareinc.com
LITWAREINC\Tim	Tim Litton	Tim@litwareinc.com

## Creating a new My Site

- Each My Sites is a site collection that...
  - is provided on demand upon first access
  - maps to a specific user profile
  - enables users to edit some aspects of their profile
  - Provides a private aspect and a publicly-facing aspect

The screenshot shows a Microsoft Internet Explorer window displaying a SharePoint site. The address bar shows the URL: http://litwareinc.com/default.aspx. The navigation bar includes links for 'Home', 'My Site', and 'My Links'. The main content area features the Windows SharePoint Services logo. On the left, there is a vertical navigation menu with items like 'View All Site Content', 'Documents', 'Lists', 'Discussions', 'Sites', 'People and Groups', and 'Recycle Bin'.

## Updating A User Profile

Brian Cox - Windows Internet Explorer  
http://ltwareinc.com:81/\_layouts/Person.aspx?accountname=LITWAREINC\BrianC

**Brian Cox**

Solutions Developer, IT  
N1003 (429) 555-0129

I love biking and designing SharePoint sites.

**Details**  
Responsibilities: Sales, Marketing

**Contact Information**  
Work e-mail: BrianC@ltwareinc.com

**Organization Hierarchy**  
Luis Bonifaz, Technology Solutions Manager  
- Brian Cox, Solutions Developer  
- David Yalovsky, Solutions Developer  
- Sandeep Katyal, Solutions Developer

**Documents**  
Find all documents by Brian Cox

Type	Name	Last Modified	Location	Properties
File	Steve	8/22/2007 4:24 PM	Profile Pictures	[Edit]

**Colleagues**

**General**

- David Yalovsky
- Luis Bonifaz
- Sandeep Katyal

[Add Colleagues] [Manage Colleagues]

## Colleagues

My Colleagues - Windows Internet Explorer  
http://ltwareinc.com:81/\_layouts/MyContactLinks.aspx

My Site > My Colleagues

**My Colleagues**

Add Colleagues | Edit Colleagues | Delete | Group By: Group Name

Name	Show To	My Workgroup	Group Name
David Yalovsky	Everyone	Yes	General
Luis Bonifaz	Everyone	Yes	General
Sandeep Katyal	Everyone	Yes	General

## Audiences

- Audience involves creating rules and then compiling
  - Rules define what user accounts should be included or excluded
  - Compilation process adds users to audience table in SQL Server

**Audience Settings**

Name:	Sales Folks
Description:	
Compiled:	Yes
Number of members:	3
Membership:	Members satisfy all of the rules
Last compilation:	12/13/2005 5:03 PM
Compilation Errors:	No error

**Audience Rules**

Operand	Operator	Value
Department	Contains	Sales

**Add rule**

## Compiled Audience

- Compiled Audience defines list of users

**View Audience Membership: Sales Folks**

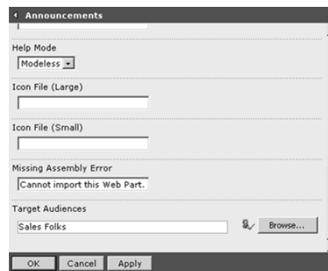
Use this page to view the current members of the audience.

Last compiled: 12/13/2005 5:03 PM  
Number of members: 3

Account Name	Preferred Name	E-Mail Address
LITWAREINC\Andrew	Andrew Dixon	Andrew@litwareinc.com
LITWAREINC\Sean	Sean Purcell	Sean@litwareinc.com
LITWAREINC\Tim	Tim Litton	Tim@litwareinc.com

## Audience Targeting

- Web Part output can be targeted at audience
  - Web Part content shown to members of that audience
  - Web Part content not shown to users not in audience
  - Great way to get content to interested parties while filtering noise from those that are not interested

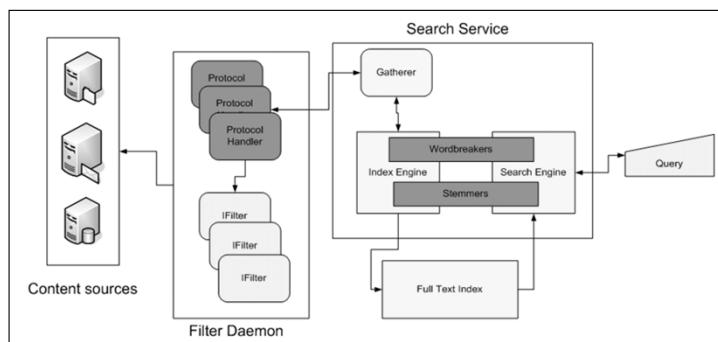


## WSS Verses Office Server Search

- Windows SharePoint Services (WSS 3.0)
  - WSS search is a subset of MOSS search feature
  - Eases transition from WSS to MOSS
  - Indexing and query always on the same machine
  - Search over site content only
- Microsoft Office SharePoint Server (MOSS)
  - Adds new search functionality over base WSS search
  - Indexer and query can be separated on different machines
  - Multiple catalog supported per indexer and query servers
  - Aggregated content - local + external

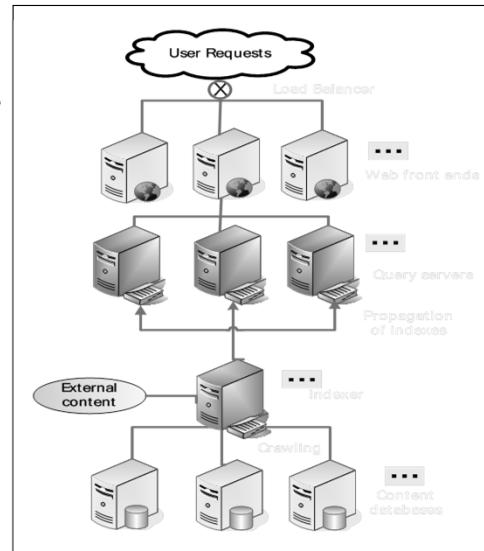
## Search Architecture and Terminology

- Key pieces to search infrastructure
  - The Gatherer
  - Content sources, protocol handlers and IFilters
  - Index Files



## Search Topologies

- Base topology
  - Web front-end servers
  - App servers
  - Database servers
- App servers
  - Indexer Service
  - Query Service



