

# SPT401

## Inside SharePoint Products and Technologies



### Schedule of lectures

1. SharePoint 2007 Roadmap
2. Developing Features
3. SharePoint Architecture
4. Pages and Site Branding
5. Developing Web Parts
6. Lists and Content Types
7. InfoPath 2007 and Forms Services
8. SharePoint Workflows
9. Extending MOSS Portal and Search
10. Web Content Management
11. Business Data Catalog
12. Excel Services and Report Center
13. Application Security

Revision Date: September 24<sup>th</sup>, 2007





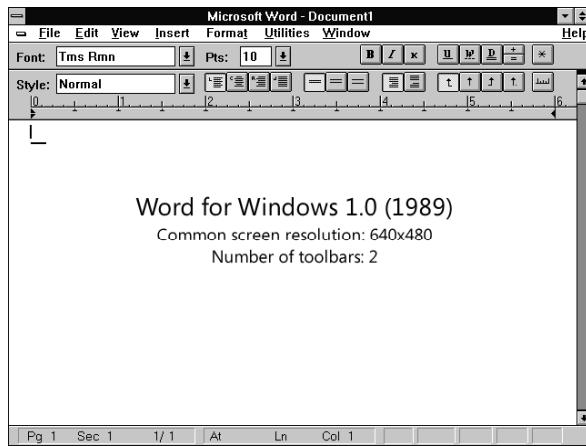
The slide features a dark background with a tiger in a forest at the top. The Ted Pattison Group logo is in the top left corner. The title 'SharePoint 2007 Developer Roadmap' is centered, followed by the subtitle 'Getting Started with SharePoint Development'.

## Agenda

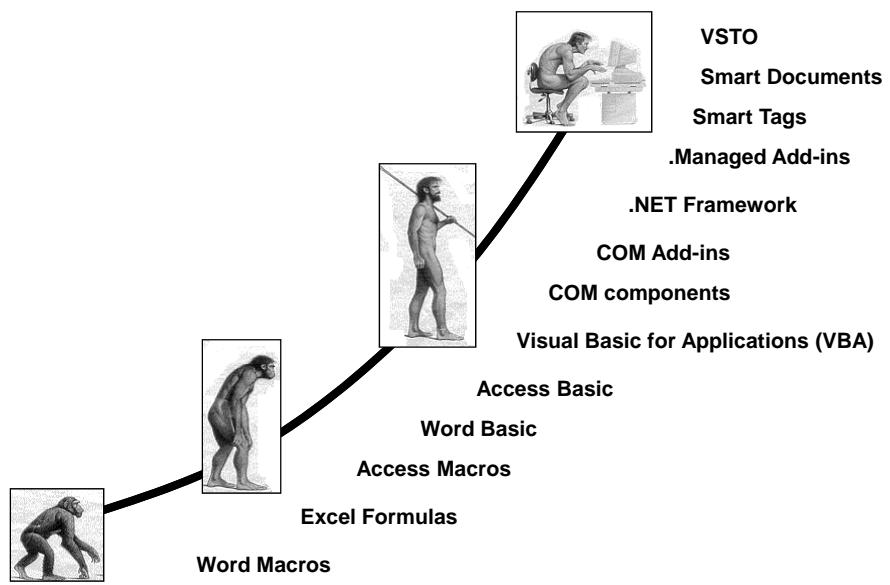
- Architectural overview of SharePoint 2007
  - Windows SharePoint Services 3.0 (WSS)
  - Microsoft Office SharePoint Server 2007 (MOSS)
- WSS as a collaboration solution
- WSS as a server-side development platform
- Overview of MOSS components and services

## Microsoft Office Through the Ages

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

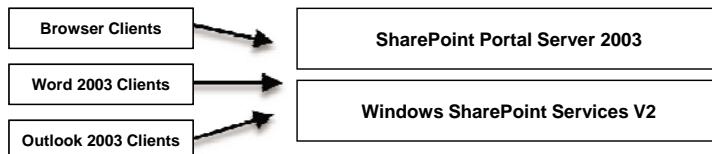


## Evolution of the Office Developer

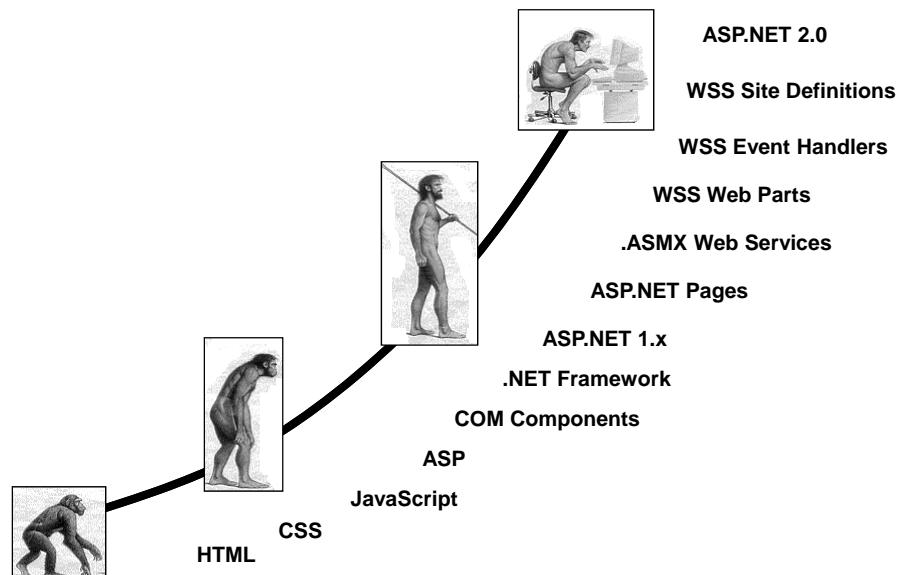


## Office 2003 Server Components

- 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

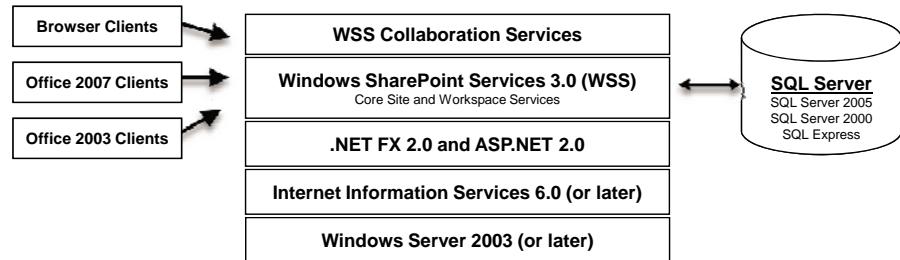
- What's Your Name?
- What Company are you with?
- How have you evolved as a Developer?
- Do you have experience with...
  - The .NET Framework and Visual Studio
  - ASP.NET (what was the latest version)
  - WSS 2.0 and SPS 2003
  - WSS 3.0 and MOSS

## 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-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

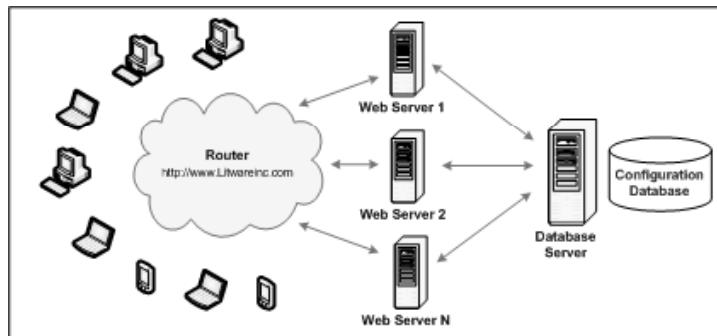
## The WSS 3.0 Server-side Platform

- 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



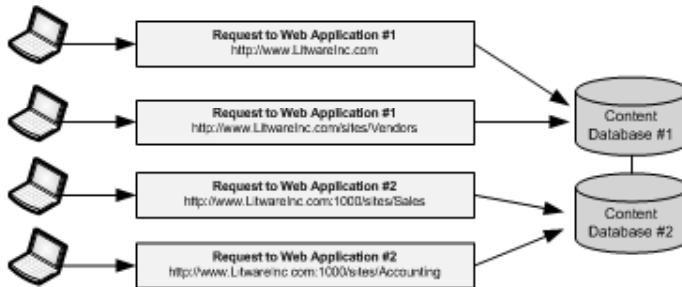
## The WSS Farm

- 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



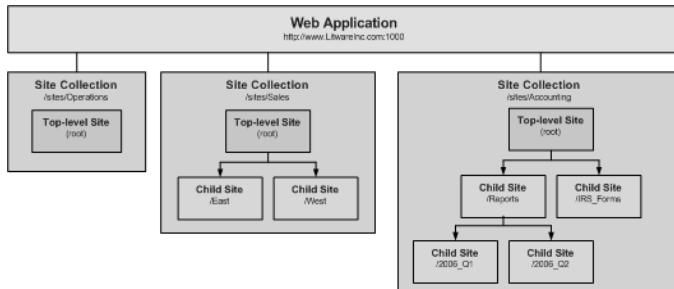
## Web Applications

- 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

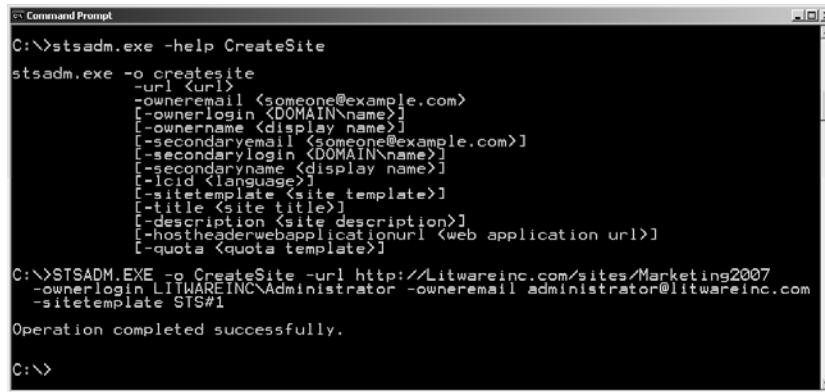


## Watch Out: Inconsistent Terminology

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

## STSADM.EXE Command-line Utility

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

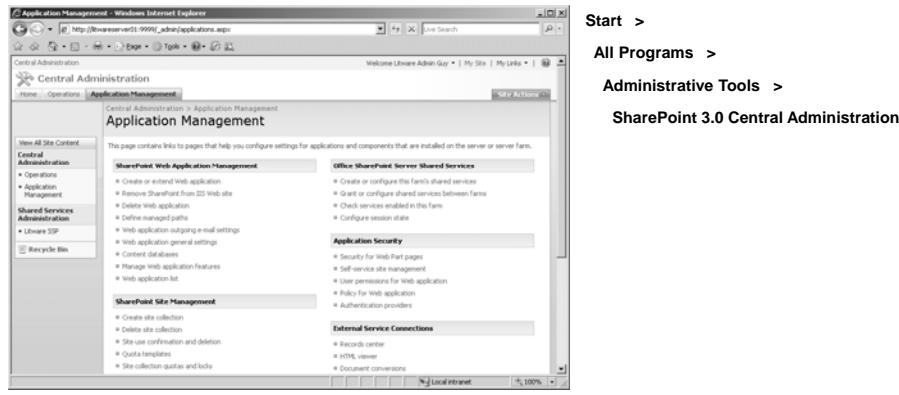


```
C:\>stsadm.exe -help CreateSite
stsadm.exe -o createsite
      [-url <url>
       [-owneremail <someone@example.com>
        [-ownerlogin <DOMAIN\name>]
        [-ownername <display name>]
        [-secondaryemail <someone@example.com>]
        [-secondarylogin <DOMAIN\name>]
        [-secondaryname <display name>]
        [-lcid <language>]
        [-sitetemplate <site template>]
        [-title <site title>]
        [-description <site description>]
        [-hostheaderwebapplicationurl <web application url>]
        [-quota <quota template>]}
C:\>STSADM.EXE -o CreateSite -url http://LitwareInc/sites/Marketing2007
      -ownerlogin LITWAREINC\Administrator -owneremail administrator@litwareinc.com
      -sitetemplate STS#1
Operation completed successfully.

C:\>
```

## WSS Central Administration (WSS CA)

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



## Creating New Site Collections

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

The screenshot shows the 'Create Site Collection' dialog box within the SharePoint 3.0 Central Administration interface. The dialog has fields for 'Web Application' (selected: 'http://lwsharec01:9999/\_vti\_bin/owssvr.dll/owssvr.dll'), 'Title' ('Unaware Home'), 'Description' ('A site for tracking Unaware sales information.'), and 'URL' ('http://lwsharec01/sites/Sales'). There are 'OK' and 'Cancel' buttons at the top right. The background shows the 'Central Administration > Application Management > Create Site Collection' page with the same form fields.

## 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



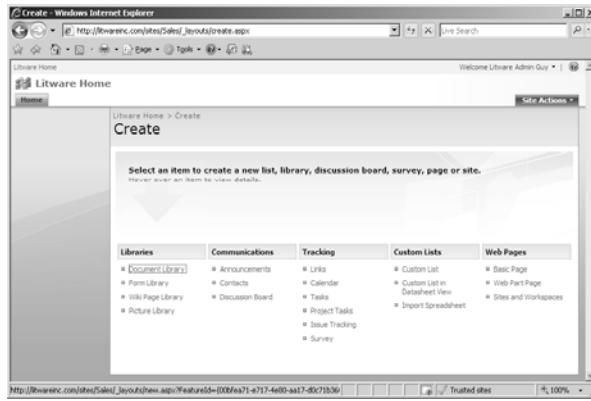
## The Site Settings Page

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



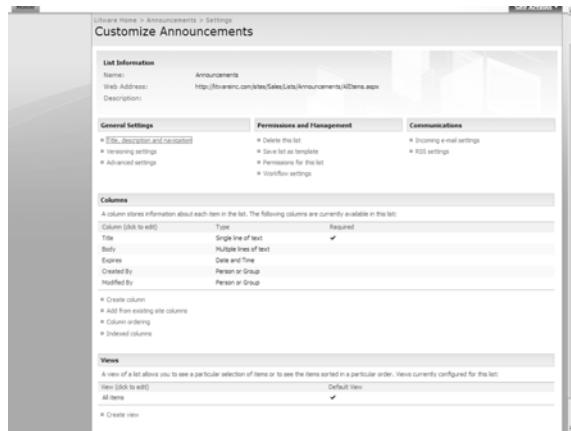
## The Create Page

- 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



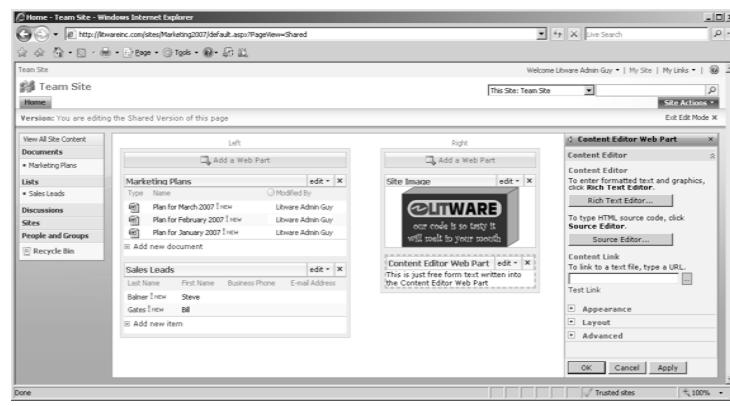
## The List Settings Page

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



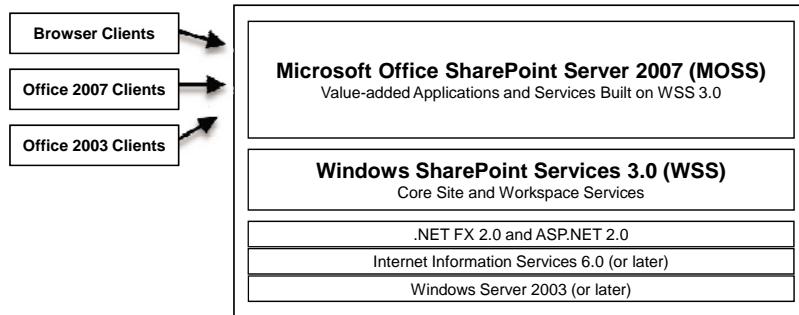
## Page Customization using Web Parts

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



## Microsoft Office SharePoint Server 2007

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

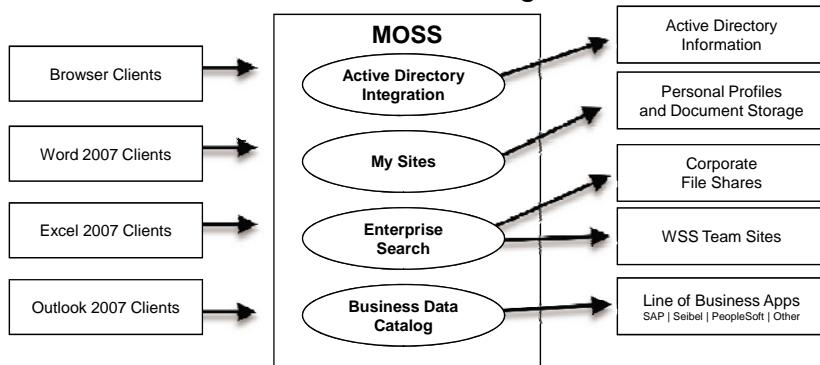


## MOSS Services and Components

- 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

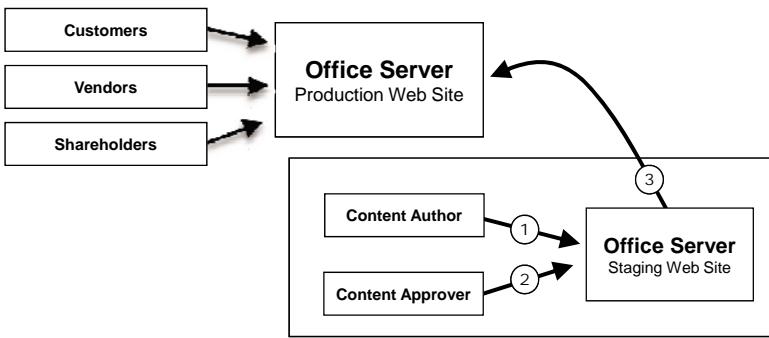
## Portal and Search

- 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

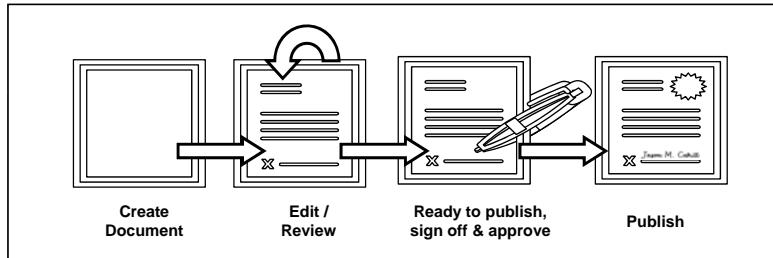


## InfoPath 2007 and Forms Services

- 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

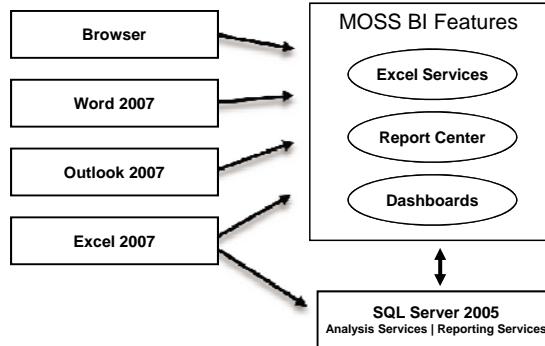
## SharePoint 2007 Workflows

- 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

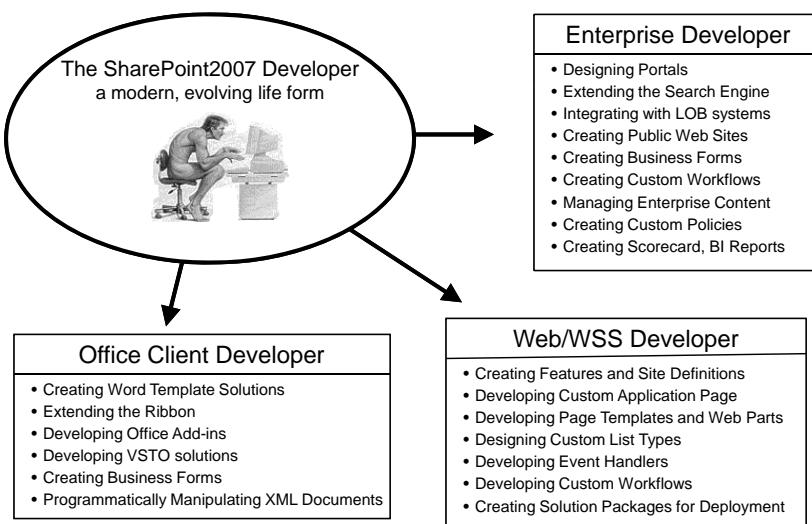


## Business Intelligence

- 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. SharePoint 2007 Roadmap <<< (you are here)
2. Developing Features
3. SharePoint Architecture
4. Pages and Site Branding
5. Developing Web Parts
6. Lists and Content Types
7. InfoPath 2007 and Forms Services
8. SharePoint Workflows
9. Portal Sites
10. Web Content Management
11. Business Data Catalog
12. Excel Services and Report Center
13. Application Security

## Summary

- Architectural overview of SharePoint 2007
  - Windows SharePoint Services 3.0 (WSS)
  - Microsoft Office SharePoint Server 2007 (MOSS)
- WSS as a collaboration solution
- WSS as a server-side development platform
- Overview of MOSS components and services



The slide features a dark background with a decorative border. At the top, there's a horizontal bar containing the Ted Pattison Group logo (a hard hat and compass) and a tiger silhouette. Below this is a large, semi-transparent gray rectangular area. In the center of this area, the title "Developing Features" is displayed in a large, serif font. Below the title, the subtitle "Creating SharePoint Components with Visual Studio" is shown in a smaller, sans-serif font.