## Configuring Search

**Crawl Settings**

Indexing status:	Idle
Items in index:	1386
Errors in log:	1
Content sources:	1 defined (Local Office SharePoint Server sites)
Crawl rules:	0 defined
Default content access account:	Litwareinc\SP_WorkerProcess
Managed properties:	128 defined
Search alerts status:	Active
Propagation status:	Propagation not required

**Scopes**

## Search Scopes

- Created at one of two different levels
  - Can be created within content of an SSP
  - Can be created within context of a site collection

**View Scopes**

Title	Update Status	Items
Shared (3)		
People	Ready	84
All Sites	Ready	1280
My New Search Scope (SSP Level)	Empty - Add rules	empty
http://itwareserver01:9998/ssp/admin/ (0)		
http://itwareinc.com/sites/LitwarePublishingSite (0)		
http://itwareinc.com:81/ (0)		
http://itwareinc.com:81/personal/brianc (0)		
http://itwareinc.com:81/personal/administrator (0)		
http://itwareinc.com/sites/LitwarePortaSite (0)		

## Adding Rules to a Search Scope

- Each search scope has one or more rules
- Rules define criteria to include/exclude content

Shared Services Administration: Litware SSP > Search Settings > Scopes > Scope Properties and Rules > Add Scope Rule

### Add Scope Rule

<b>Scope Rule Type</b>	<input type="radio"/> Web Address ( <a href="http://server/site">http://server/site</a> ) <input checked="" type="radio"/> Property Query (Author = John Doe) <input type="radio"/> Content Source <input type="radio"/> All Content
<b>Property Query</b>	Enter the property restriction as a comparison of property to a value. All items matching the property query will be included in the scope. To manage additional properties available for use in scopes, navigate to the managed properties list and select "Allow this property to be used in scopes" for the desired managed properties. Add property restrictions: Author = Brian Cox Example: Author = John Doe
<b>Behavior</b>	<input type="radio"/> Include - Any item that matches this rule will be included, unless the item is excluded by another rule <input checked="" type="radio"/> Require - Every item in the scope must match this rule <input type="radio"/> Exclude - Items matching this rule will be excluded from the scope
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

## Search Administration

Home Site Actions

### Search Administration

**Administration**

- Search Administration
- Central Administration
- Crawling**
- Content sources
- Crawl rules
- Crawl log
- Default content access account
- File types
- Reset all crawled content
- Crawler impact rules
- Proxy and timeouts

**Queries and Results**

- Authoritative pages
- Federated Locations
- Metadata properties
- Scope
- Server name mappings
- Search result removal

**Usage Reports**

- Queries report
- Results report

**System Status**

Crawl status	Crawling - Local Office SharePoint Server sites			
Items in index	0			
Server status	LITWARESERVER : C: 10GB			
Propagation status	Propagation not required			
Default content access account	litwareinc\sys-spSearch			
Contact e-mail address	administrator@litwareinc.com			
Proxy server	none			
Scopes update status	Idle			
Scopes update schedule	Automatically scheduled			
Scopes needing update	0			
Search alerts status	On Disable			
Query logging	On Disable			

**Active crawls**

Content Source	Status	Duration	Success	Error
Local Office SharePoint Server sites	Crawling Full	00:00:26	0	0

**Recently completed crawls**

Content Source	Type	Finished	Duration	Success	Error
There are no completed crawls.					

**Shortcuts**

There are currently no favorite links to display. To add a new link, click "Add new link" below.

- Add new link

**I Want to ...**

- Add new content to crawl
- Federate results from other locations
- Add users to search
- Customize the Search Center
- Secure the search system
- Enhance the user experience
- Read online documentation

## Search Center

- Customize and extend Search Center
  - Modify query parameters
  - Add tabs
  - Modify XSLT, CSS on results
  - Custom search Web Parts

## Search Web Parts

- 9 OOB web parts including
  - Search Box
  - Core Results
  - High Confidence
  - Statistics
  - Pagination
  - Action Links
  - Matching Keywords and Best Bets
  - Search Summary (Did you mean?)
  - Advanced Search
  - Share data through hidden object
- Web Part properties such as
  - Formatting
  - From turning stemming on/off to the # of results returned
  - XSL

## Search As An Extensible Service

- Run search as a centralized enterprise service
  - Use in SharePoint sites and custom apps
  - Via shared services or query web service
- New tools and frameworks
  - Coherent, comprehensive admin APIs
  - New query objects and syntax
  - Reusable, customizable UI controls
  - Index custom repositories

## Summary

- Collaboration Portals
- Shared Service Provider (SSP) Architecture
- Creating corporate portal sites
- User Profiles
- Audience Targeting
- MySites
- Extending Search



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.

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

**Critical Path TRAINING**

## Agenda

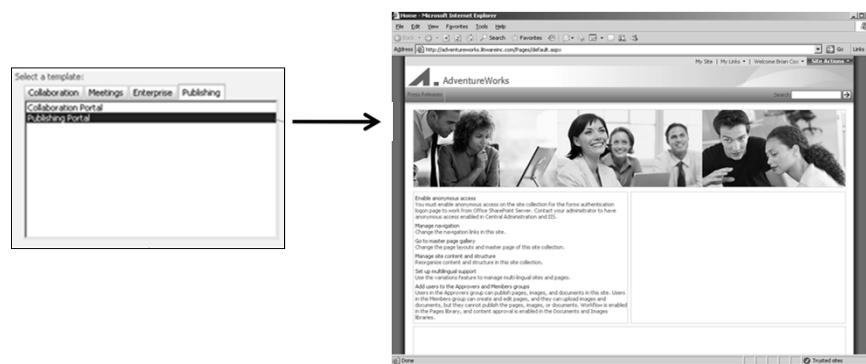
- The Publishing Site template
- The MOSS Approval Process
- Creating custom page layouts
- Converting Office documents
- Content Translation using Variations
- Optimization through Caching Profiles