## Agenda

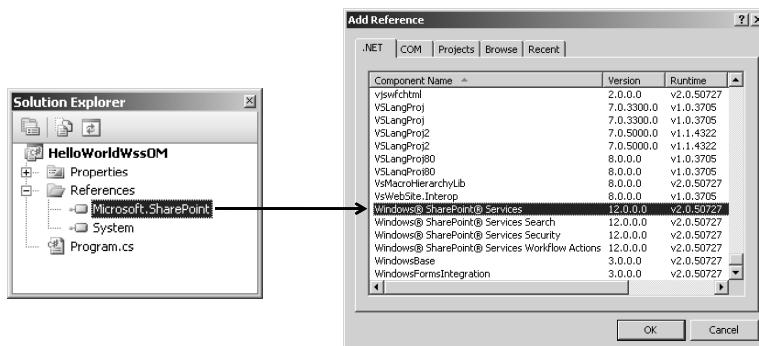
- SharePoint Customization versus Development
- The WSS system directories
- What Are Features?
- Developing a Custom Feature
- Adding Event Handlers to a Feature

## 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 sitePath = "http://itwareinc.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://litware.home/JQuery/ManageFeatures.aspx. The page displays a list of site features with columns for Name, Status, and Action buttons (Activate or Deactivate). The features listed are:

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 Provide team collaboration capabilities for a site by making standard lists, such as document libraries and issues, available.	Deactivate Active
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.	Deactivate 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

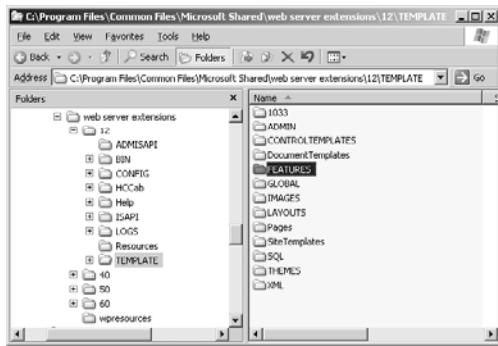
- Developers must learn WSS system directories

\12\TEMPLATE

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

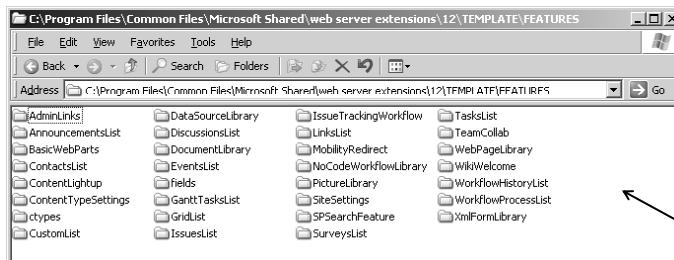
\12\TEMPLATE\IMAGES

\12\TEMPLATE\LAYOUTS



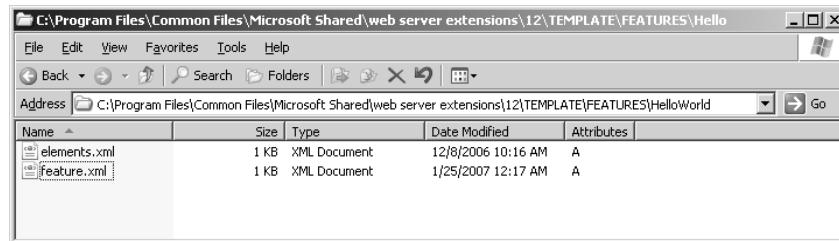
## The Features Directory

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



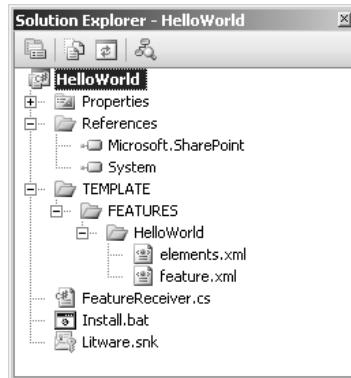
## Developer's View of a Feature

- 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="B2CB42E2-4FOA-4380-AABA-1EF9CD526F20"
  Title="A Sample Feature: Hello World"
  Description="Hi mom, class is fun. I am doing great"
  Scope="Web"
  Hidden="FALSE"
  ImageUrl="TPI\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="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 Visual Studio 8\SDK\v2.0\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

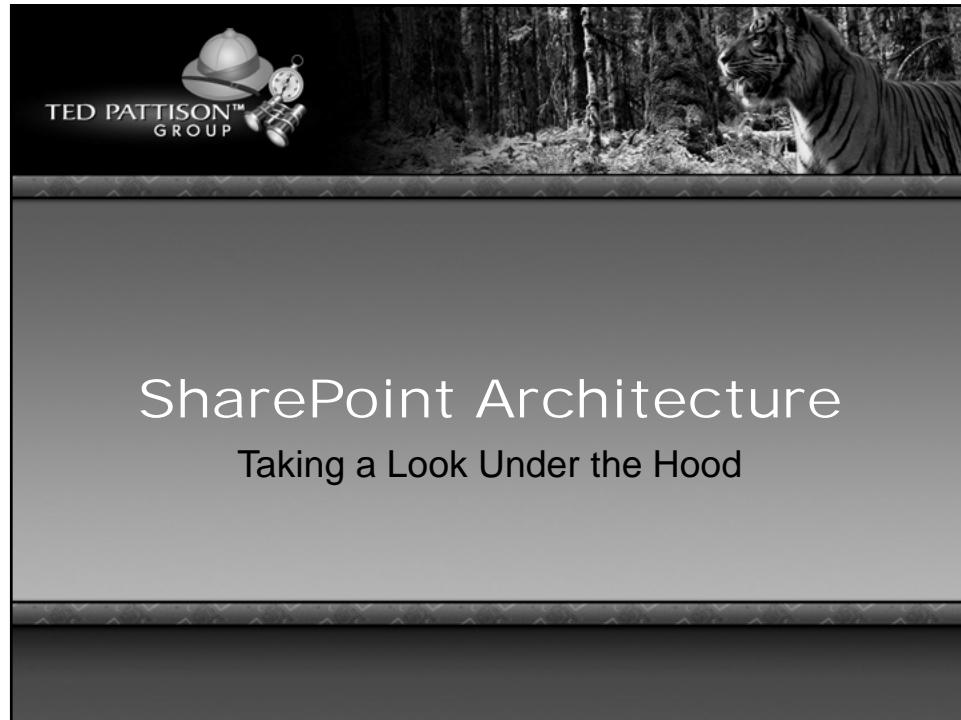
```
<Feature
  Id="B2CB42E2-4F0A-4380-AABA-1EF9CD526F20"
  Title="A Sample Feature: Hello World"
  Description="Hi mom, class is fun. I am doing great"
  Scope="Web"
  Hidden="FALSE"
  ImageUrl="TPG\WhitePlaceholder.gif"
  ReceiverAssembly="HelloWorld, [full 4-part assembly name]"
  ReceiverClass="HelloWorld.FeatureReceiver"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

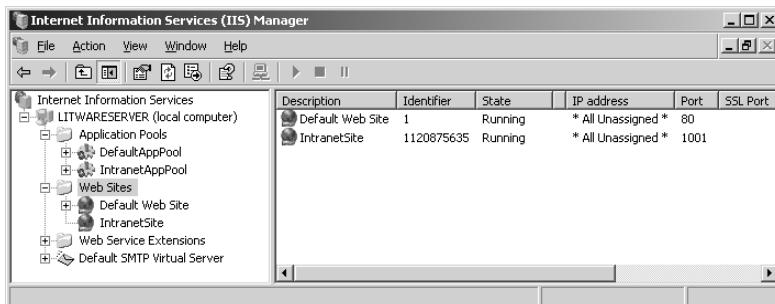


## Agenda

- 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

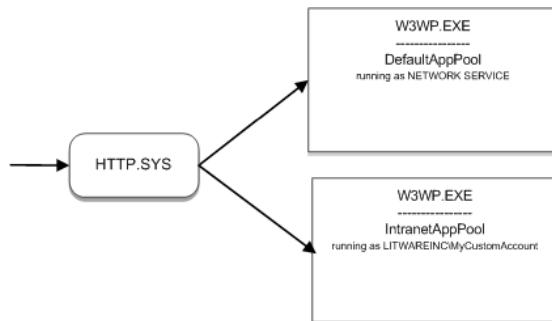
## IIS Web Sites

- 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

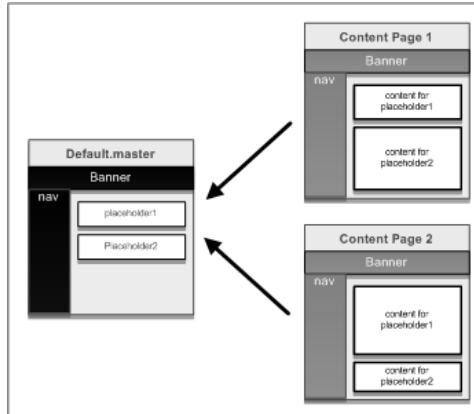
- ASP.NET development typically based on pages
  - Pages are deployed as .ASPX files to Web server
  - .ASPX files parsed and compiled on first request
  - Compiled page class inherits from **System.Web.UI.Page**

```
<%@ 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

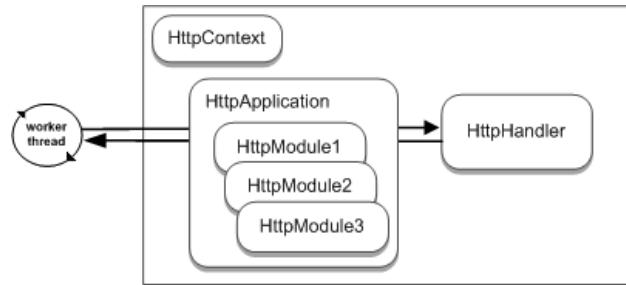
```
<!-- default.master -->
<%@ Master %>
<html><body>
  <form id="frmMain" runat="server">
    <table width="100%">
      <tr><td> <h1>Listware Inc. </h1><hr /></td></tr>
      <tr>
        <td> <!-- Display Main Body of Page -->
          <asp:ContentPlaceholder ID="PlaceHolderMain" runat="server"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

Internet Information Services	Name	Path
LITWARESERVER (local computer)	_controltemplates	C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\template\controltemplates
Application Pools	_layouts	C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\template\layouts
Web Sites	_vti_bin	C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\vti_bin
PublicWebSite	_wresources	C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\wresources
	App_Browsers	
	App_GlobalResources	
	aspnet_client	
	bin	
	wresources	
	_app_bin	
	_vti_pvt	
	global.asax	
	web.config	

- 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
    - \_wresources

## The WSS-extended web.config file

- WSS replaces HttpApplication object

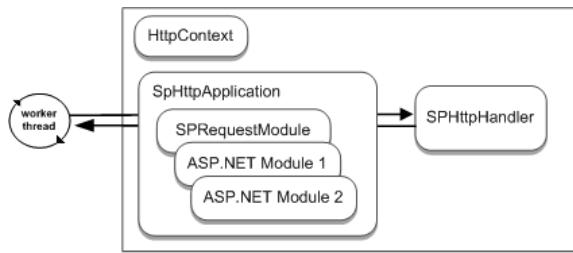
```
<!-- global.asax file at root of WSS Web Application -->
<@Application Inherit="Microsoft.SharePoint.ApplicationRuntime.SPHttpApplication" >
```

- 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="WebPartLists" 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 />
    <WebPartLists />
    <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" />
  </system.web>

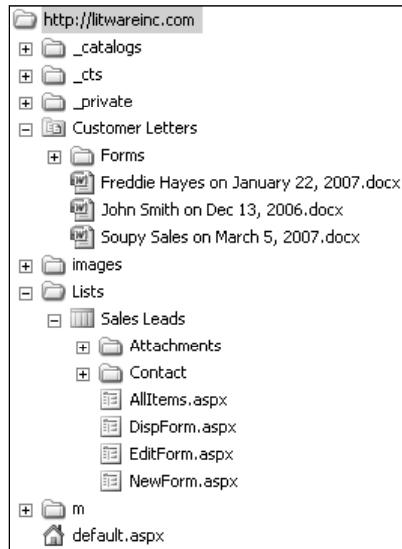
```

set to true

set to Off

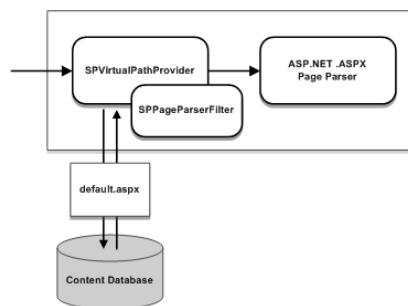
## The Virtual File System of a Site

- 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

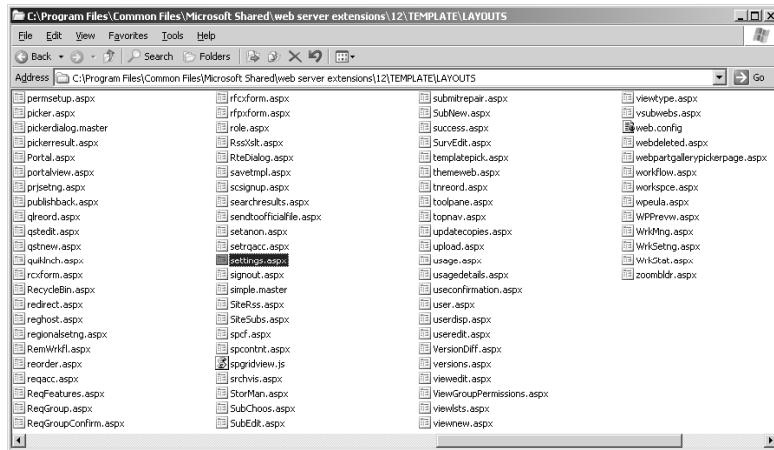


## The \_layouts Virtual Directory

- 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

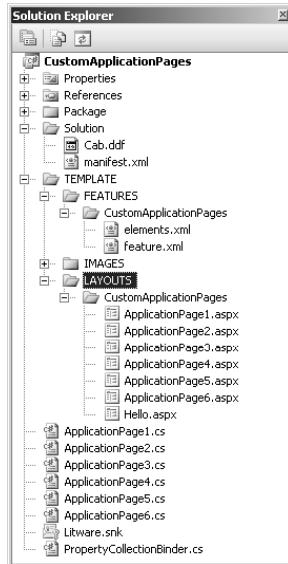
- Standard Application Pages are part of WSS



## Site Pages Versus Application Pages

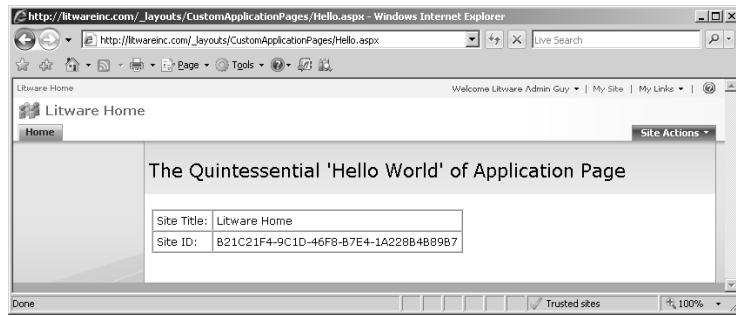
- Site Pages exist within virtual file system of site
  - They may or may not be ghosted
  - They support customization via Web Parts
  - They support customization via SharePoint Designer
  - Customized pages impact performance and security
- Application Pages are deployed once per farm
  - They do 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

## Demo: CustomApplicationPages



## 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
        LabelTitle.Text = site.Title;
        LabelTitleD.Text = site.ID.ToString().ToUpper();
    }
</script>
<asp:Content ID="Main" contentplaceholderid="PlaceHolderMain" runat="server">
    Site Title: <asp:Label ID="LabelTitle" runat="server"/><br />
    Site ID: <asp:Label ID="LabelTitleD" runat="server" />
</asp:Content>
<asp:Content ID="PageTitleArea" runat="server"
    contentplaceholderid="PlaceHolderPageTitleArea" >
    The Quaintest of 'Hello World' of Application Page
</asp:Content>
```

## Adding a Feature for Navigation

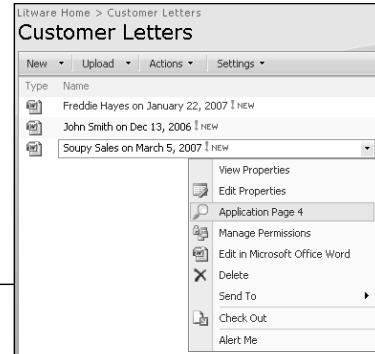
- Feature can be used with custom applications
  - Custom actions provide navigation menu items

```
<?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 ActionUrl="~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 ActionUrl="~site/_layouts/CustomAppli cati onPages/
  Appli cati onPage4.aspx?Itemld={Itemld}&ListId={ListId}" />
</CustomAction>
```

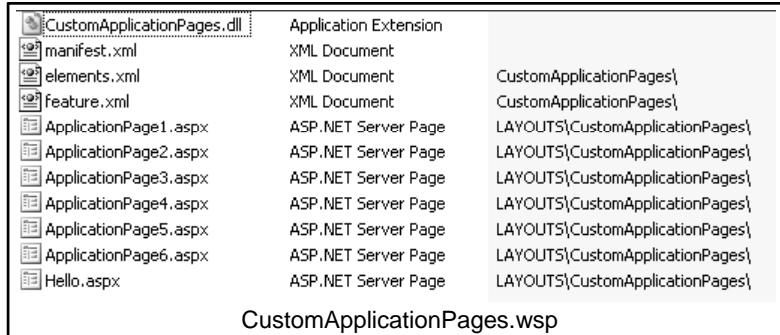


## Deployment using Solution Packages

- 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-BBCC-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

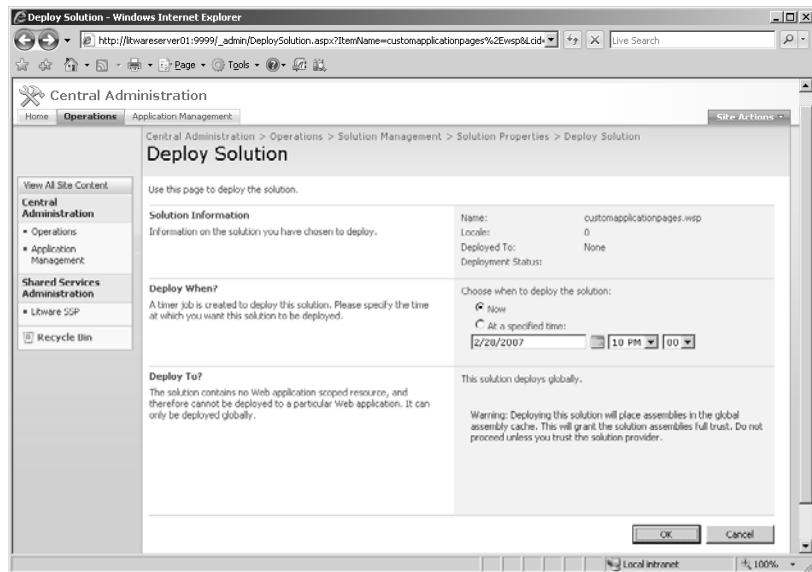
- Solution Package Installation
  - WSP file copied into configuration database
  - Done using **addsolution** operation of STSADM.EXE
- Solution Package Deployment
  - WSP files copied to each FE Web Server and deployed
  - Done using **deploysolution** operation of STSADM.EXE

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

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

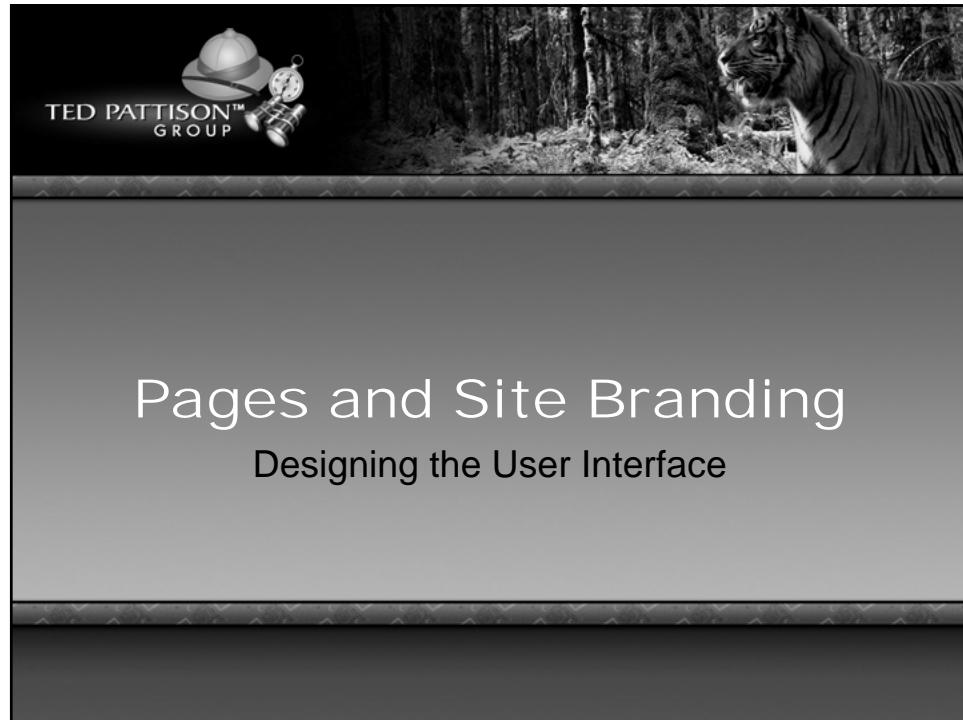
Echo Deploying Solution Package CustomApplicationPages.wsp
%STSADM% -o deploysolution -name CustomApplicationPages.wsp -immediate -allowGacDeployment
%STSADM% -o execadmsvcjobs
```

## Deploying Solution Packages



## Summary

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



## Agenda

- 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

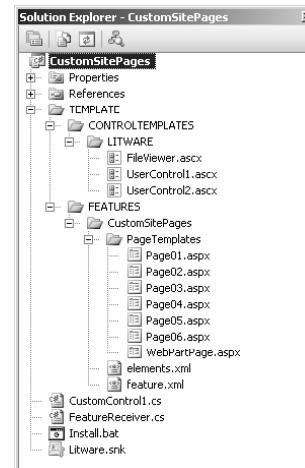
## Site Page Fundamentals

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



## Demo: CustomSitePages

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



## 'Hello World' Page Template

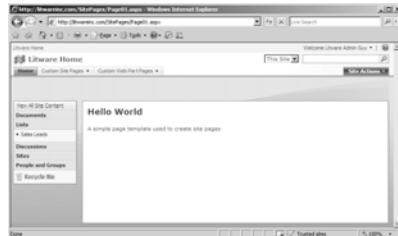
- Page Template can be added to feature
  - MasterPageFile points to ~masterurl /default.master
  - progid adds support for SharePoint Designer

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

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

## Provisioning a Page Instance

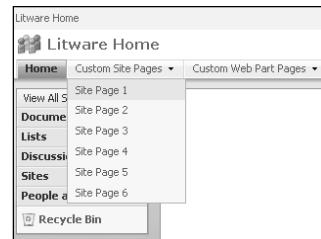
- Module element used to provision page instance
  - File element per page instance
  - Supports page ghosting



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

## Adding Navigation Support for Pages

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

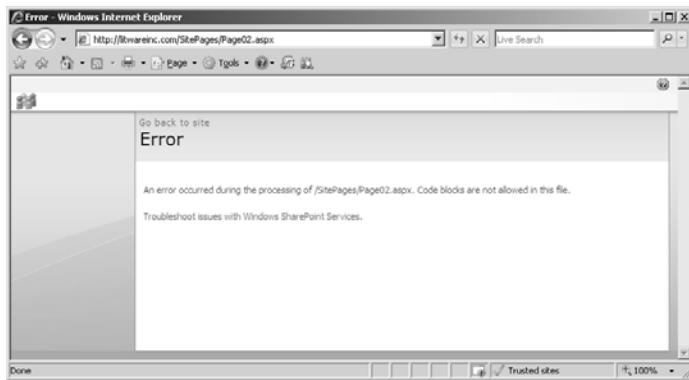


```
public class FeatureReceiver : SPFeatureReceiver {
    public override void FeatureActivated(SPFeatureReceiverProperties properties) {
        // get a hold off current site in context of feature activation
        SPWeb site = (SPWeb)properties.Feature.Parent;
        SPNavigationNodeCollection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

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

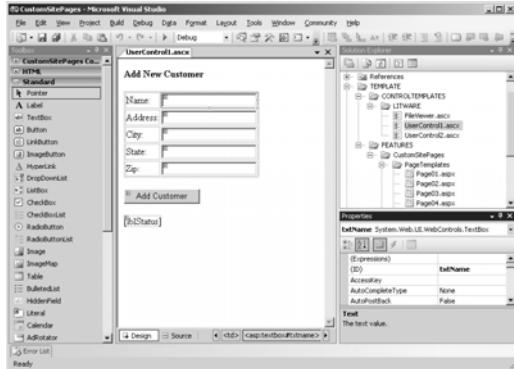
## Safe Mode Processing

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



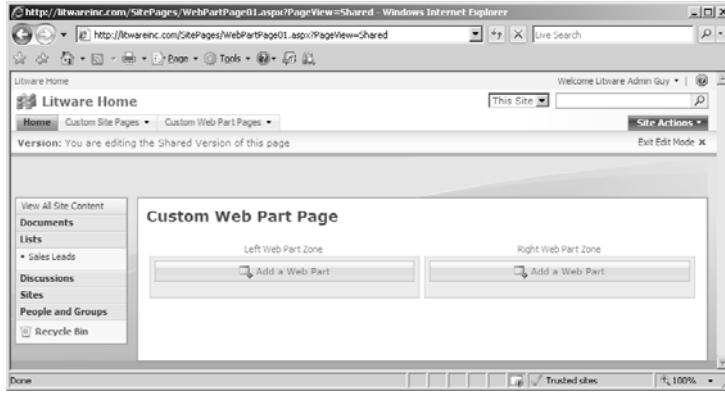
## Designing Pages with Controls

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



## Designing Web Part Pages

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



```

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

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

<asp:Content ID="main" runat="server" ContentPlaceHolderID="PlaceHolderMain">

<h3>Custom Web Part Page</h3>

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

</asp:Content>

```

## Adding Web Parts into Zones

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

```

<File Url="WebPartPage.aspx" Name="WebPartPage03.aspx" Type="Ghostable" >
    <!-- Add a Web Part to right zone -->
    <AllUsersWebPart WebPartZoneID="Right" WebPartOrder="0">
        <i><![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>

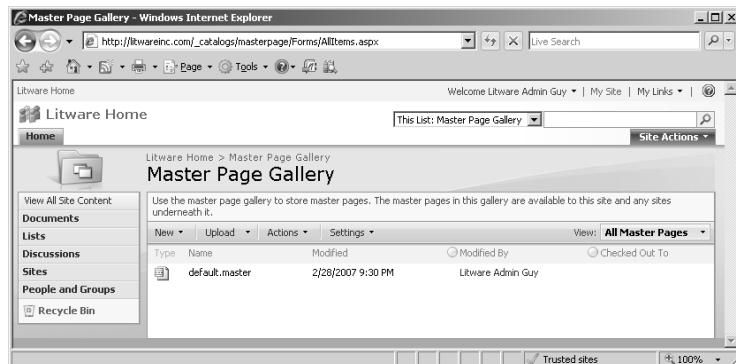
```

## Master Pages in WSS

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

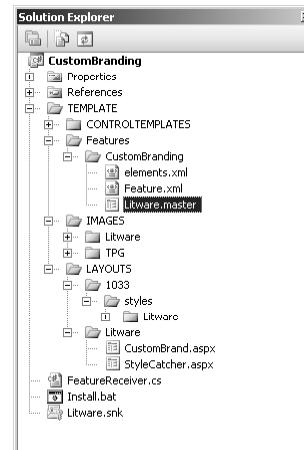
## The Master Page Gallery

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



## Demo: CustomBranding

- Important Concepts
  - Custom Master Page Templates
  - Custom CCS 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 like site page template
  - Support ghosting and unghosting
  - Provisioned using a File element within a Module

```
<Elements xml 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" runat="server">
        <asp:ContentPlaceholder ID=PlaceHolderPageTitle runat="server"/>
    </Title>
    <asp:ContentPlaceholder ID="PlaceHolderAdditionalPageHead" runat="server"/>

    <!-- Named 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 Master Page in 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);
        }
    }
}
```

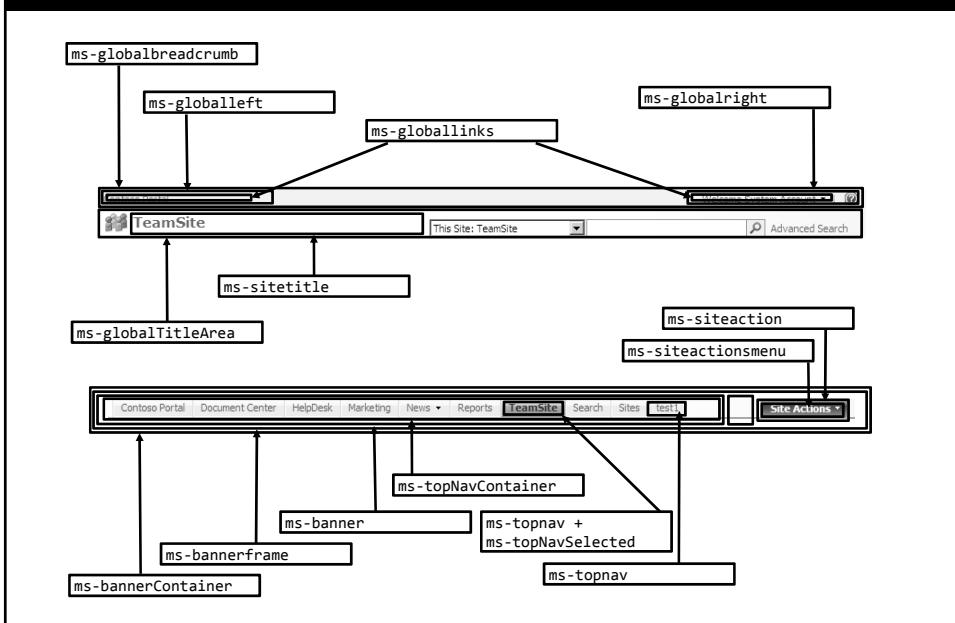
## Understanding core.css

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

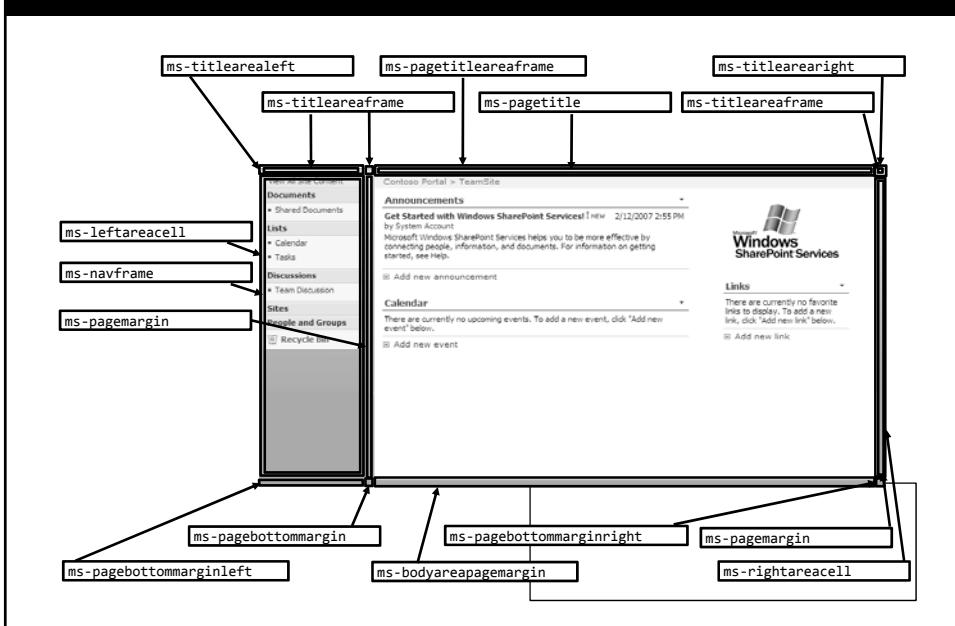
## Applying Custom Branding