## 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

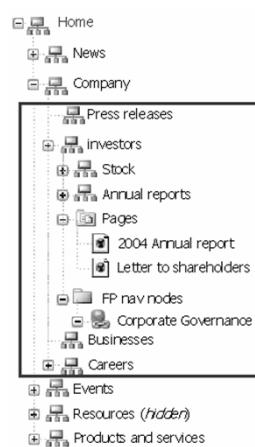


## Site Hierarchy

- 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

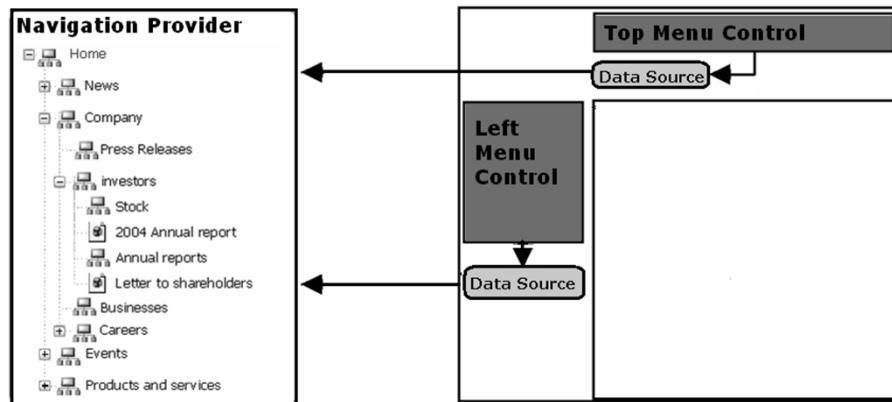


## Navigation



- 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

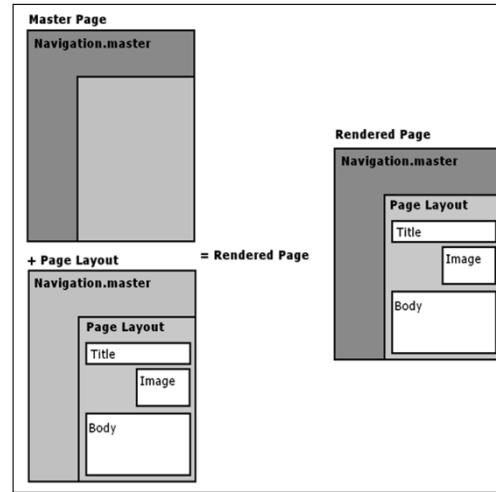
## Site Content and Structure

The screenshot shows the Site Content and Structure interface for the Litware Internet Site. The left pane displays a tree view of site content, including sections like About Litware, Case Studies, Job Postings, Press Releases, Master Page Gallery, and Style Library. The right pane shows a list of pages under 'Pages - All Documents' with columns for Type, Name, Modified, Modified By, Approval Status, Scheduling Start Date, Scheduling End Date, Contact, and Page Layout. One page, 'default.aspx', is selected. Below the main list, there is a section titled 'Resources Related To 'default.aspx'' showing two items: 'WelcomeLinks.aspx' and 'PR.gif'.

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

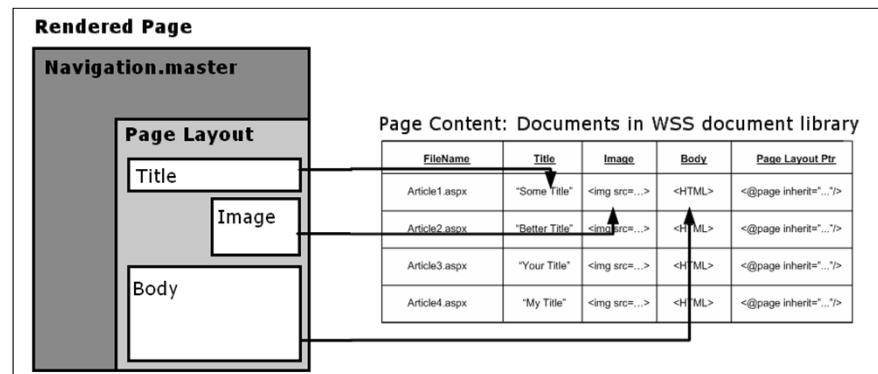


## Page Layouts

- 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

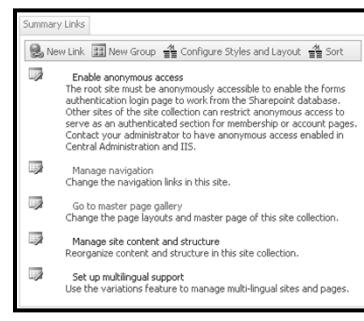
- Create site columns
- Create content type
- Add created site columns to content type
- In the Master Page Gallery
  - Create new Page Layout file
  - Check-out file and edit in SharePoint Designer
  - Populate the file with content fields
  - Check-in and approve
- Use the new page layout file

## Publishing Cycle

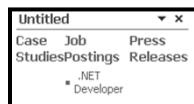
- Workflow based on Windows Workflow Foundation
- Light-weight approval workflow is active OOB
  - Based on approval
  - Minor versions need to be approved to become major versions
  - Visitors only see the major (published) versions
- Workflow can be replaced by custom workflow
  - OOB delivered with 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



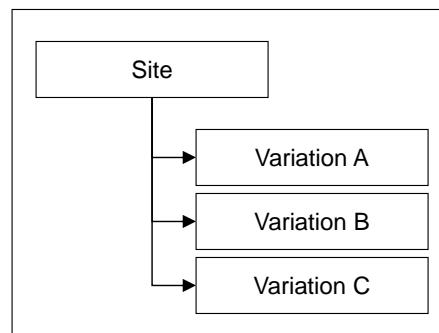
- Content Query Web Part
  - Displays a dynamic view of the content in your site

## 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.

**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**  
 Enable output cache

**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:**  
 Publishing sites can use a different page output cache profile

**Page Layouts:**  
 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.

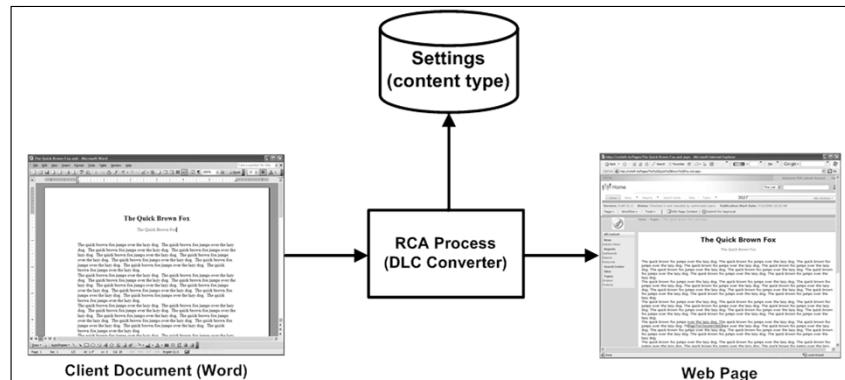
Service	Description	Status	Action
Central Administration	All services run on this server	Started	Stop
Document Conversions Launcher Service	Web Application and Search Query services run on this server	Stopped	Start
Document Conversions Load Balancer Service	Search Indexing service runs on this server	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		Started	Stop
Windows SharePoint Services Incoming E-Mail		Started	Stop
Windows SharePoint Services Search Service		Started	Stop
Windows SharePoint Services Web Application		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