```
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);  
    }  
}
```

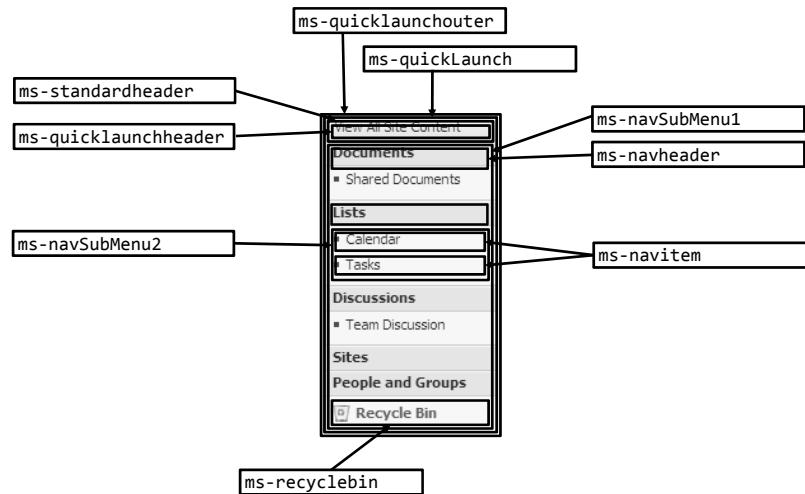
## Top Navigation Areas



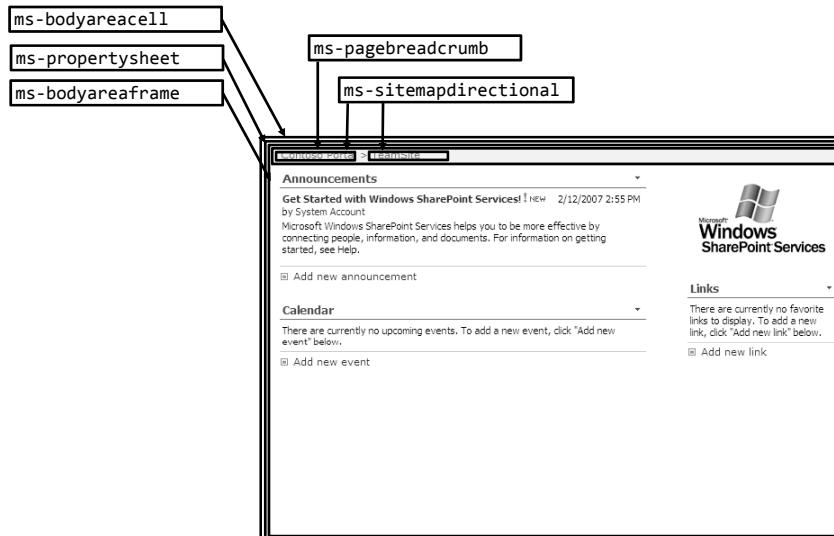
## Body Structure



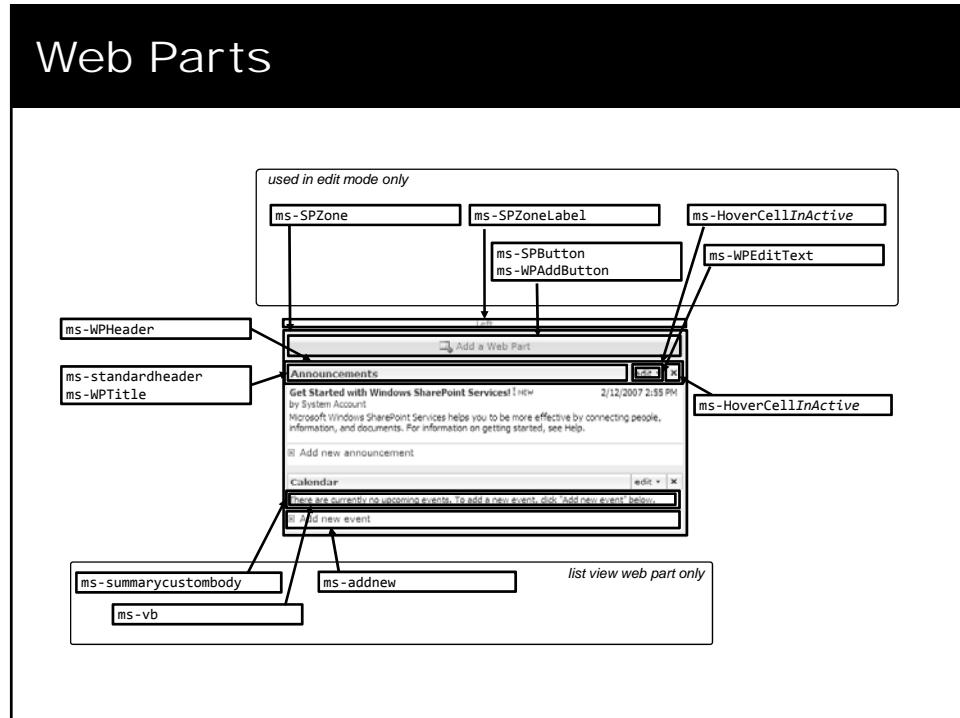
## Quick Launch



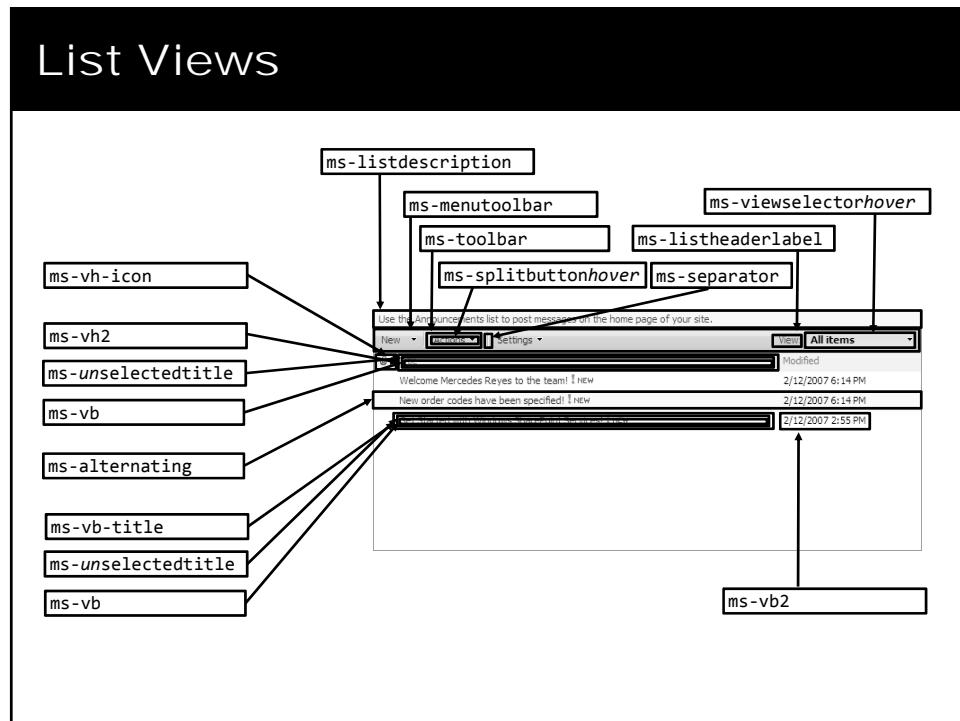
## Page Body



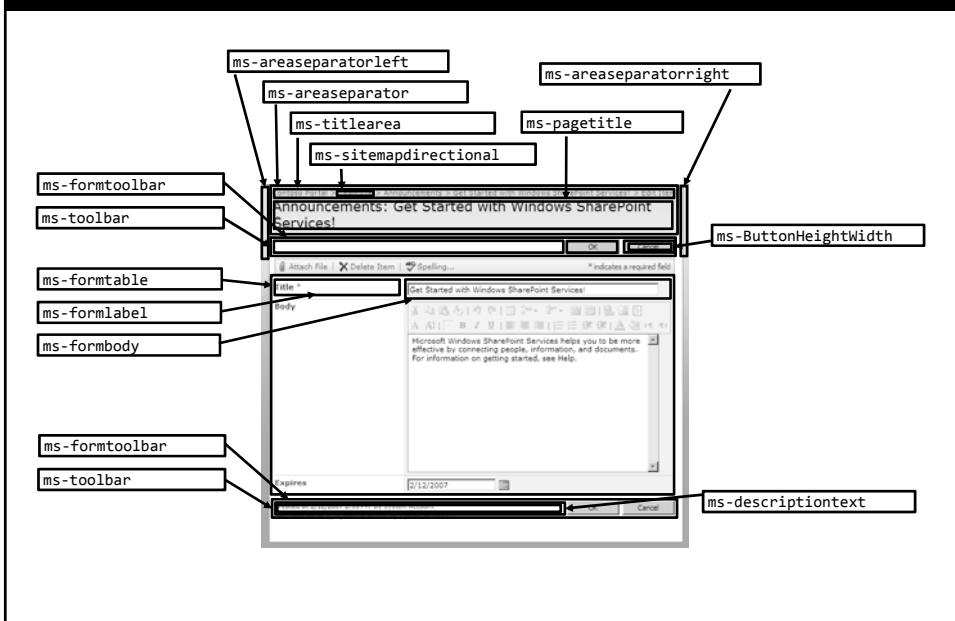
## Web Parts



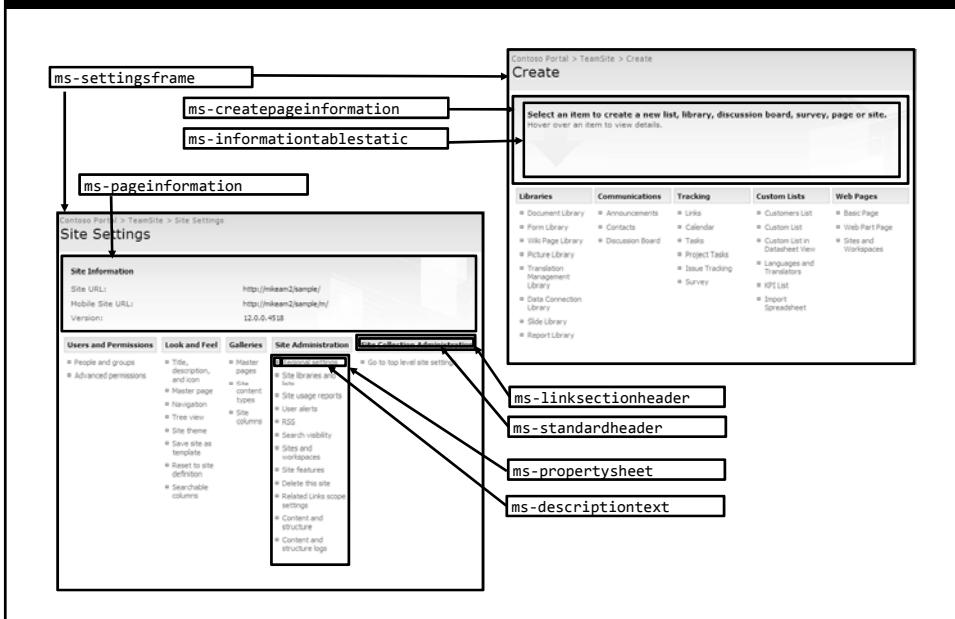
## List Views

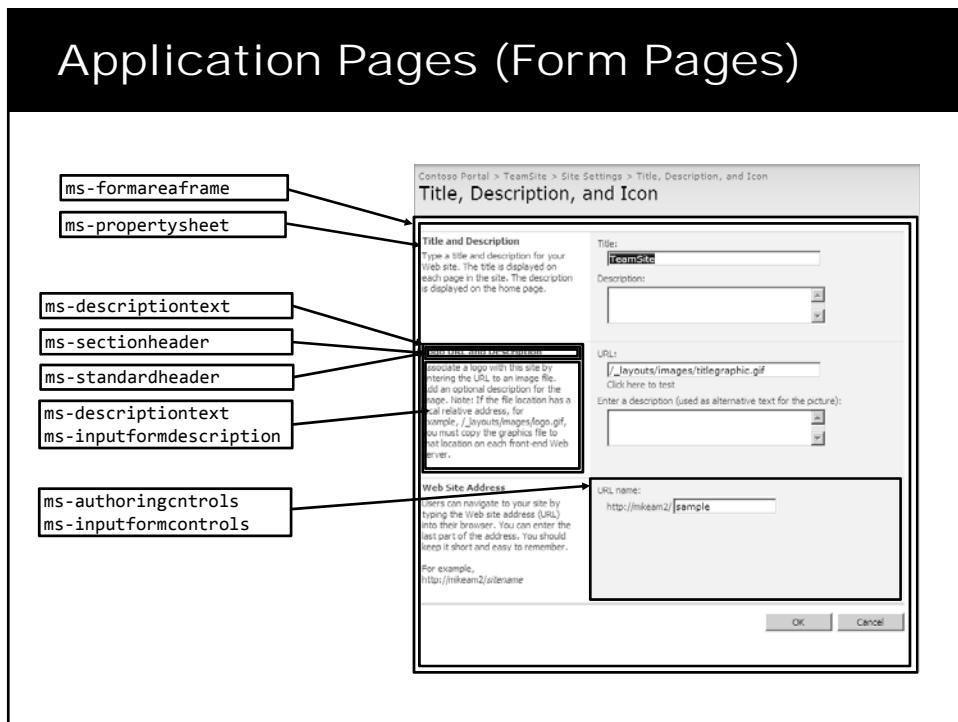


## Forms



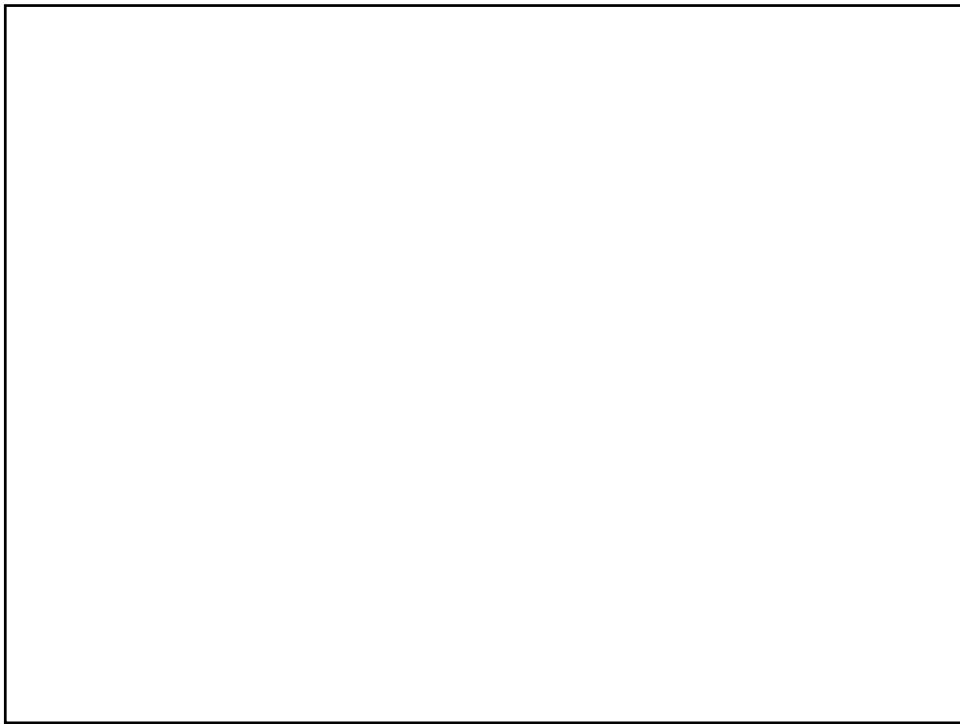
## Application Pages (Landing Pages)





## Summary

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





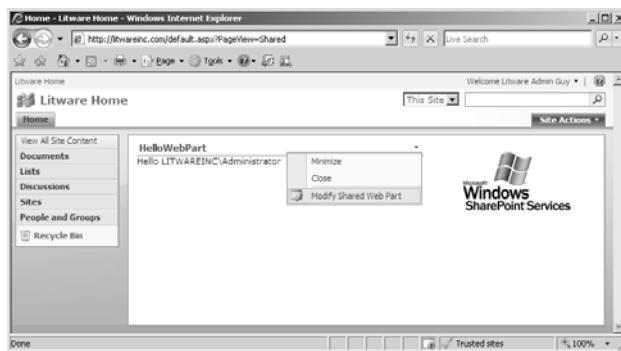
**Developing Web Part**  
Creating User Interface Components that  
Support Customization and Personalization

## Agenda

- Developing ASP.NET Web Parts for WSS 3.0
- Persistent Web Part properties
- Importing Web Parts into the Web Part Gallery
- Deploying Web Parts
- Advanced Web Part Techniques

## Web Parts

- Web Parts 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

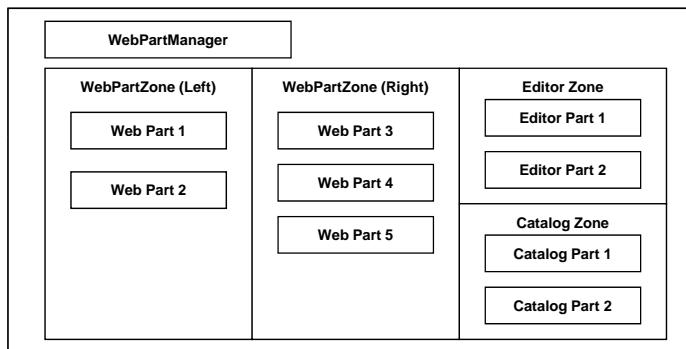


## Web Part History

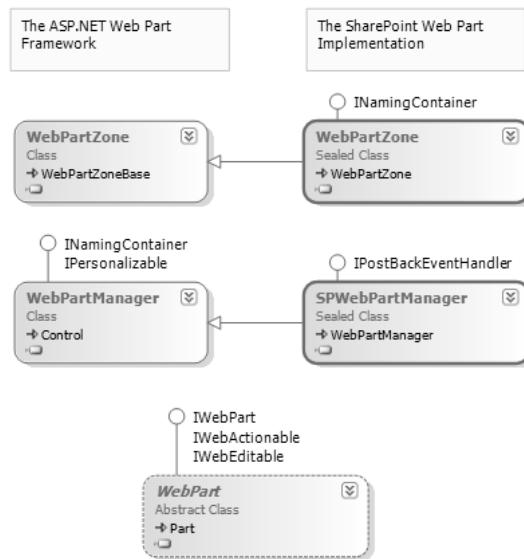
- Windows SharePoint Services 2.0 (WSS V2)
  - Designed with its own Web Part infrastructure
  - WSS serializes/stores/retrieves personalization data
- ASP.NET 2.0
  - Designed with newer universal Web Part infrastructure
  - Serializes/stores/retrieves personalization data
  - More flexible and more 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 WebPartManager class
  - One or more Web Part Zones
  - Optionally an Editor Zone and/or a Catalog Zone

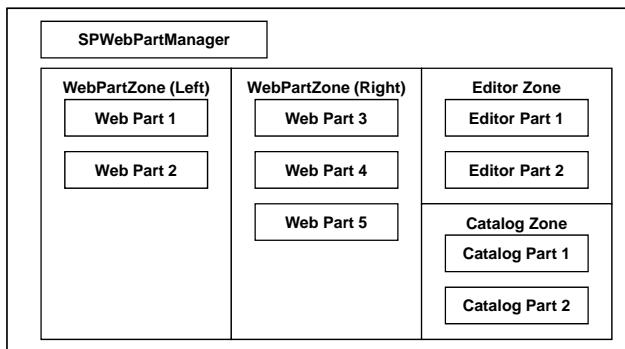


## SharePoint's Web Part Implementation



## WSS Web Part Page Structure

- Web Part Pages in WSS
  - Inherits from 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 WebPart base class
  - All Web Part inherit common functionality

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

namespace Li twareWebParts {
    public class HelloWorld : WebPart {
        protected override void RenderContents(HtmlTextWriter writer) {
            writer.WriteLine("Hello, world");
        }
    }
}
```

## Persistent Web Part Properties

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

```
namespace Li twareWebParts {
    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>
    <!-- <trust Level="WSS_Minimal" originUrl="" /> -->
    <trust Level="Full" originUrl="" />
  </system.web>
</configuration>
```

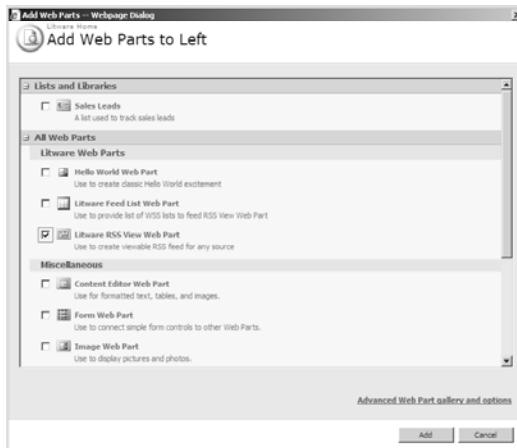
## The Web Part Gallery (WPG)

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

Litware Sales Site > Web Part Gallery Web Part Gallery				
Use this Web Part Gallery to store and retrieve Web Parts. The Web Parts in this gallery are available to this site and all sites under it.				
Type	Web Part	Edit	Modified	Modified By
	DemoAspWebPart.webpart [ NEW ]		1/2/2006 10:23 PM	LitwareInc Administrator
	DemoHybridWebPart.webpart [ NEW ]		1/2/2006 10:23 PM	LitwareInc Administrator
	DemoWssWebPart.dwp [ NEW ]		1/2/2006 10:23 PM	LitwareInc Administrator
	MSContentEditor.dwp		12/30/2005 11:03 AM	LitwareInc Administrator
	MSImage.dwp		12/30/2005 11:03 AM	LitwareInc Administrator
	MSMembers.dwp		12/30/2005 11:03 AM	LitwareInc Administrator
	MSPageViewer.dwp		12/30/2005 11:03 AM	LitwareInc Administrator
	MSSimpleForm.dwp		12/30/2005 11:03 AM	LitwareInc Administrator
	MSXml.dwp		12/30/2005 11:03 AM	LitwareInc Administrator

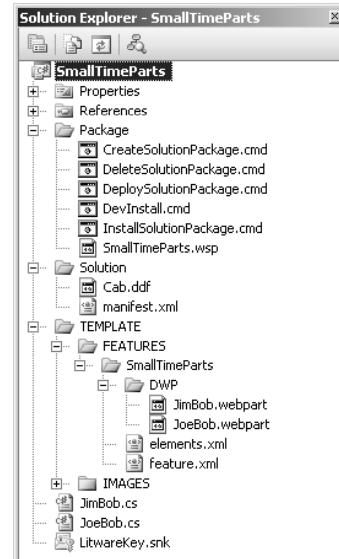
## Adding Web Parts from the Gallery

- WSS provides standard dialog for adding parts



## Demo: SmallTimeParts

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



## Provisioning .webpart files

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

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

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

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

  <File Uri ="JimBob.webpart" Type="GhostableInLibrary" >
    <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\SmallBinoscopes.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>

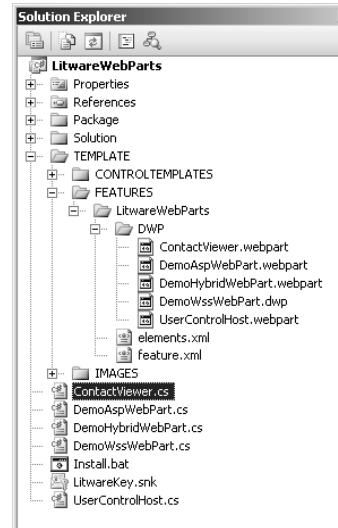
    <PolicyItem>
      <!-- create permission set for this policy -->
      <PermissionSet class="NamedPermissionSet" version="1">
        Description="Permission set for SmallImageParts assembly"
        <!-- add generic .NET CAS security permission -->
        <Permission class="SecurityPermission" version="1"
                   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



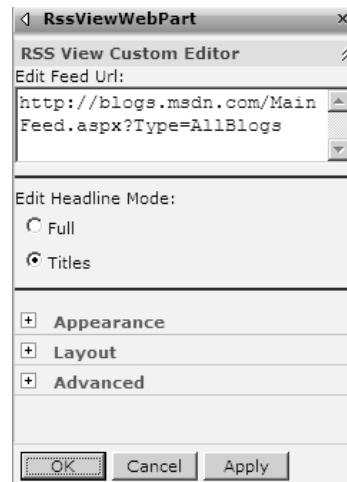
## Standard Editor Parts

- WSS provides standard editor parts



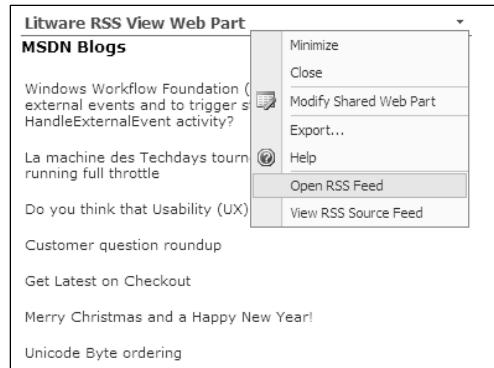
## A Custom Editor Part

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



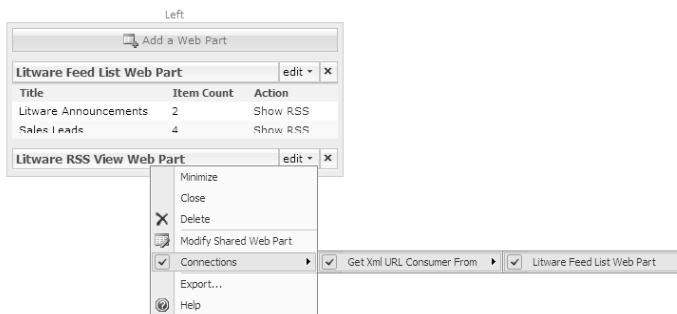
## Web Part Verbs

- 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



## 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 header features the Ted Pattison Group logo on the left, which includes a hard hat icon and the text "TED PATTISON™ GROUP". To the right of the logo is a black and white photograph of a tiger standing in a forest.