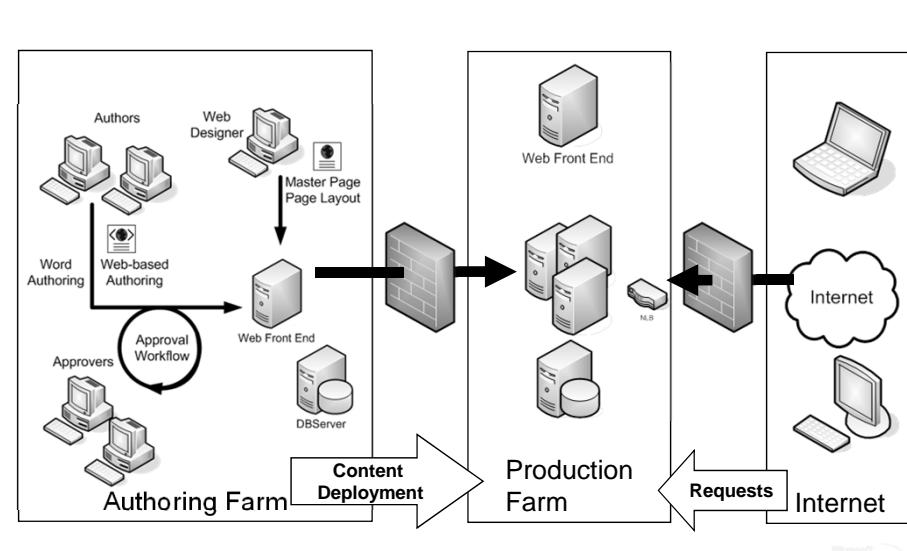
## Smart Client Authoring



- 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

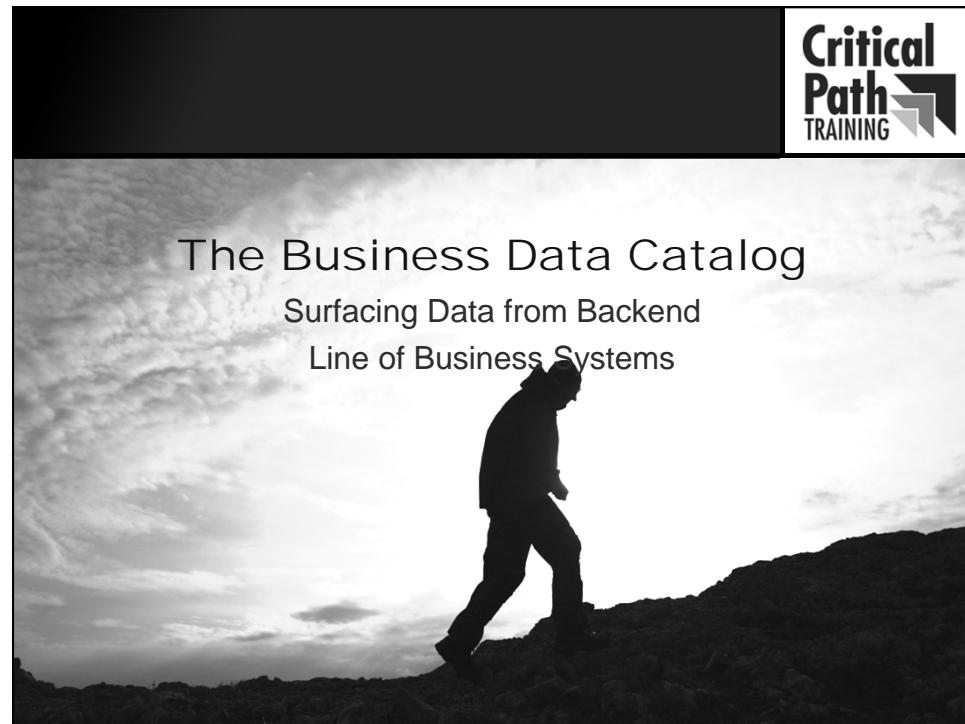
## Content Deployment

- 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

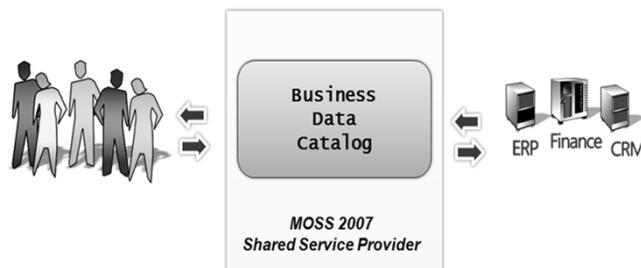


## Agenda

- 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

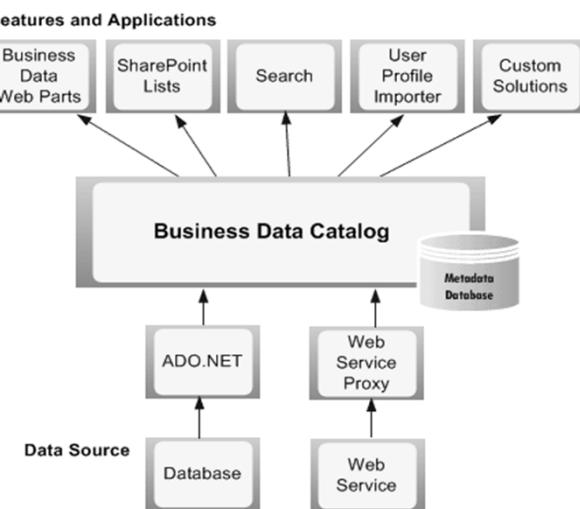
## Why do we need the BDC?