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

## Platform Storage Enhancements

- Parity between lists and document libraries
  - Folders supported for lists as well as document libraries
  - Versioning supported for list items as well as document
  - Events supported on lists as well as 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 expreditem in items) {
    Console.WriteLine(expreditem["Title"]);
}
```

## SPSiteDataQuery

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

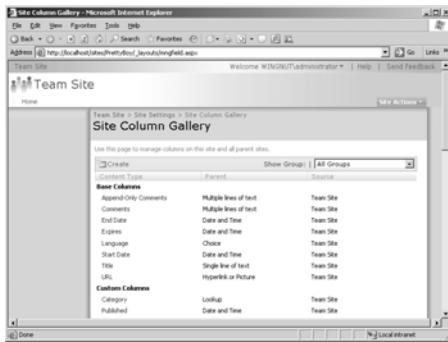
```
SPSiteDataQuery query = new SPSiteDataQuery();
query.Lists = @"<Lists ServerTemplate='104' />";
query.ViewsFields = @"<Fields FieldRef Name='Title' /><Fields FieldRef Name='Created' />";
query.Webs = "<Webs Scope='SiteCollection' />";
string queryText =
@"<Where>
<Eq>
<FieldRef Name='Created' />
<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
  - 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 application should make list of actions available to users depend on the type of content or document
- WSS provides new features to solve these problems
  - Site Columns
  - Content Types

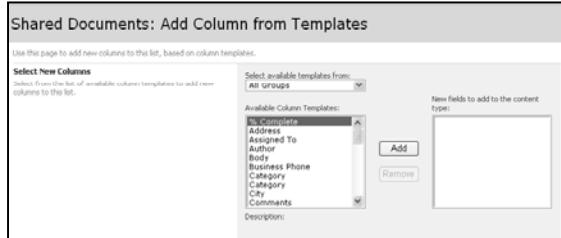
## Site Columns

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



## Using a Site Column in a List

- Site Columns can be used in List Definitions
  - Site column represents a reusable, named column definition
  - Site column used in lists, document library 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



## 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

- Content type for Proposals
  - Requires string column for tracking client
  - Requires boolean column for legal review
- Content Type for Presentation
  - Requires string column for tracking client
  - Requires boolean column for art review

## Content Types

- A content type definition can include...
  - Columns to represent metadata or properties
  - A document template for creating new documents
  - Custom forms for New, Edit, and Display
  - Event handlers
  - Workflows

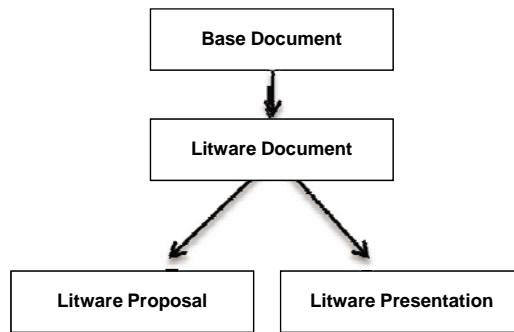
## Supporting Multiple Content Types

- Lists can support multiple content types
  - Makes it possible to support heterogeneous content
  - “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 type
  - Base content type inherited by more specific content types

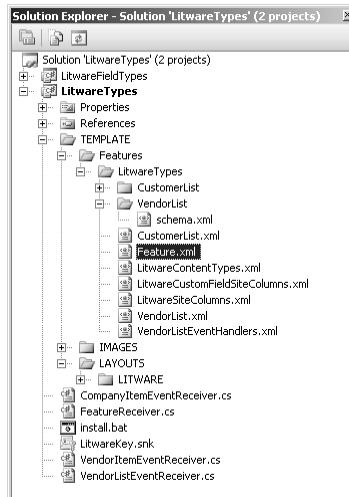


## Creating Content Types

# DEMO

## 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 significantly improved
  - Events supported for lists, document libraries and content types
  - Events supported for changes to list schema as well as items
  - Events supported at site collection and site level
  - Events supported for incoming email messages
  - Support for synchronous events and asynchronous events
  - Synchronous events occur before the fact and are cancel-able
- How do you get events to work
  - Create custom class inheriting a WSS receiver class  
e.g. SPItemEventReceiver or SPWebEventReciever
  - 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();
            SPListItems 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



**TED PATTISON™  
GROUP**

**Forms Services  
and InfoPath 2007**

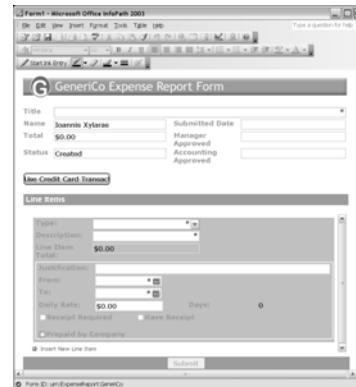
Designing browser-based forms to  
capture schema-validated XML data

## Agenda

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

## The Role of InfoPath in Office 2003

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



## Challenges with InfoPath 2003

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

## Inside an InfoPath Form

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

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

## Security – Trust Levels



### Restricted

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



### Domain

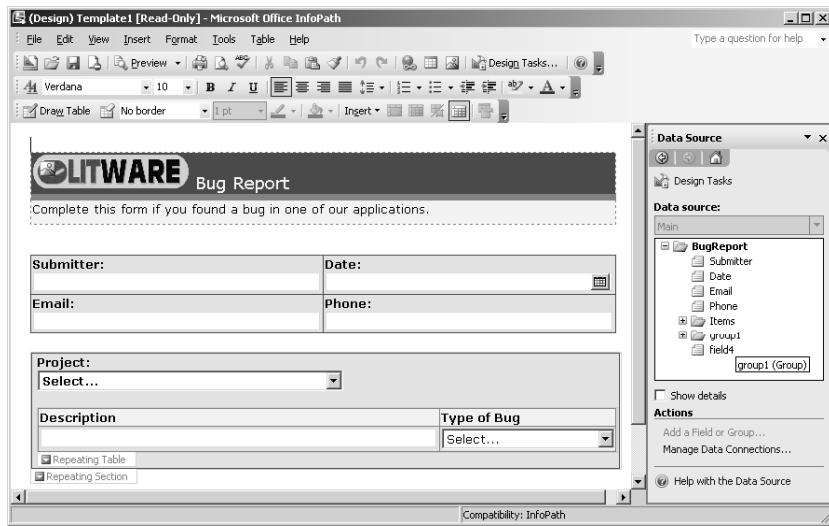
- Deployed to SharePoint library, browser forms
- Connect to own server only, no code for browser forms
- Use trusted Data Connection Library for cross-domain



### Full Trust

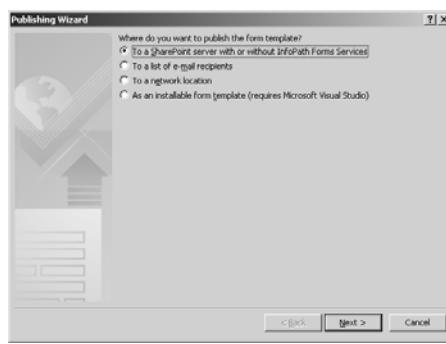
- Installed, Digitally Signed, or .NET Code Group
- Must be admin-deployed for browser forms
- Connect to any server, managed code in browser forms

## Demo: The InfoPath Forms Designer

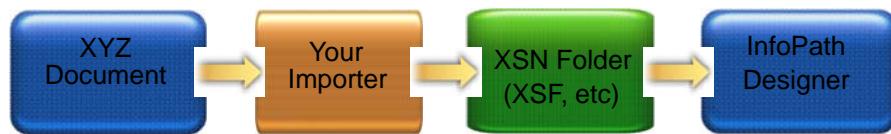


## InfoPath Integration with WSS

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

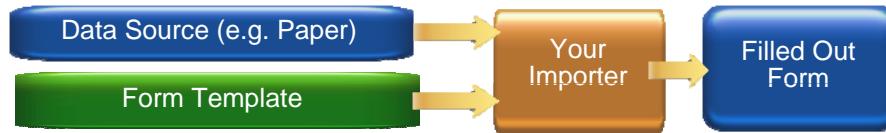


## Template Importing



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

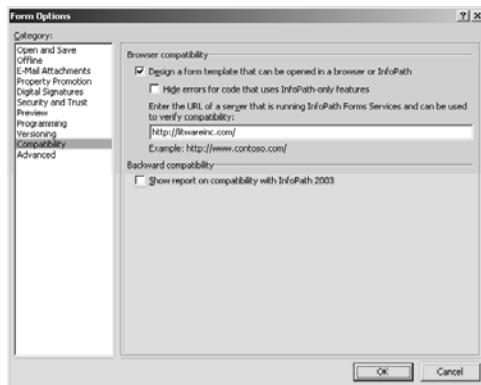
## Data Importing



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

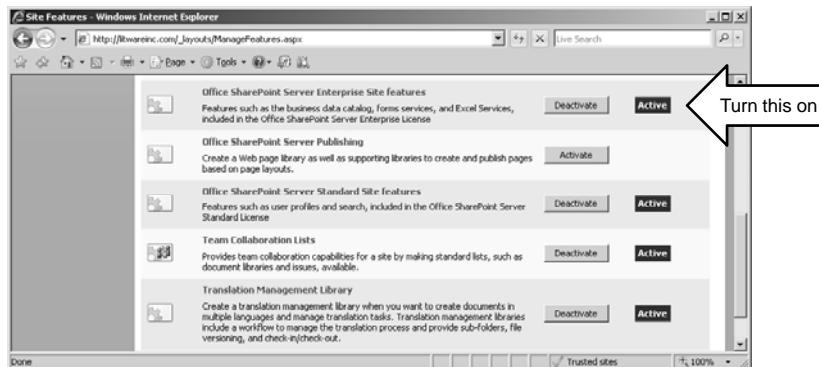
## Browser-based Forms

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

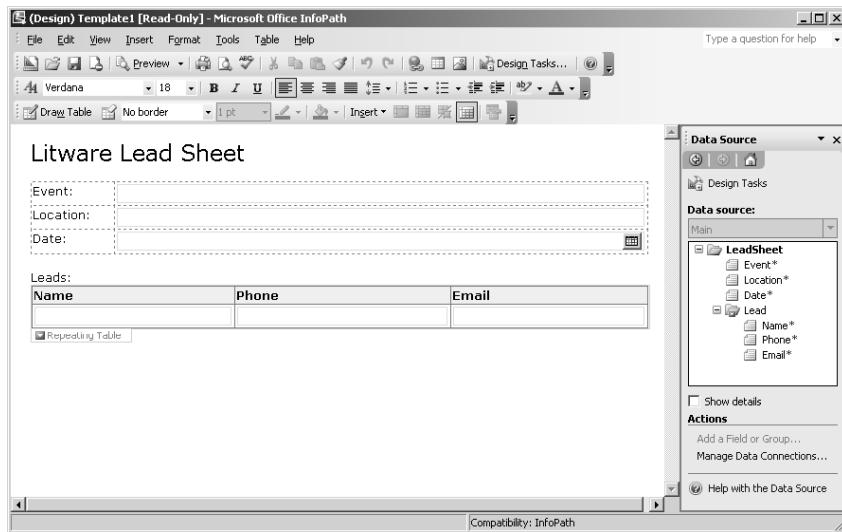


## Sites and Browser-based Publishing

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

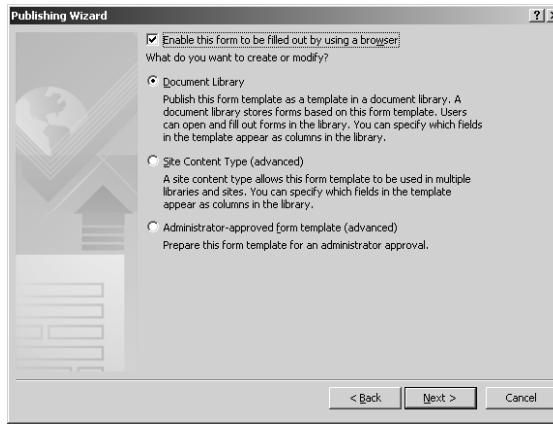


## Designing a Browser-based Form



## Publishing a Browser-based Form

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



## Forms Library Settings

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

Litware Home > Leadsheets > Settings > Advanced Settings

**Form Library Advanced Settings: Leadsheets**

<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.	Allow management of content types? <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.	Template URL: <input type="text" value="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.	Opening browser-enabled documents <input type="radio"/> Open in the client application <input checked="" type="radio"/> Display as a Web page

## Browser-based Rendering

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

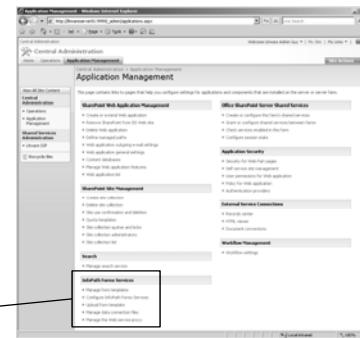
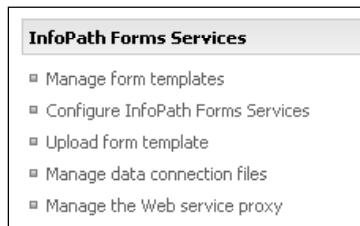
The screenshot shows a Microsoft Internet Explorer window displaying a web-based form titled "Leadsheets". The form is titled "Litware Lead Sheet". It contains several input fields: "Event" (Tech Ed Europe 2006), "Location" (Barcelona), and "Date" (11/6/2006). Below these is a table with the header "Leads:" and three data rows:

Name	Phone	Email
Bob Hinkle	(123)456-5789	a@b.com
Fred Dinkle	(987)654-3210	c@d.com
Cindy Sprinkle	(866)248-1357	e@f.com

At the bottom of the form is a link "Insert item". The browser status bar at the bottom shows the URL "http://litwareinc.com/\_layouts/FormServer.aspx?OrLocation=http://litwareinc.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

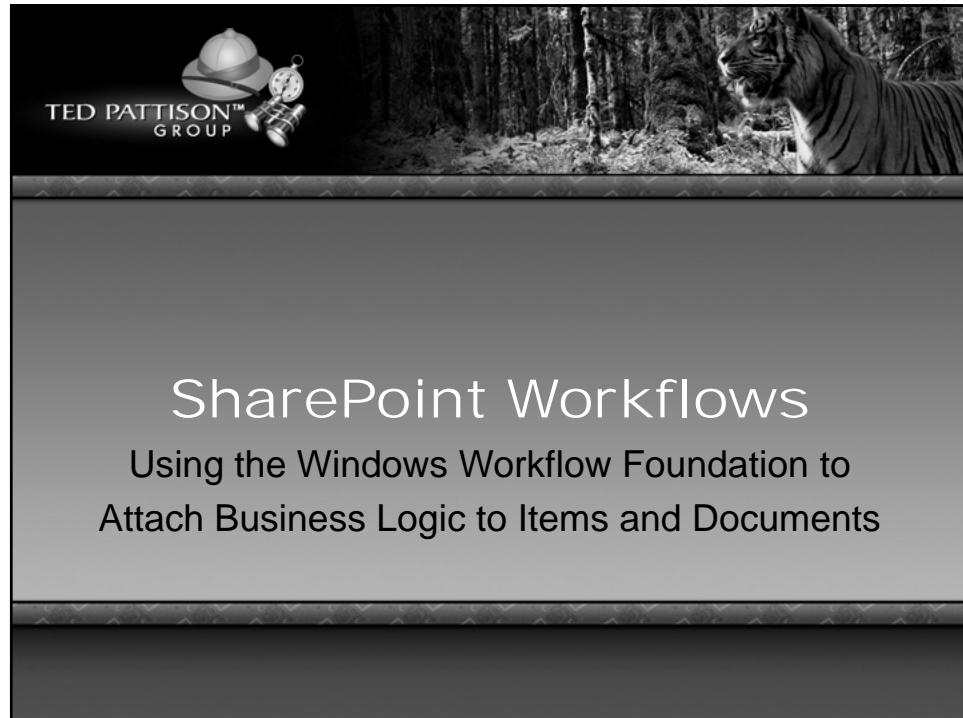
Manage Form Templates - Windows Internet Explorer					
Central Administration > Application Management > Manage Form Templates					
Salt Name	Type	Name	Version	Modified	Category
All Forms	Upload Form Template	ReRevolutionForm.vsn	1.0.0.30	2/21/2007	Ready
		ReviewTaskForm.vsn	1.0.0.34	2/21/2007	Ready
		CollectSignature_Sign_1003.vsn	12.0.0.1	2/21/2007	Workflow
		Expiration_Complete_1003.vsn	12.0.0.1	2/21/2007	Workflow
		RR_COB_ImportB_R_1003.vsn	12.0.0.1	2/21/2007	Workflow
		ReRevolutionForm_Access_1003.vsn	12.0.0.1	2/21/2007	Workflow
		ReRevolutionForm_Inv_1003.vsn	12.0.0.1	2/21/2007	Workflow
		ReRevolutionForm_Modify_1003.vsn	12.0.0.1	2/21/2007	Workflow
		ReRevolutionForm_Archive_1003.vsn	12.0.0.1	2/21/2007	Workflow
		ReRevolutionForm_UpdateTask_1003.vsn	12.0.0.1	2/21/2007	Workflow
		Rate_Complete_1003.vsn	12.0.0.1	2/21/2007	Workflow
		Rate_Inv_1003.vsn	12.0.0.1	2/21/2007	Workflow

## 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)



The slide features a dark background with a decorative border. In the top left corner is the Ted Pattison Group logo, which includes a hard hat icon and the text "TED PATTISON™ GROUP". The main title "SharePoint Workflows" is centered in large white font. Below it is a subtitle in smaller white font: "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

## Activities

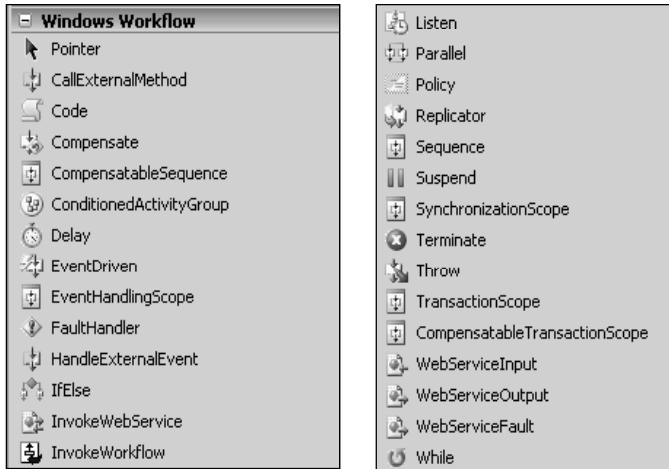
- An activity is...
  - atomic set instructions used complete a unit of work
  - reusable component used to compose WF programs
- Activities are like controls in forms development
  - You drag and drop them onto a design surface
  - You modify their properties through property sheet
  - You generate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

## 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

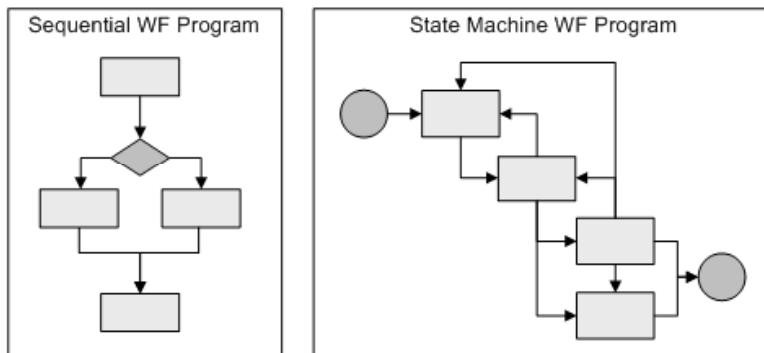
## WF Base Activity Library

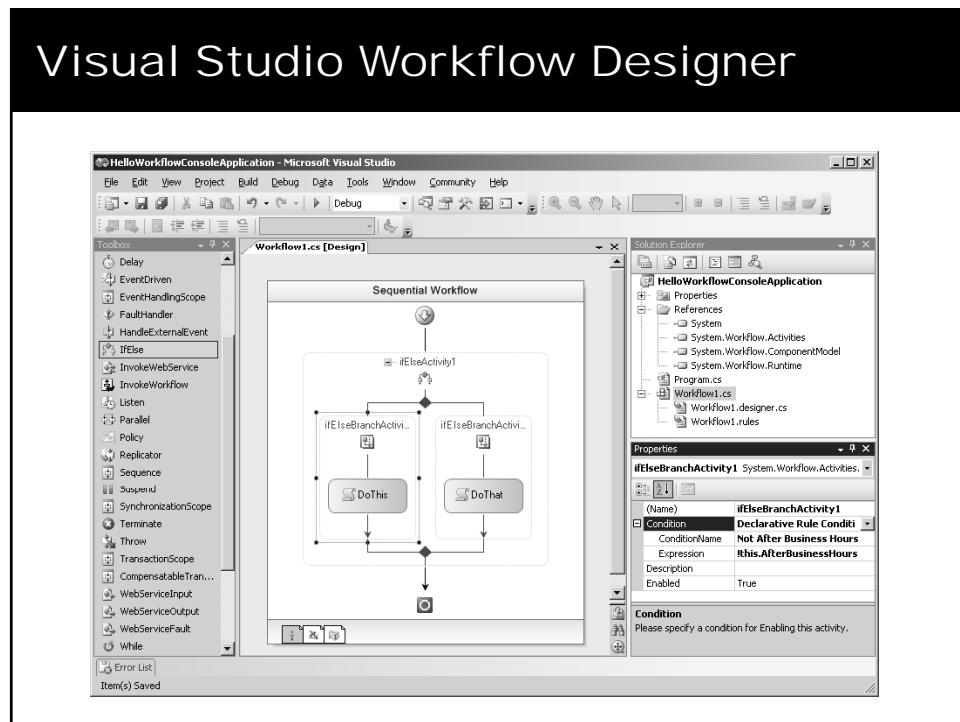
- Standard WF activities provide basic building blocks



## 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
            WorkflowRuntime wfRuntime = new WorkflowRuntime();
            wfRuntime.StartRuntime();

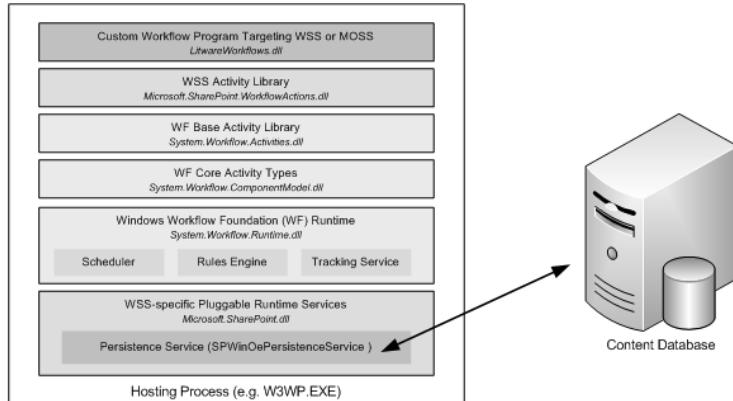
            // create and start workflow instance
            WorkflowInstance workflowInstance1;
            workflowInstance1 = wfRuntime.CreateWorkflow(typeof(Workflow1));
            workflowInstance1.Start();