- 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">SqlServer</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: Litware SSP' interface. The page has a left navigation bar with links like 'View All Site Content', 'Back to Central Administration', 'Shared Services Administration', and 'Litware SSP'. The main content area is titled 'Import Application Definition' and contains the following sections:

- Application Definition:** A brief description of what an application definition is and how it's used.
- Application Definition File:** A file input field containing the path 'F:\emos\l1\_BDC\BDC\_AppDef\AdventureWorks\_HR.xml' with a 'Browse...' button.
- File Type:** A section where users can choose the type of application definition file to import, with options for 'Model' (selected) and 'Resource'.
- Resources to Import:** A section where users can choose resources to import, with checkboxes for 'Localized Names' (checked), 'Properties' (checked), and 'Permissions' (unchecked).
- Buttons:** At the bottom right 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 arrow 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: checked), 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, 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).

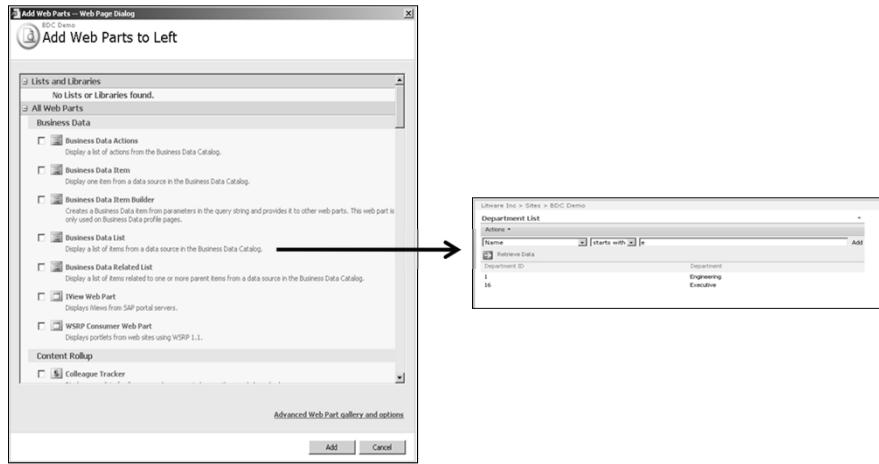
## Adding Actions to an Entity

The screenshot shows the 'Add Action' dialog box. The 'Name' field contains 'Open data in custom page'. The 'URL' field shows a placeholder URL: '[http://example.com/edit.aspx?id={0}]'. The 'URL Parameters' section shows a parameter property for 'DepartmentID'. The 'Icon' section shows a standard edit icon. At the bottom are 'OK' and 'Cancel' buttons.

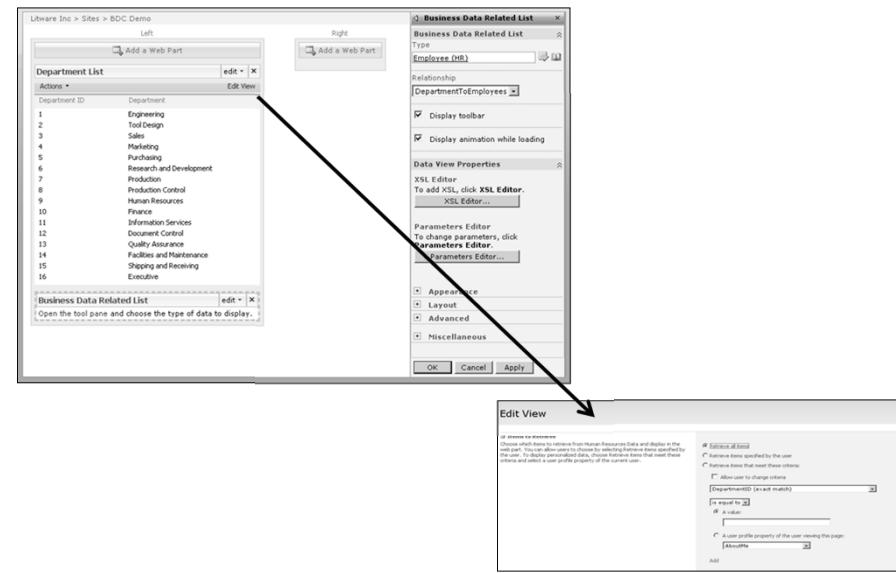
## Administrating Security

The screenshot shows the 'Manage Permissions: Department' page. It lists a single user/group entry: 'User/Group Name: LITWAREINC\administrator' and 'Rights: Edit, Execute, Selectable in clients, Set Permissions'. There 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, Sanya 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 list named 'Colleagues'. The 'Name and Type' section has the column name set to 'Department'. The 'Type' dropdown shows 'Department (BDC)'. The 'Additional Column Settings' section includes fields for 'Description' (empty), 'Require that this column contains information' (set to 'Yes'), 'Type' (set to 'Department (BDC)'), 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 columns for 'Department' and 'Department ID', and a checkbox for 'Add to default view'.

## 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



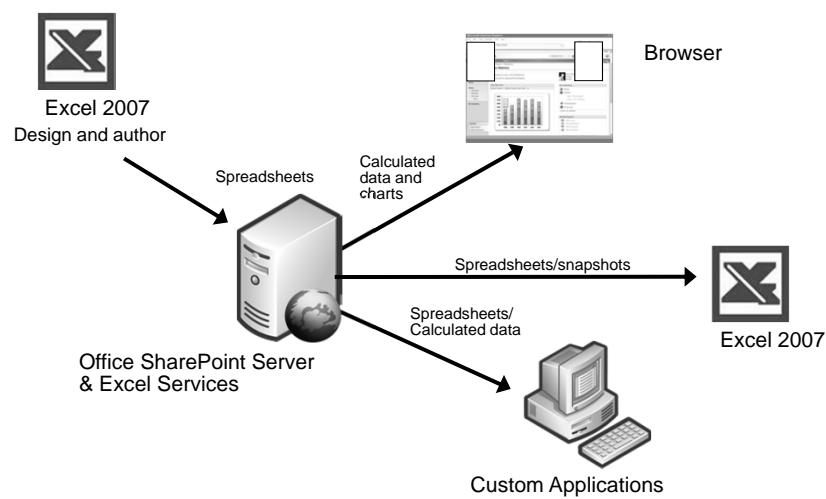
## Agenda

- Excel Services
  - Publishing spreadsheets that render in the browser
  - Configuring Trusted Locations
  - Connections
  - Using user-defined functions (UDFs)
- Report Center
  - Creating Dashboards
  - Key Performance Indicators (KPIs)
  - Filters

## Why Do We Need Excel Services?

- Common customer requirements/complains
  - Distributing spreadsheets to users creates many copies
  - Excel doesn't play well in the BI dashboard and reporting world
  - It's difficult to protect proprietary information in spreadsheets
  - Incorporating Excel logic into applications is hard
  - Excel was designed as a desktop application  
*(read: Excel really stinks as a server-side application)*