            // pause until test workflow instance completes
            Console.ReadLine();

            // stop WF runtime
            wfRuntime.StopRuntime();
            wfRuntime.Dispose();
        }
    }
}
```

## WF Runtime Services

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



## SharePoint Workflows 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 status on 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 item

## Creating a Workflow Association

Add a Workflow: Proposals

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

**Workflow**  
Select a workflow to add to this document library. If the workflow template you want does not appear, contact your administrator to get it added to your site collection or workspace.

**Name**  
Type a name for this workflow. The name will be used to identify this workflow to users of this document library.

**Task List**  
Select a task list to use with this workflow. You can select an existing task list or request that a new task list be created.

**History List**  
Select a history list to use with this workflow. You can select an existing history list or request that a new history list be created.

**Start Options**  
Specify how this workflow can be started.

Select a workflow template:	Description:
<input checked="" type="checkbox"/> Collect Feedback <input checked="" type="checkbox"/> Collect Signatures <input checked="" type="checkbox"/> Disposition Approval <input checked="" type="checkbox"/> Three-state	Use this workflow to track items in a list.
Type a unique name for this workflow:	
My First Workflow Association	
Select a task list:	Description:
<input checked="" type="checkbox"/> Tasks	Task list for workflow.
Select a history list:	Description:
<input checked="" type="checkbox"/> Workflow History	History list for workflow.
<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**

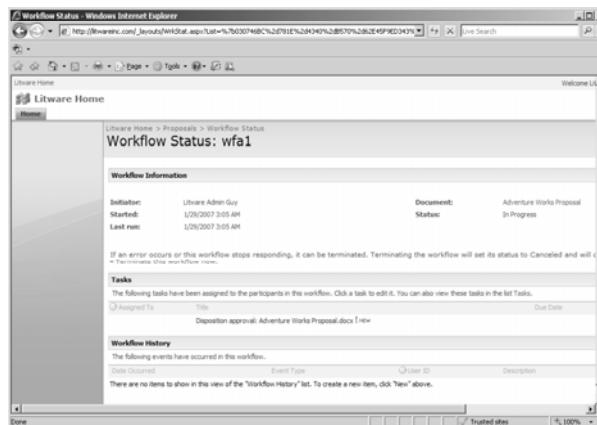
## Starting a Workflow Instance

- Users can manually start workflows



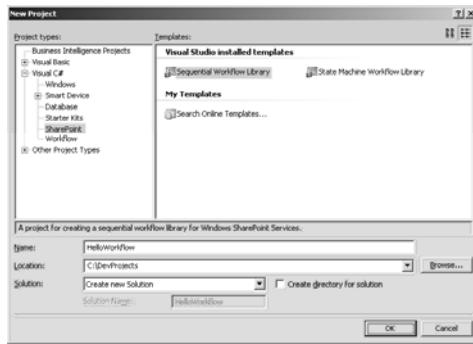
## The Workflow Status Page

- Any user can see the status of workflow instance



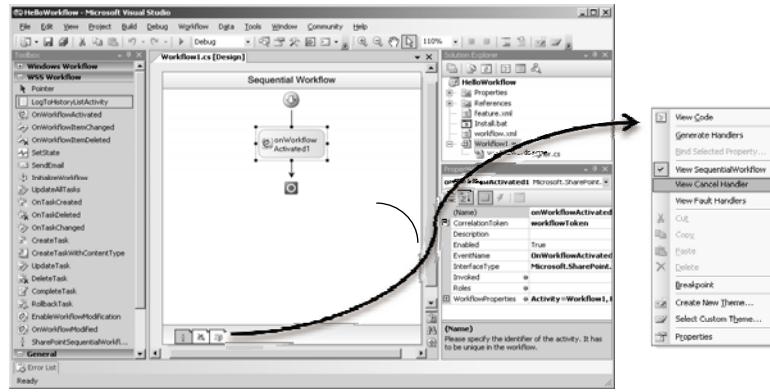
## Creating a Workflow Template Project

- Creating a SharePoint Workflow Template
  - First, install the WF Extensions for Visual Studio
  - Second, install either the WSS SDK or the MOSS SDK
  - Third, start creating SharePoint Workflow projects



## 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 SharePoint Activity Library



## Working in Code View

- Here is what you get as a starting point

```
using Microsoft.SharePoint.Workflow;

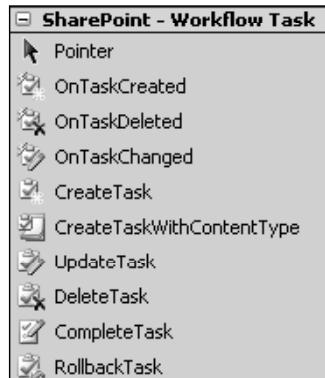
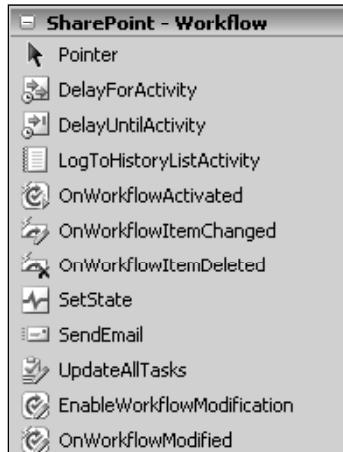
namespace HelloWorkflow {
    public partial class Workflow1 : SharePointSequentialWorkflowActivity {
        // code to call wizard-generate code
        public Workflow1() {
            InitializeComponent();
        }

        // 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
    }
}
```

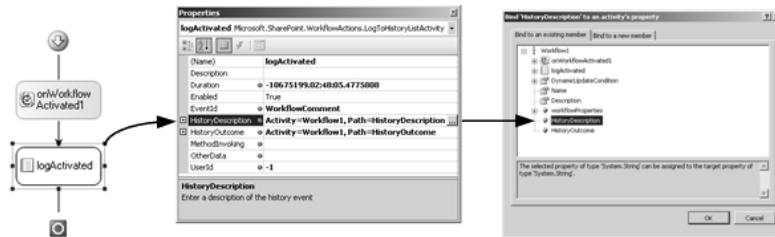
## SharePoint Activity Library

- WSS-specific activities used to create SharePoint WF Programs



## Data Bound Properties

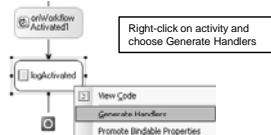
- WF supports data binding of properties
  - Allows for declarative flow of data between activities
  - Used extensively for creating SharePoint WF programs



```
public partial class Workflow1 : SharePoint.SequentialWorkflowActivity {
    // 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 : SharePoint.SequentialWorkflowActivity {
    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

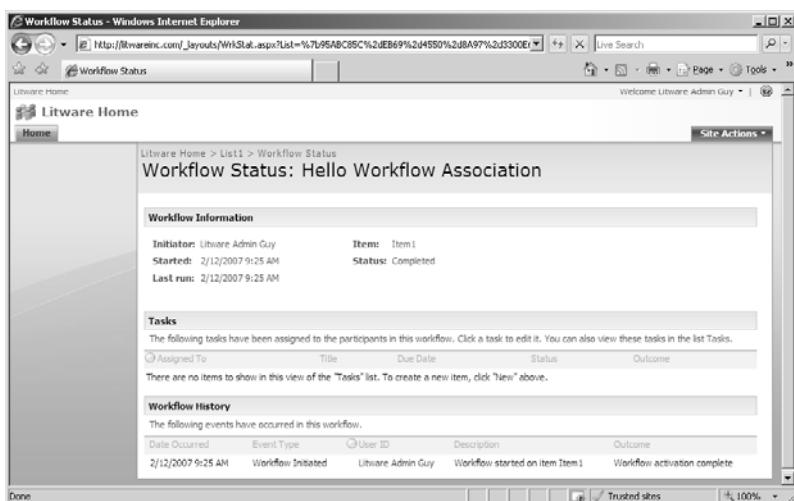
```
<Feature  
  Id="0CEED7AE-D327-41ad-BC33-B3F3F8A4DAD2"  
  Title="Hello World Workflow Template Feature"  
  Description="This feature installs our Hello World Workflow Template"  
  Version="12.0.0.0"  
  Scope="Site"  
  xmlns="http://schemas.microsoft.com/sharepoint/">\n  
  <ElementManifests>  
    <ElementManifest Location="workflow.xml" />  
  </ElementManifests>\n  
</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/">\n<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]">\n  
  <Categories/><!-- no categories needed -->\n  <MetaData /><!-- no metadata needed -->\n</Workflow>\n</Elements>
```

## Testing 'Hello World' Workflow Template

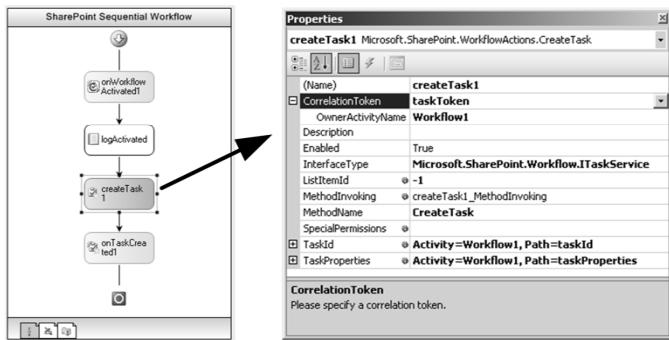


## Creating and Waiting on Tasks

- 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

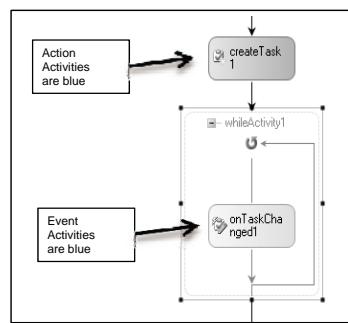
## Task GUIDs and Correlation Tokens

- 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 event
  - Their event handlers run after the extent has occurred



## Initializing a New Task

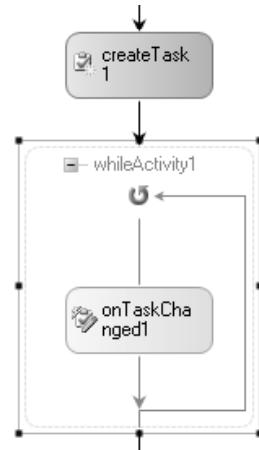
- Add event handler behind CreateTask activity
  - This event handlers fires before task creation
  - Gives you a chance to initialize task properties

```
// 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);
}
```

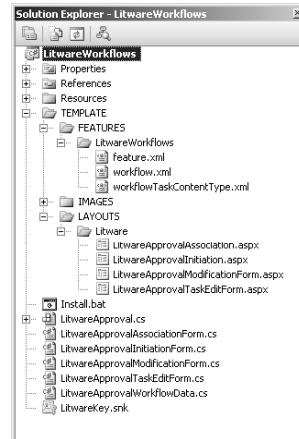
## Waiting on a Task

- 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



## Integrating Workflow Input Forms

- Workflow Input Form Types
  - Association Form
  - Initiation Forms
  - Modification Forms
  - Task Edit Form
- Sample Project
  - LitwareWorkflows



## Adding Forms to a Workflow Template

```
<Workflow
  Id="4797A6A6-4F31-40ca-9814-746402C2DB56"
  Name="Litware Approval"
  Description="Sample workflow template demonstrating workflow input forms"
  CodeBasisClass="LitwareWorkflows.LitwareApproval"
  CodeBasisAssembly="LitwareWorkflows, [full 4-part assembly name]"
  AssociationUrl="_layouts/Litware/LitwareApprovalAssociation.aspx"
  InstantiationUrl="_layouts/Litware/LitwareApprovalInitiation.aspx"
  ModificationUrl="_layouts/Litware/LitwareApprovalModificationForm.aspx"
  TaskListContentTypeId="0x0108010084565D92BEFE4a75A28C2F658B7BECCA"
>

<MetaData>
  <ModificationInstanceWithCustomLitwareForm>
    Modify workflow instance with a custom Litware form
  </ModificationInstanceWithCustomLitwareForm>
</MetaData>

<Categories/>

</Workflow>
```

## The Association Form

The screenshot shows the 'Create Workflow Association' page. At the top, it says 'Create Litware Workflow Association 'Legal Approval''. Below that, there's a note: 'Click OK to create a new Litware workflow association named Legal Approval'. The 'Default Approver' field contains 'Brian Cox'. Under 'Default Approval Scope', the radio button 'For Internal Litware Use Only' is selected. In the 'Instructions to Approver' section, a text area contains: 'Please indicate whether this item or document meets the approval of the Litware Legal Department for customer-facing content.' At the bottom right are 'OK' and 'Cancel' buttons.

## The Initiation Form

The screenshot shows the 'Start New Workflow Instance' page. At the top, it says 'Start New Litware Approval Workflow Instance on Item1'. Below that, there's a note: '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'. The 'Approver' field contains 'Brian Cox'. Under 'Approval Scope', the radio button 'For Internal Litware Use Only' is selected. In the 'Instructions to Approver' section, a text area contains: '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.' At the bottom right are 'OK' and 'Cancel' buttons. At the very bottom left is a 'Done' link.

## Invoking the Modification Form

The screenshot shows the 'Workflow Status' page in a browser window. The title bar says 'Workflow Status - Windows Internet Explorer'. The main content area is titled 'Workflow Status: Legal Approval'. It displays 'Workflow Information' with initiator 'Angela.Barbarol', item 'Item1', status 'InProgress', and last run '2/13/2007 8:39 AM'. Below this is a 'Tasks' section with one task assigned to 'Brian.Cox' with title 'Approval required for Item1' and due date '2/15/2007'. A 'Workflow History' section shows two events: 'Workflow Initiated' by 'Angela Barabroli' and 'Task Created' by 'Angela Barabroli'. A callout bubble points to the 'Modify this workflow with custom Litware form' link at the bottom of the 'Workflow Information' section.

## The Task Edit Form

The screenshot shows the 'Approve or Reject Item' page in a browser window. The title bar says 'Approve or Reject Item - Windows Internet Explorer'. The main content area is titled 'Approve or Reject Item1'. It has sections for 'Item Requiring Approval' (listing item 'Item1', list 'List1', and site 'http://litwareinc.com'), 'Instructions to Approver' (with a note about reviewing the item), and 'Approver Comments' (with a note about copyright). A callout bubble points to the 'Approver Comments' section. 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

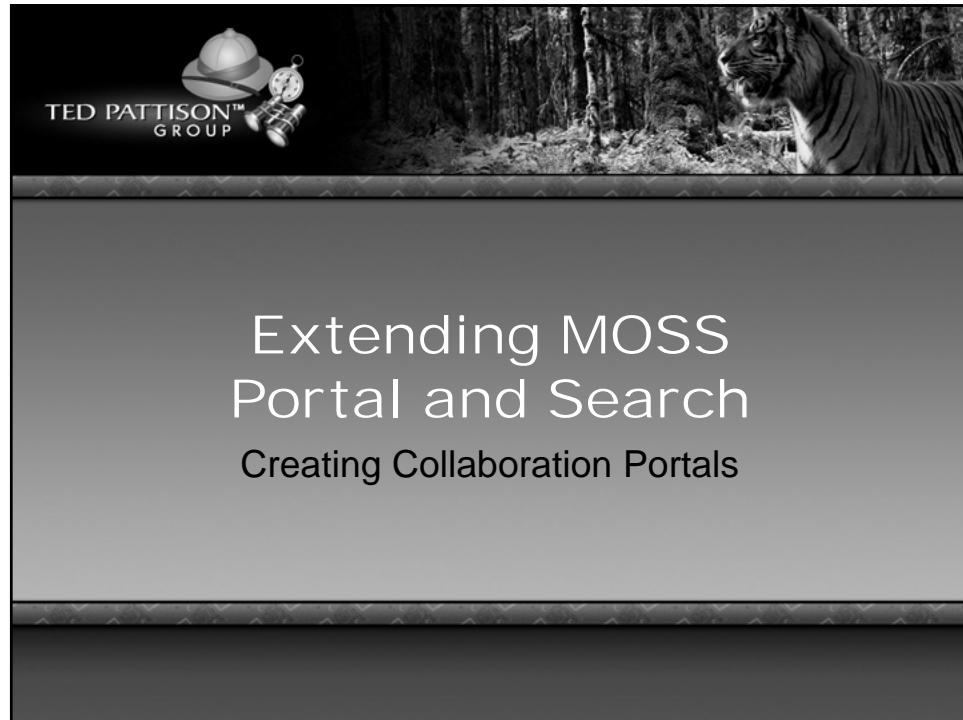
## InfoPath Workflow Form Intergration

```
<Elements xml ns="http://schemas.microsoft.com/sharepoint/">
<Workflow
  Name="ListwareWorkflowInfoPath"
  Description="Simple workflow with InfoPath forms."
  Id="48500BEB-D1BE-4ec4-8D21-5DEF76BEEDA8"
  CodeBesideClass="ListwareWorkflowInfoPath.Workflow1"
  CodeBesideAssembly="ListwareWorkflowInfoPath, [full assembly name]"
  TaskListContentTypeId="0x01080100C9C9515DE4E24001905074F980F93160"
  AssociationUrl="_layouts/CstWrkflIP.aspx"
  InstantiationUrl="_layouts/LniWrkflIP.aspx"
  ModelUrl="._layouts/ModWrkflIP.aspx">

  <standard MOSS task content type>
  <standard MOSS application pages>
  <urn:schemas-microsoft-com:office:infopath:ReviewInitiatonForm2:-myXSD-2005-11-22T23-49-53>
</Workflow>
</Elements>
```

## 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

## Site Content and Structure

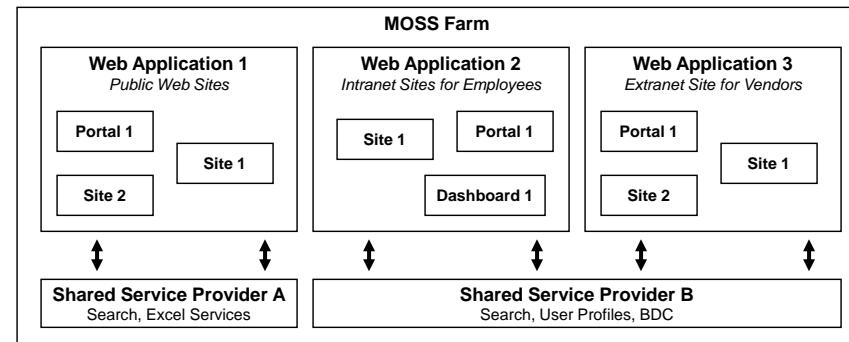
Type	Name	Modified	Modified By	Checked Out To	Contact	Page Layout
	Default (default)	2/3/2007 3:17 AM	Litware Admin Guy			Welcome page with Web Part zones

## 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)

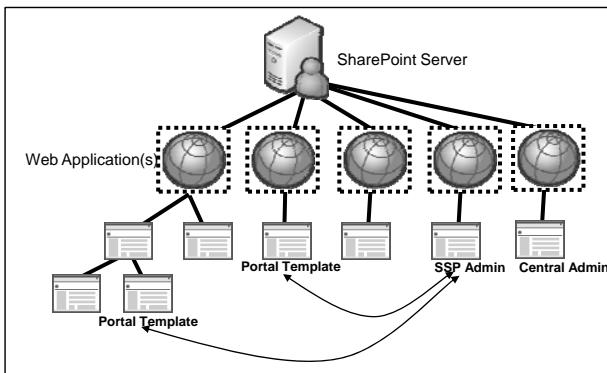
## SSP Architecture

- 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



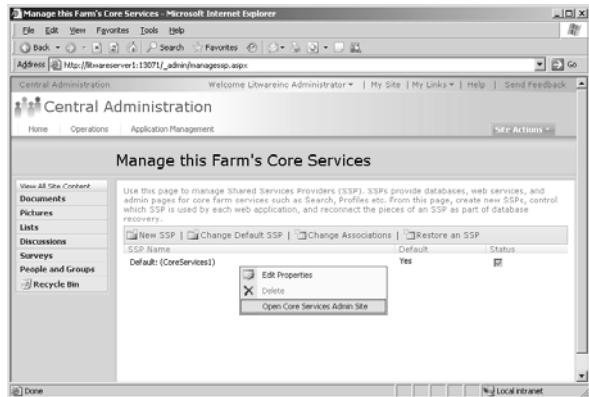
## So what is a Portal, Really?

- 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



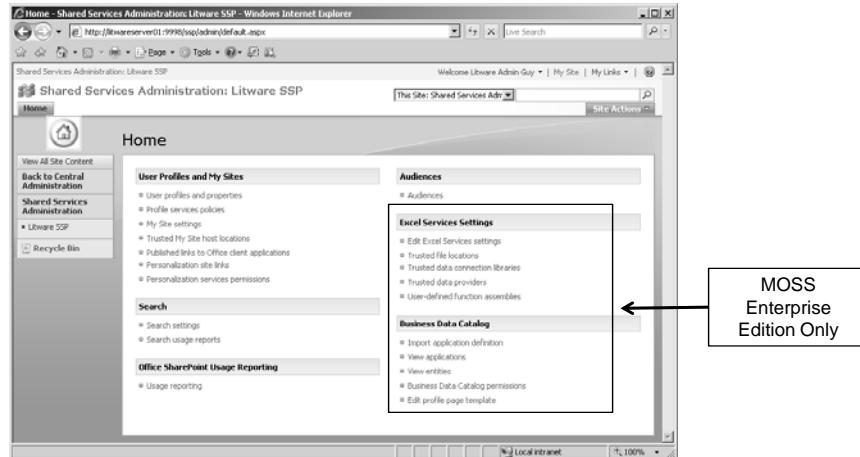
## Creating An SSP

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



## SSP Administration

- Functionality of SSP Broken out into Sections



## 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. It includes sections for 'Profile and Import Settings' and 'User Profile Properties'. The 'Profile and Import Settings' section displays various configuration details like source of user sets, import status, and import schedule. The 'User Profile Properties' section shows the number of user profile properties and mapped properties.

## Viewing User Profiles

- User profiles can be added and deleted

The screenshot shows the 'View User Profiles' page. It lists 8 user profiles with checkboxes next to each account name. The columns are 'Account name', 'Preferred name', and 'E-mail address'. The users listed are LITWAREINC\Administrator, LITWAREINC\Bart, LITWAREINC\Carol, LitwareInc\Mike, LITWAREINC\Sean, LITWAREINC\Ted, and LITWAREINC\Tim.

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\Ted	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



## Updating A User Profile

A screenshot of a user's profile page in a SharePoint portal. The browser window title is "Brian Cox - Windows Internet Explorer". The address bar shows the URL: http://litwareinc.com:81/\_layouts/Person.aspx?accountname=LTWAREINC\BrianC. The page title is "My Site". The main content area displays the user's profile information: "Brian Cox" (Job Title: Solutions Developer, IT), "N1003 (425) 595-0129" (Work phone), and "BrianC@LitwareInc.com" (Work e-mail). It also shows the "Organization Hierarchy" with Luis Bonifaz as the manager. The "Details" section lists responsibilities: Sales, Marketing. The "Contact Information" section includes a work phone number and email. The "Documents" section shows a list of documents, and the "Colleagues" section lists David Yalovskiy, Luis Bonifaz, and Sandeep Kalyal. The "Memberships" section shows the user is part of the "Litware FTE" distribution list. The "Links" section indicates no links have been shared by the user.

## Colleagues

My Colleagues - Windows Internet Explorer  
http://litwareinc.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
General			
David Yalovsky	Everyone	Yes	General
Luis Bonifaz	Everyone	Yes	General
Sandeep Katyal	Everyone	Yes	General

Edit | Delete

Done Trusted sites 100%

## 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

Site Settings Manage Audiences View Audiences

**View Audience Properties: Sales Folks**

Use this page to view and edit the properties of this audience.

**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

Edit audience    View membership

**Audience Rules**

Use these links to add, edit, and delete audience rules.

Operand	Operator	Value
Department	Contains	Sales

Add rule

## Compiled Audience

- Compiled Audience defines list of users

The screenshot shows a Microsoft Internet Explorer window titled "View Audience Membership: Sales Folks". The page content includes:

- Last compiled: 12/13/2005 5:03 PM
- Number of members: 3
- A table listing members:

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

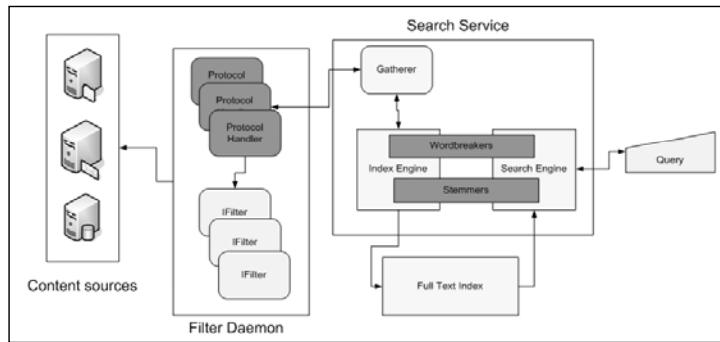


## WSS Verses Office Server Search

- 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



## Configuring Search

Configure Search Settings

**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

Use this page to view and manage search scopes. The order in which the search scopes appear in this list is the order in which they will appear in the search scope list next to the Search box.

Title	Update Status	Items
Shared (3)		
People	Ready	84
All Sites	Ready	1280
My New Search Scope (SSP Level)	Empty - Add rules	empty
http://litwareserver01:9998/_sp/admin (0)		
http://litwareinc.com/sites/LitwarePublishingSite (0)		
http://litwareinc.com:81/ (0)		
http://litwareinc.com:81/personal/brianc (0)		
http://litwareinc.com:81/personal/administrator (0)		
http://litwareinc.com/sites/LitwarePortalSite (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> Scope rules define what is in or not in a scope. Use different types of rules to match items in various ways	<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. Items matching the property query will be added to the scope. To include 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> Decide how this rule should be applied to the overall scope. The scope-wide filter is used when combining the items matching all rules to determine what is in the scopes overall.	<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

OK Cancel

## 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

## Customize UI With XSLT

- Appropriate for scenarios requiring
  - Change results layout, look and feel
  - Pivoting using the keyword syntax
- Web part property
- Power of XSLT
  - Formatting
  - Logic
  - Math

## Extending Search

- Customizing the Query and Results
  - Query Object Model
- Customizing the Index
  - Index custom data - Protocol Handlers, IFilters, BDC
  - Custom Query Time Security Trimmer

## Agenda

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



**TED PATTISON™  
GROUP**

**Web Content Management  
using Publishing Portals**

Creating and managing content on  
Internet-facing sites with SharePoint Server 2007

## Agenda

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

## 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

## MOSS WCM Design Goals

- Integrate CMS and SharePoint
  - Remove forced choice “CMS vs. SharePoint”
  - Move towards integrated ECM solution
  - Add Internet readiness to Sharepoint
- Make creation of dynamic, produced websites dramatically faster and easier
  - Lower amount of costly custom code
  - Provide great OOB experience
  - Improve user experience across all features

## Creating A Publishing Portal

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

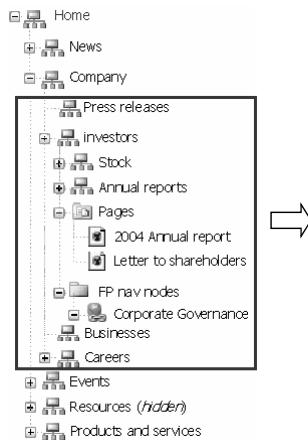
The screenshot shows the 'Select a template' dialog box in Microsoft Internet Explorer. The 'Publishing' tab is selected, and 'Publishing Portal' is chosen. An arrow points from this dialog to a screenshot of the 'AdventureWorks' site collection homepage. The homepage features a banner with five people and several links: 'Enable anonymous access', 'Manage navigation', 'Go to master page gallery', 'Change the page header and master page of this site collection', 'Manage content and structure', 'Use the translation feature to manage multi-lingual sites and pages', and 'Users in the Approvers group can publish pages, images, and documents in this site. Users in the Reviewers group can publish pages, images, and documents. Authors is enabled for this site, and content approval is enabled for the documents and images branches.'

## 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

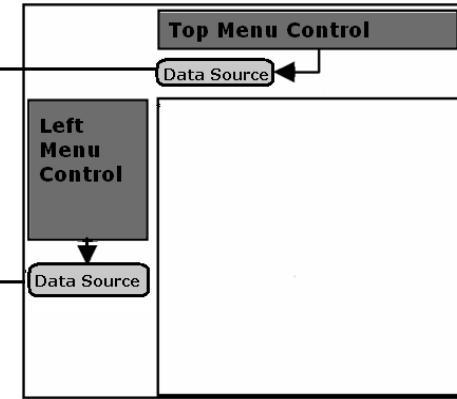
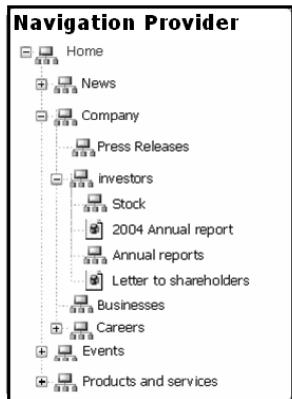
The screenshot shows a SharePoint 2007 site with a navigation bar at the top. The 'Create Page' option in the context menu is highlighted with a red box. The menu also includes 'Edit Page', 'View All Site Content', 'View Reports', 'Site Settings', and 'Manage Content and Structure'. Below the menu, there is a note about enabling anonymous access.

## 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 supports runtime object caching

## Site Content and Structure

Litware Internet Site

Site Content and Structure

Pages - All Documents

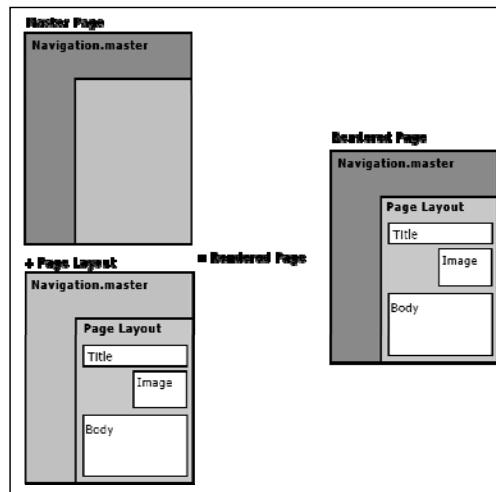
Type	Name	Modified	Modified By	Approval Status	Scheduling Start Date	Scheduling End Date	Contact	Page Layout
default (default)	4/23/2006 1:18 PM	System Account	Approved					/catalogs/masterpage/Welcome

Resources Related To 'default.aspx'

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

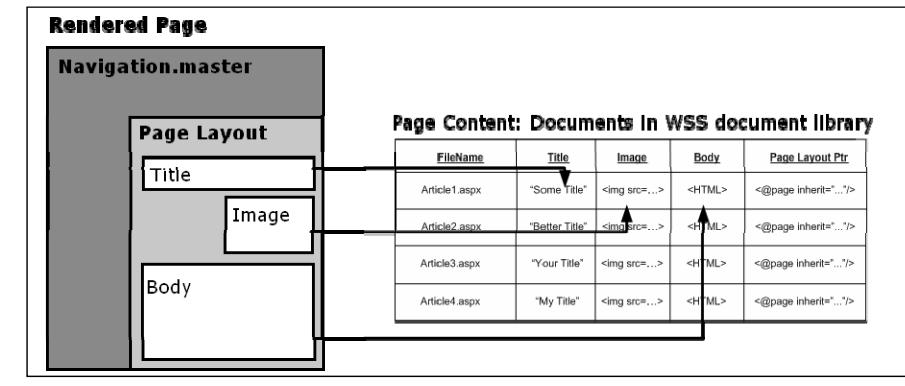


## Page Layouts

- 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

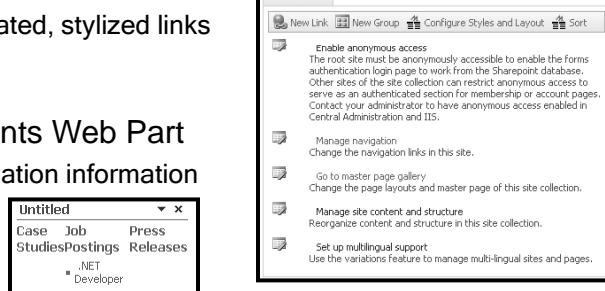
- Create shared columns
- Create content type
- Add created site columns to content type
- In the Master Page Gallery
  - Create new Page Layout file
  - Check-out file and edit in SharePoint Designer
  - Populate the file with content fields
  - Check-in and approve
- Use the new page layout file

## 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

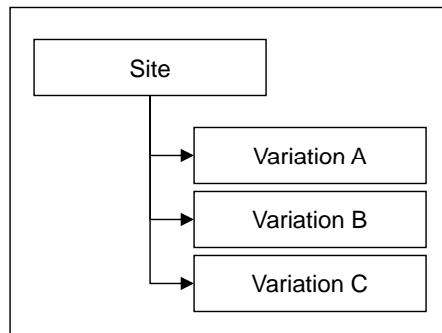


## Multilingual sites

- 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.

<b>Output Cache</b> Select the <b>Enable output cache</b> check box to enable output caching in this site collection.	<input checked="" type="checkbox"/> Enable output cache
<b>Default Page Output Cache Profile</b> 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 <input type="button" value="Public Internet (Purely Anonymous)"/>
	Optimized for public Internet facing sites or areas that are meant to serve the same content to all users. No authentication check is done and any user requesting a page receives the same page as any other user.
	Authenticated Cache Profile <input type="button" value="Disabled"/>
	Caching is not enabled
<b>Page Output Cache Policy</b> You can allow administrators and page layout designers to choose a different page output cache profile from the profile specified above.	Publishing Sites: <input type="checkbox"/> Publishing sites can use a different page output cache profile Page Layouts: <input type="checkbox"/> Page layouts can use a different page output cache profile

## Configuring Document Conversion

Central Administration > Operations > Services on Server

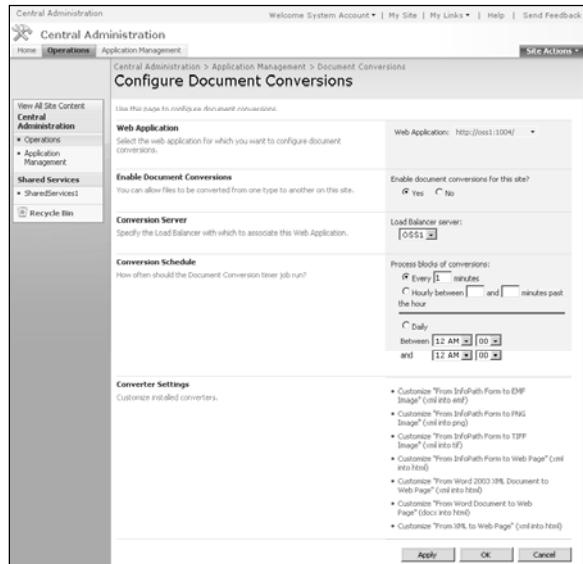
### Services on Server: OSS1

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

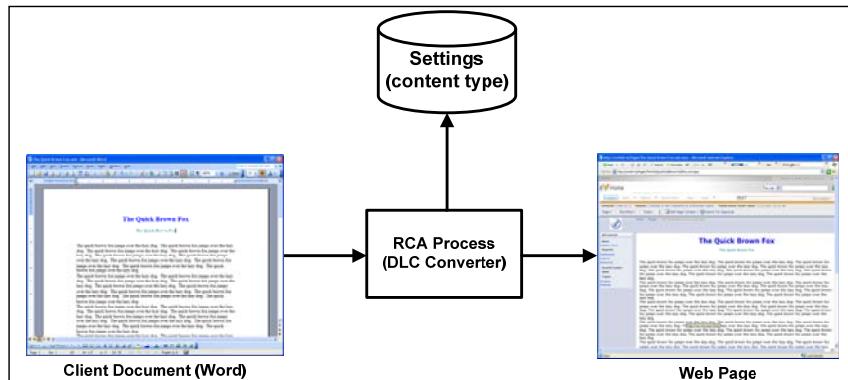
Service	Description	Server:	Action
Central Administration	All services run on this server	OSS1	Started
Document Conversions Launcher Service	Web Application and Search Query services run on this server	OSS1	Stopped
Document Conversions Load Balancer Service	Search Indexing service runs on this server	OSS1	Stopped
Excel Calculation Services	Excel Calculation service runs on this server	OSS1	Started
Office SharePoint Server 2007 (Beta) Search Service [Index: On; Query: On]	Services you choose run on this server	OSS1	Started
Windows SharePoint Services Administration		OSS1	Started
Windows SharePoint Services Incoming E-Mail		OSS1	Started
Windows SharePoint Services Search Service		OSS1	Started
Windows SharePoint Services Web Application		OSS1	Started

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



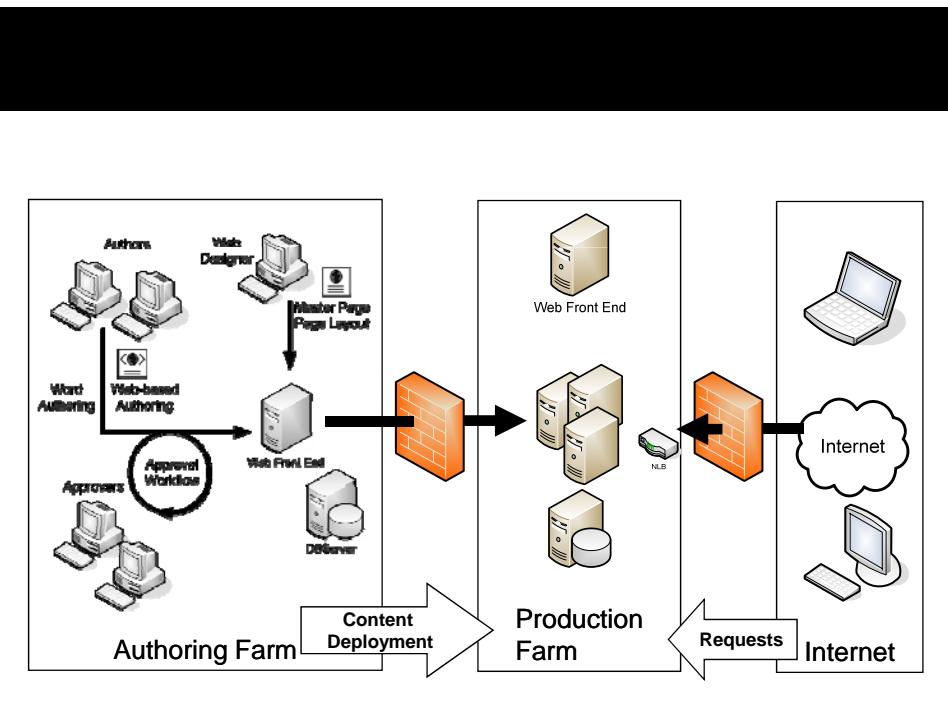
## Smart Client Authoring



- 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
- Optimization through Caching Profiles
- Content Translation using Variations
- Converting Office documents
- Paths and Deployment



# Leveraging the Business Data Catalog

Surfacing backend data in SharePoint Server sites

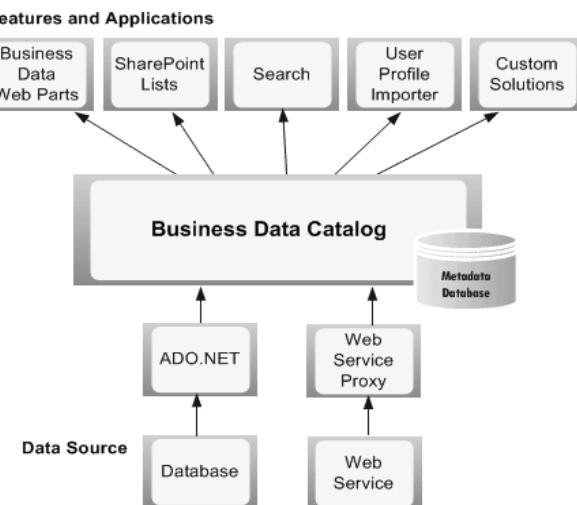
## Agenda

- 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

## Design Motivations

- Reduce the need for custom front-end code
  - Reuse SOA investments already in place
- Create manageable method for reusing data
  - Centralized deployment
  - Centralized data security
  - Low latency
- Designed for portal & collaboration scenarios
  - Data query, indexing, personalization
- The BDC is not about: transactions, workflow, data transformation, adapters, precomposition

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

## Tools for Building Definitions

- BDC Meta Man
  - <http://www.bdcmetaman.com/>
  - Free version for SQL Server data sources
  - Licensed version adds Web service, Oracle, ODBC
- Action builder
- Visual designer
- Multiple entity support
- Configuration of LOBSystems, LOBSystemInstances, Entities, Methods, etc.
- Simego MOSS BDC Design Studio
  - <http://www.simego.com/#MOSS>
- Various efforts on CodePlex, GotDotNet
- Some SDK-delivered functionality is in the works

## Importing Application Definition Files

Shared Services Administration: SharedServices1

Welcome LITWAREINC\administrator | My Site | My Links | Site Actions

This Site: Shared Services Admin

Home

View All Site Content

Back to Central Administration

Shared Services Administration

- SharedServices1
- Recycle Bin

SharedServices1 > Import Application Definition

### Import Application Definition

Application Definition

An application definition describes a database or web service. It includes connection settings, authentication mode, definitions of available entities, and security settings. After you upload an application definition to the Business Data Catalog, you can use its entities in lists, web parts, search, user profiles and custom applications.