## Excel Services



## Walk Through of using Excel Services

1. Add the URL of the document library as trusted location
  1. SharePoint Central Administration
  2. In Application Management, configure the Farm's core services
  3. In Excel Services Management, add the URL of the doc lib as a trusted file location
2. In Excel 2007, publish to Office Server
  1. Decide what worksheets to publish
  2. Named ranges can be dynamically populated with values in the browser
  3. Give the URL of a document library
3. Users now have Web access to the spreadsheet

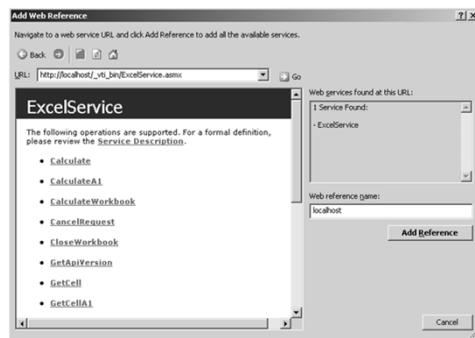


## Excel 2007 Web Services

- Using server-side Excel logic in applications
  - Author part of the business logic in Excel
  - Protect and maintain proprietary information
- Automating spreadsheet updates on servers
  - Refresh external data and parameterize
  - Process generated spreadsheets
  - Create, store and deliver snapshots
- Custom UI to server-side Excel calculation

## Excel 2007 Web Services

- Follow the previous steps to publish a spreadsheet to Office Server
  - Add Web Reference to your .NET application



## Excel 2007 Web Services

```
//-- create instance of proxy and take care of authentication
ExcelService ws = new ExcelService();
ws.Credentials = System.Net.CredentialCache.DefaultCredentials;

//-- open the Excel workbook
Status[] status = null;
string sessionId = null;
sessionId = ws.OpenWorkbook(textBoxURL.Text, "en-US", "en-US", out status);

//-- set the cell values
status = ws.SetCellA1(sessionId, "Mortgage Calculator", "CustomerName", textBoxCustomerName.Text);
status = ws.SetCellA1(sessionId, "Mortgage Calculator", "MortgageAmount", textBoxAmount.Text);
status = ws.SetCellA1(sessionId, "Mortgage Calculator", "InterestRate", textBoxInterestRate.Text);
status = ws.SetCellA1(sessionId, "Mortgage Calculator", "MortgageLength", textBoxLength.Text);

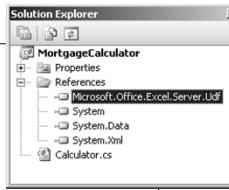
//-- calculate the workbook and get result
status = ws.CalculateWorkbook(sessionId, CalculateType.Recalculate);
object result = null;
result = ws.GetCellA1(sessionId, "Mortgage Calculator", "Payment", true, out status);

//-- display result and close workbook
if (result != null)
    MessageBox.Show("You pay " + result.ToString());
status = ws.CloseWorkbook(sessionId);
```

## Using UDFs with Excel 2007 Services

### UDF definition

- Methods in .NET classes, callable from Excel formulas



```
using System;
using Microsoft.Office.Excel.Server.Udf;

namespace MortgageCalculator {
    [UdfClass]
    public class calculator {
        [UdfMethod]
        public double calculateMortgage(int salesPrice, int mortgageLength,
                                         double downPaymentPercentage,
                                         double annualInterestPercentage) {
            // implementation
            double financed = (1 - downPaymentPercentage) * salesPrice;
            int nrofMonths = mortgageLength * 12;
            double monthlyInterestRate = annualInterestPercentage / 12;
            return financed *
                (monthlyInterestRate /
                (1 - Math.Pow((1 + monthlyInterestRate), nrofMonths * -1)));
        }
    }
}
```

## Report Center

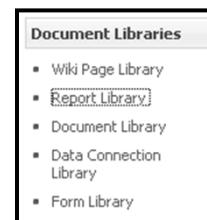
- With Office 2007, SharePoint Server become the hub for BI on the server
  - Excel Services and Reporting Services in the portal
  - Out-of-the-box BI portal experience
  - Dashboards, KPIs, and Report Libraries
  - Integrated with Portal, Collaboration, Enterprise Content Management and Workflow functionality

## SQL Server Reporting Services

- Integrated with WSS to enable publishing, viewing, and management of reports
- Microsoft Office SharePoint Server light up
  - Report library integration
  - Dashboards and filter Web Parts
- Integration is interesting because...
  - New capabilities for Reporting Services users
  - Great example of deep integration for ISVs

## Storing Reports

- Reports have more specific needs than documents
  - History is very important
  - Many instances of the same report can exist
- Therefore, Office Server adds a Report Library template
  - Displays current spreadsheet / report by default
  - History available via search and list views
  - Custom profile page
  - Can be part of Report Center
  - A list template that can be used anywhere



## Viewing Reports

- Works in both non-integrated mode (via IFRAME) as well as integrated mode
- Remote render in response though callbacks via the click handler
  - .RDL files are registered in SharePoint Services
- On rendering, callback to SharePoint Services object model to synchronize contents
- Consumes filter part to part connection for specifying report parameter values

Full Name	Sales Figures by Fiscal Year		
	2002	2003	2004
Michael Murphy	\$3,501,000.00	\$4,310,000.00	\$4,510,000.00
Lori Morris	\$1,000,000.00	\$1,400,000.00	\$1,200,000.00
Allen Davis	\$1,300,000.00	\$1,600,000.00	\$1,500,000.00
Osman T. Yigit	\$1,150,000.00	\$1,400,000.00	\$1,350,000.00
Tina Michael Fisher	\$1,000,000.00	\$1,200,000.00	\$1,100,000.00
Pamela O'Connor Whitt	\$1,400,000.00	\$1,600,000.00	\$1,500,000.00
Sara K. Lee	\$1,000,000.00	\$1,200,000.00	\$1,100,000.00
Joel Edwards Smith	\$1,300,000.00	\$1,400,000.00	\$1,350,000.00
David R. Campbell	\$1,200,000.00	\$1,300,000.00	\$1,250,000.00
Lee B. P. Jones	\$1,100,000.00	\$1,200,000.00	\$1,150,000.00
Randy W. Valley Chisholm	\$1,000,000.00	\$1,100,000.00	\$1,050,000.00
Tony A. Mease Asante	\$1,000,000.00	\$1,100,000.00	\$1,050,000.00
Rachel D. Valdez	\$1,000,000.00	\$1,100,000.00	\$1,050,000.00
Lynn M. Terfner	\$1,000,000.00	\$1,100,000.00	\$1,050,000.00

## Key Performance Indicators

- Goals
  - KPIs can be an important instrument in the organization, so let's make it easier to create them
  - And let's use the environment we already know – the portal and team sites
  - Support KPI types from simple to enterprise class
- Types
  - Manually entered
  - SharePoint list
  - Excel workbook
  - SQL Server Analysis Services
- Technologies
  - KPI web part & list
  - Customizable KPI profile page
  - Dashboard template focused on KPIs

## Dashboards

- Dashboards are SharePoint pages
- Dashboard pages are in same document library as spreadsheets and reports
- Types
  - Generic dashboard
  - KPI focused



## Filtering

- Filtering is the natural next step after building a dashboard
- Filter for eastern region, last quarter
- Automatically show just *your* customers when you load page
- Accept values from query string
  - <http://server/dashboard.aspx?Product=452>

## Filtering Web Parts

### Display options

- Type in value
- Pick from list
- Tree view
- Hidden

### Filter value sources

- User entered value
- Manual list
- SharePoint list
- Analysis Services
- Bus. Data Catalog
- SharePoint profile
- Query string

### Extensibility

- Custom providers & consumers
- Standard interfaces that ship in WSS

## Summary

### ▪ Excel Services

- Publishing spreadsheets that render in the browser
- Configuring Trusted Locations
- Connections
- Using user-defined functions (UDFs)

### ▪ Report Center

- Creating Dashboards
- Key Performance Indicators (KPIs)
- Filters



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 with their names, statuses, .NET Framework versions, managed pipeline modes, and identities. The 'SharePoint Default App Pool' is highlighted. To the right, an 'Advanced Settings' dialog box is open for this pool, showing configuration details like .NET Framework Version (v2.0), Managed Pipeline Mode (Classic), Name (SharePoint Default App Pool), Queue Length (1000), Start Automatically (True), CPU settings, Process Model settings (Identity set to 'ltwareinc\sys-spWorkerP'), and a note about the Name field.

## 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

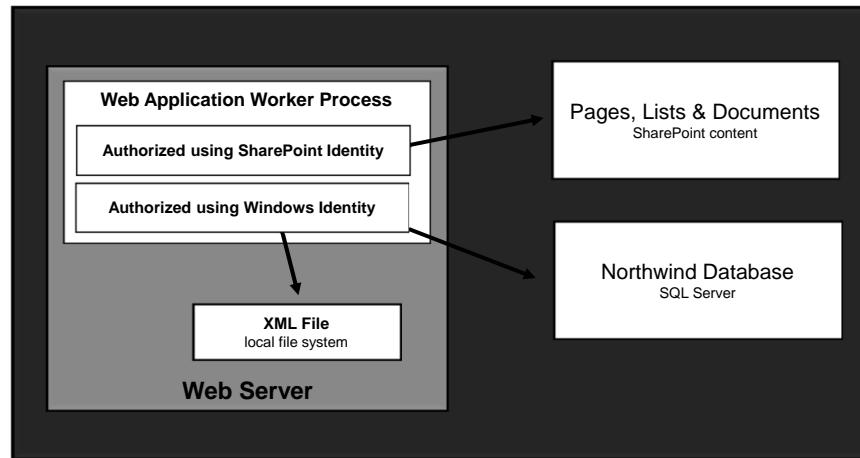
- What Identity is used to access SQL Server?
  - WSS system code accesses configuration database and content databases with Application Pool Identity
  - Custom code (e.g. Web Parts) run with impersonation and access SQL Server database with user identity

## 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;

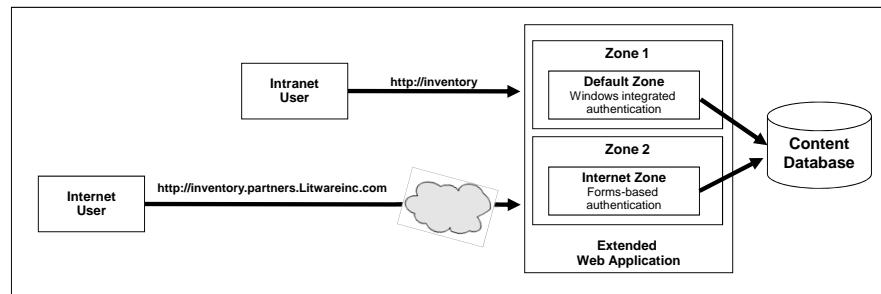
    SPSecurity.RunWithElevatedPrivileges(delegate() {
        // create new object while running with SYSTEM\SHAREPOINT identity
        using (SPSite ElevatedsiteCollection = new SPSite(siteCollection.ID)) {
            using (SPWeb Elevatedsite = ElevatedsiteCollection.OpenWeb(site.ID)) {
                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

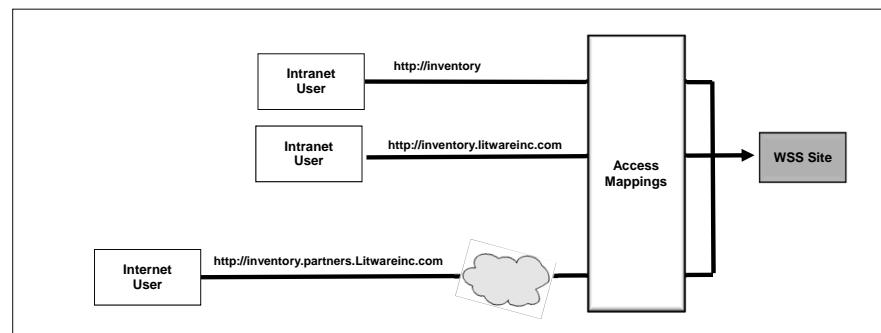
## Authentication And WSS Zones

- 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



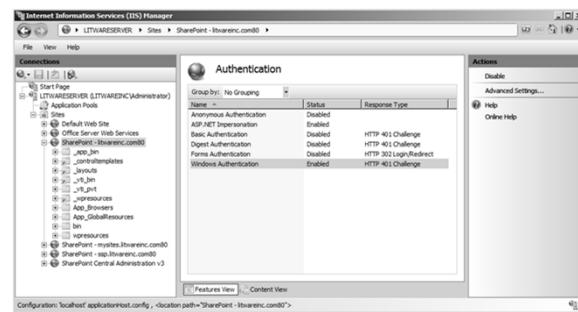
## Alternate Access Mapping

- 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



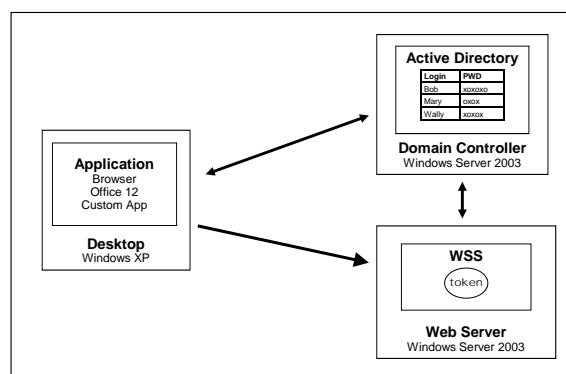
## Configuring Windows Authentication

- 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



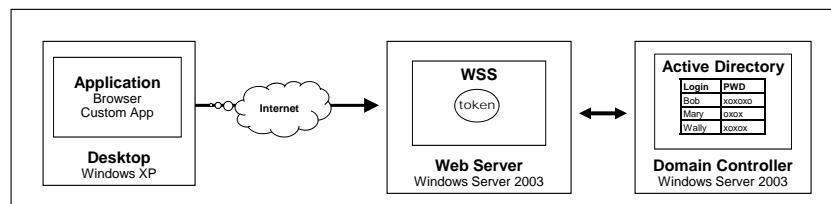
## Integrated Windows Authentication

- 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



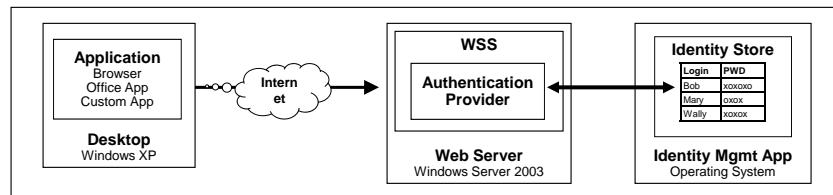
## Windows Authentication Zones

- **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



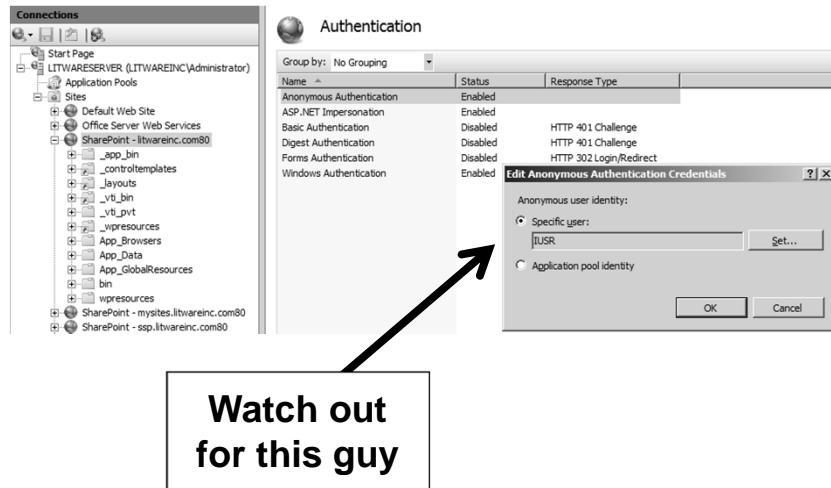
## Forms Authentication Zones

- 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 from a SharePoint site named 'Sales Site'. At the top, it says 'Site Settings > Permission Levels'. Below that, it says 'This Web site has unique permissions.' There are buttons for 'Add a Permission Level' and 'Delete Selected Permission Levels'. A table lists the following permission levels:

Permission Level	Description
<input 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 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	List Rights	Personal 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	<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 - Delete 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.	<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

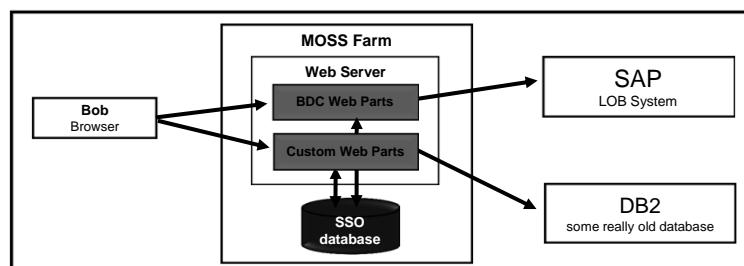
## Web Application Security Policy

- 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 checked. Other options like 'Full Control', 'Deny Write', and 'Deny All' are available but unchecked.

## 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.



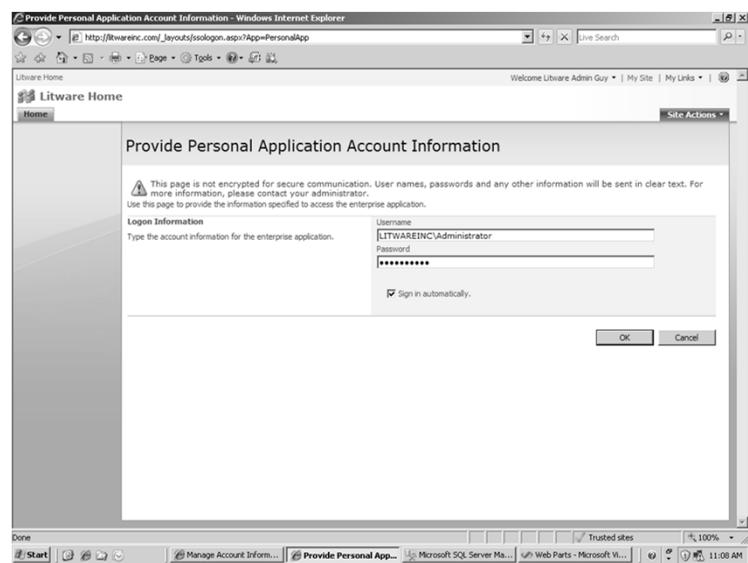
## Programming SSO

```
// 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
    }
}
```



## 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)