Uploading a new application definition will overwrite any existing settings and changes you have made to the application. This includes security settings and actions. Before uploading a new application definition, export the current state of your application definition.

Choose an application definition file and click Import.

Learn about writing application definition files

**File Type**

Choose the type of application definition file to import:

Model

Resource

**Resources to import**

Choose resources to import:

Localized Names

Properties

Permissions

Import Cancel

## Administration of BDC Applications

Shared Services Administration: SharedServices1

Welcome LITWAREINC\administrator | My Site | My Links | Site Actions

This Site: Shared Services Admin

Home

View All Site Content

Back to Central Administration

Shared Services Administration

- SharedServices1
- Recycle Bin

SharedServices1 > Business Data Catalog Applications > AdventureWorks\_HR

### View Application: AdventureWorks\_HR

**Application Information**

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

Manage Permissions

Export Application Definition

Delete Application

**Entities**

Name	Employee
Department	

## Examining BDC Application Entities

The screenshot shows the Entity Information page for the Employee entity. The page includes:

- Entity Information:** Name: Employee, Application: AdventureWorks\_HR, Createable: No.
- Fields (of default view):** A table showing fields like Name, Email, Employee ID, First Name, Job Title, Last Name, and Phone, each with its type (e.g., System.String) and display by default status.
- Relationships:** A table showing relationships like DepartmentToEmployees (Related To: Department).
- Actions:** A table showing actions like View Profile (Type: Default, URL: http://[mess]:1661/[sp]/admn/Content/Employee.aspx?EmployeeID={0}).
- Filters (of finder method):** A table showing filters like EmployeeID (Type: ComparisonFilter, ValueOrFilter).

## Adding Actions to an Entity

The screenshot shows the Add Action dialog for the Employee entity. The dialog includes:

- Name:** Action Name: Open data in custom page.
- URL:** Navigate to this URL: [http://example.com/edit.aspx?id={0}]. Example: http://example.com/edit.aspx?id={0}.
- URL Parameters:** Parameter Property: EmployeeID, Add Parameter button.
- Icon:** Choose an icon to display next to the action. Options: No icon (selected), Standard icon (edit icon), or The image at this URL: [ ].

## Administrating Security

The screenshot shows the 'Manage Permissions' page for the 'AdventureWorks\_HR' application under 'Business Data Catalog Applications'. The page title is 'Manage Permissions: AdventureWorks\_HR'. It displays a list of users and their rights:

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

## Using BDC Web Parts

The screenshot shows two windows related to BDC Web Parts:

- Add Web Parts - Web Page Dialog:** A dialog box titled 'Add Web Parts to Left' containing a list of available web parts. One item, 'Business Data Actions', is selected.
- Department List:** A configuration dialog box for the 'Business Data Actions' web part. It shows a table with three rows: 1, 16, and Engineering. The 'Engineering' row is highlighted.

## Editing BDC Web Parts

**Business Data Related List**

Type: Employee (HR)

Relationship: DepartmentToEmployees

Display toolbar:

Display animation while loading:

Data View Properties:

- XSL Editor: To add XSL, click XSL Editor...
- Parameters Editor: To change parameters, click Parameters Editor...
- Advanced Parameters Editor...

OK Cancel Apply

**Edit View**

Items to Reference: This section allows you to choose from Human Resources Data and display in the web part. You can allow users to choose by selecting Reference items managed by user profile properties. You can also choose to reference items managed by the current user's profile or items that meet certain criteria.

Reference items managed by user profile properties of the user viewing this page:

- Allow user to change items:
- Profile items (or all items):
- A value:

Reference items that meet these criteria:

- Allow user to change items:
- Profile items (or all items):
- A value:

A user profile property of the user viewing this page:

Add

## Connecting Web Parts with Associations

Employee ID	First Name	Last Name	Job Title	Department
2	Kevin	Dixon	Marketing Assistant	Marketing
6	David	Bradley	Marketing Manager	Marketing
46	Saria	Harnpadoungsataya	Marketing Specialist	Marketing
106	Mary	Gibson	Marketing Specialist	Marketing
119	All	Williams	Marketing Specialist	Marketing
203	Terry	Eminhizer	Marketing Specialist	Marketing
269	Wanda	Benshoof	Marketing Assistant	Marketing
271	John	Wood	Marketing Specialist	Marketing
272	Mary	Dempsey	Marketing Assistant	Marketing

## Searching through BDC Applications

Shared Services Administration: SharedServices1 > Search Settings > Content Sources > Add Content Source  
**Add Content Source**

Use this page to add a content source.  
 \* Indicates a required field

**Name**  
 Type a name to describe this content source.  
 Name: \*

**Content Source Type**  
 Select what type of content will be crawled.  
 Note: This cannot be changed after this content source is created since other settings depend on it.

SharePoint Sites  
 Web Sites  
 File Shares  
 Exchange Public Folders  
 Business Data

**Applications**  
 Select whether to crawl all applications in the Business Data Catalog, or include only selected applications in this content source.  
 An application may be included in only one Business Data content source.

Crawl entire Business Data Catalog  
 Crawl selected applications  
 AdventureWorks [B]

**Crawl Schedules**  
 Select the crawl schedules for this content source.

Select the schedule that this should be a part of:  
 Full Crawl:  Create schedule  
 Incremental Crawl:  Create schedule

**Start Full Crawl**  
 Select "Start full crawl of this content source" and click "OK" to start a full crawl of this content source.  
 Start full crawl of this content source

## Adding BDC Columns to WSS Lists

Create Column: Colleagues

Use this page to add a column to this list.

**Name and Type**  
 Type a name for this column, and select the type of information you want to store in the column.

Column name:

The type of information in this column is:

Single line of text  
 Multiple lines of text  
 Choice (menu to choose from)  
 URL (1, 10, 100)  
 Currency (\$, .00)  
 Date and time  
 Look-up (information already on this site)  
 Yes/No (check box)  
 Person or Group  
 Hyperlink or Picture  
 Calculated (calculation based on other columns)  
 Business data

**Additional Column Settings**  
 Specify detailed options for the type of information you selected.

Description:

Require that this column contains information:  
 Yes  No

Type:

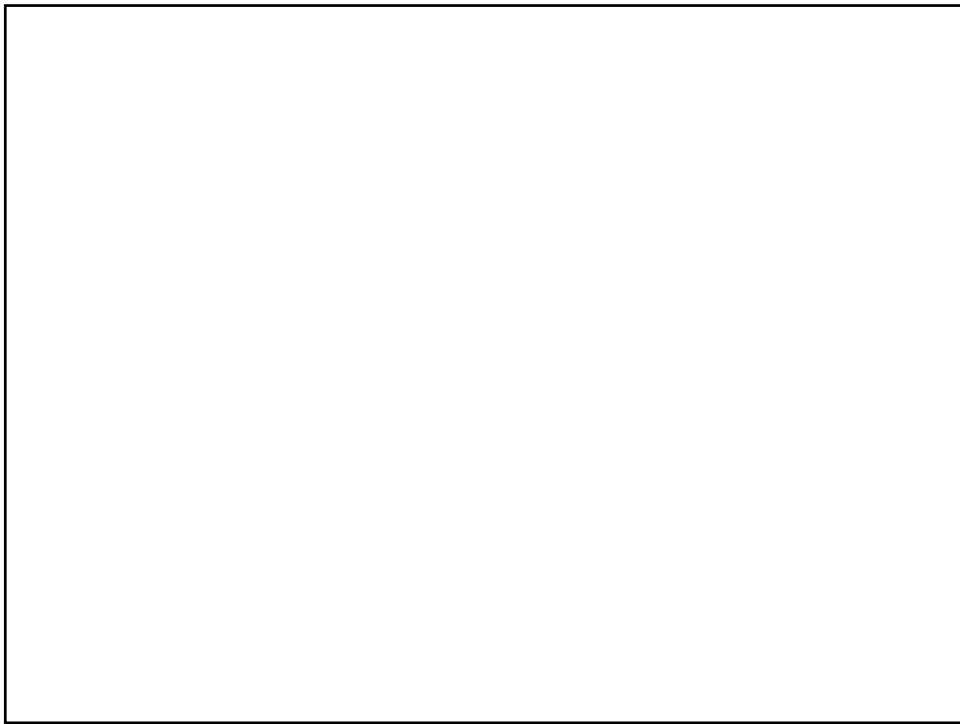
Display this field of the selected type:

Display the actions menu  
 Link this column to the profile page

Add a column to show each of these additional fields:  
 Department  
 Department ID  
 Add to default view

## 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





**Excel Services  
and Report Center**

Leveraging the BI features  
of Office SharePoint Server

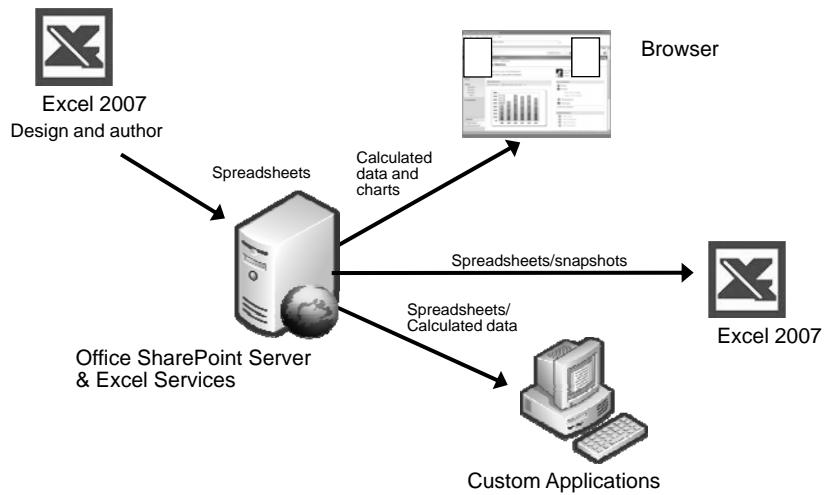
## Agenda

- 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

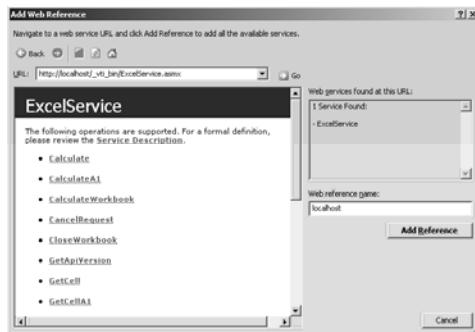


## Excel 2007 Web Services

- 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

- ⦿ E.g.:

```
=MyUDF(A1, 17, B3:B13)
```

- ⦿ Scenarios

- ⦿ Extending Excel's calculation on the server
  - ⦿ Connecting to custom data sources

## Report Center

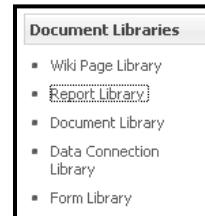
- With Office 2007, SharePoint Server become the hub for BI on the server
  - Excel Services and Reporting Services in the portal
  - Out-of-the-box BI portal experience
  - Dashboards, KPIs, and Report Libraries
  - Integrated with Portal, Collaboration, Enterprise Content Management and Workflow functionality

## 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	2002	2003	2004
	\$1,301,091.29	\$4,147,221.44	\$4,071,471.21
Linda C. Morris	\$2,800,029.13	\$4,491,867.70	\$3,317,413.63
Allen Clegg	\$1,301,091.29	\$4,147,221.44	\$4,071,471.21
Osman F. Yigit	\$1,115,639.26	\$1,400,136.03	\$1,764,938.89
Tara Michael Potts	\$1,301,091.29	\$4,147,221.44	\$4,071,471.21
Frankie Lorraine Wohl	\$1,471,074.05	\$865,361.20	\$1,014,481.00
Sita E. Yo	\$2,046,111.62	\$1,370,530.04	\$1,015,721.49
Joan Elizabeth Smith	\$2,532,300.00	\$1,481,793.34	\$1,188,336.23
David R. Campbell	\$1,241,200.77	\$1,371,410.33	\$1,936,081.56
Lee A. Price	\$1,301,091.29	\$1,400,136.03	\$1,764,938.89
Renee F. Valery Chisholm	\$1,471,074.05	\$865,361.20	\$1,014,481.00
Tina A. Mousa-Amin	\$1,301,091.29	\$1,400,136.03	\$1,764,938.89
Rebekah Valdez			\$1,241,304.04
Lynn H. Tiefbar			\$1,758,381.83

## 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





# Application Security

Securing Your SharePoint Business Solutions

## Agenda

- User Authentication
  - Windows Authentication
  - Forms Authentication
- WSS Identities and Security Contexts
  - Application Pool Identity
  - The SHAREPOINT\System account
  - Escalation of Privilege
  - Delegating User Credentials
- The MOSS Single Sign-On Service (SSO)

## Security Concepts 101

- Authentication
  - Verifying identity whom someone claims to be
  - based on using credentials to verify identity
- Authorization
  - Process of determining what an identity has access to
  - Takes place after authentication

## WSS Identities

- WSS Application Pool Identity
  - Configured with IIS or WSS admin tools
  - Authenticated when worker process is launched
- WSS System Identity
  - Used by WSS to hide application pool identity
- User Identity
  - Authenticated by Windows or another authentication provider
  - Used for authorization and auditing

## WSS Worker Process 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



## 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
- SQL Server must authenticate WSS identity
  - Can be configured with Integrated Windows Authentication (best)
  - Can be configured with standard SQL authentication
- WSS accesses SQL Server using App Pool Identity
  - When using Integrated Window Security, SQL Server authenticated WSS using the Application Pool Identity
  - WSS Identity must be in SQL Server roles of Database Creators and Security Administrators

## The WSS System Account

- Some WSS operations require more permissions than user has
  - WSS will sometimes run code under IIS app pool identity
  - IIS app pool identity has full control to WSS content database
  - Exposing IIS app pool identity poses security risks
  - WSS "V3" introduces new WSS System Account
- System account has a login name is SHAREPOINT\system
  - SID as S-1-0-0 (Null SID)
  - User id as 1073741823 (0x3FFFFFFF)
  - When a list item is created by the application pool identity, it will show as "created by SHAREPOINT\system"

## WSS User Authentication Options

- 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

## Authentication And WSS Zones

- WSS partitions user authentication into zones
  - 1 Zone = 1 WSS-extended Web Application
  - Each zone based on its own IIS Web site
  - Each zone has its own web.config file
- Zone can have only one authentication provider
  - Choose between (1) Windows, (2) FormsAuth or (3) WebSSO
- Can WSS site support multiple authentication types?
  - Yes, two WSS Web Applications can point to same content
  - More on this later this lecture

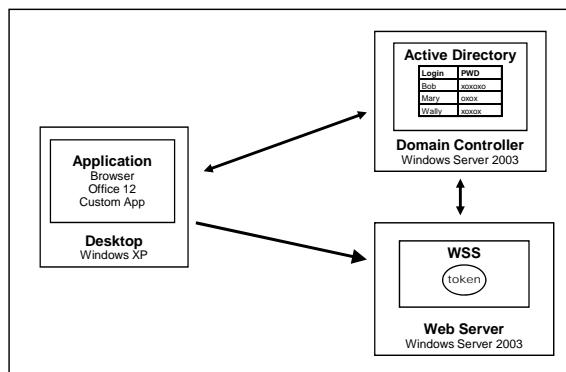
## Configuring Windows Authentication

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



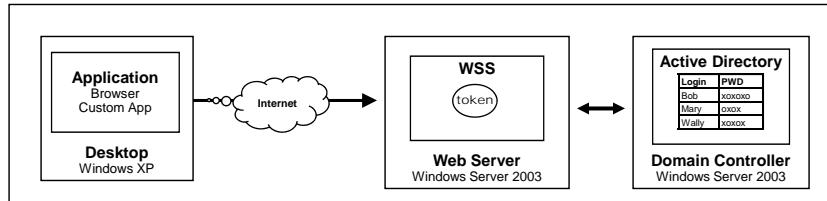
## Integrated Windows Authentication

- 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, firewall-friendly protocol for authentication



- No need for client to access Windows domain controller
- Authenticates to Windows account and creates security tokenUser name and password passed in clear text
- Must use HTTPS for any reliable level of security

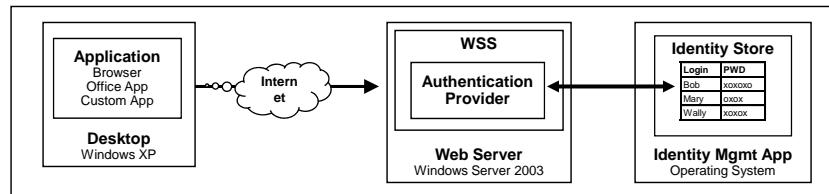
## Windows Authentication Zones

- 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>
```

## ASP.NET Forms Authentication

- WSS V3 supports ASP.NET forms authentication
  - Allows you authenticate without requiring Active Directory
  - Based on ASP.NET 2.0 pluggable authentication providers
  - ASP.NET 2.0 role providers can optionally be used as well



- Out-of-the-box Authentication providers
  - ASP.NET 2.0 SQL Server authentication provider
  - LDAP Authentication provider

## WSS-supplied Configuration UI

Edit Authentication

<b>Zone</b> These authentication settings are bound to the following zone.	Zone Default
<b>Authentication Type</b> Choose the type of authentication you want to use for this zone.	Authentication Type <input checked="" type="radio"/> Windows <input checked="" type="radio"/> Forms <input type="radio"/> Web single sign on
<b>Anonymous Access</b> You can enable anonymous access for sites on this server or disallow anonymous access for all sites. Enabling anonymous access allows site administrators to turn anonymous access on. Disabling anonymous access blocks anonymous users in the web.config file for this zone.	<input checked="" type="checkbox"/> Enable anonymous access
<b>Membership Provider Settings</b> Enter the configuration settings for the membership provider. The settings specified will be copied into the web.config file for this zone.	Membership provider configuration <membership defaultProvider="AspNetSqlMembershipProvider"/>
<b>Role Manager Settings</b> Enter the configuration settings for the role manager. The settings specified will be copied into the web.config file for this zone.	Role manager configuration

Save Cancel

## Configuring Forms Authentication

- Forms Authentication requires anonymous access
  - All users run under identity of single Windows account
  - Identity configurable within IIS administration tools



## 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)

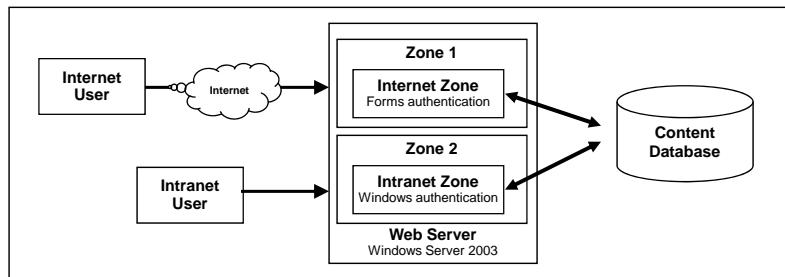
```
<!-- selected snippets from web.config for Forms auth -->
<configuration>
  <system.web>
    <!-- use Forms Authentication -->
    <authentication mode="Forms">
      <forms loginUrl="/_layouts/login.aspx" />
    </authentication>
    <!-- configure membership provider -->
    <membership defaultProvider="AspNetSqlMembershipProvider"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

## Forms Authentication By Smart Clients

- A truly clever smart client could
  - Recognize a HTTP redirect request
  - Open a browser window for the login form
  - Submit the form content back to the server
- But
  - It can't confidently know that the redirect request is for a logon form URL
- Therefore
  - Users must first visit the site with a browser
  - Server must issue a (short-lived) persistent cookie
- This affects all Office System client applications

## Multiple Authentication Types

- Two WSS Web Application can access same sites
  - Configure Web Applications to point to same content database
  - Makes it possible to support multiple authentication providers



- An important issue with access control configuration
  - ACLs will mix of users from different authentication systems
  - Authorization for users using 2 zones must be configured twice

## Authorization and Access Control

- Authorization involves granting rights to identities
  - Rights what users can and cannot do in a site
- Abstractions make configuring authorization easier
  - Users and Groups from authentication provider
  - WSS Groups
  - WSS Permission Levels (a.k.a. roles)

## Sites and Site Collections

- Site Collection is atomic unit of security configuration
  - Security configuration is not shared across site collections
- With a site collection...
  - Site collection owners have full administrative rights
  - Site collection administrators have full administrative rights
  - Rights for other users configured on per site basis

## Permissions Managed Using Rights

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

List Rights
<input checked="" type="checkbox"/> Manage Lists - Add or remove columns in a list, and add or remove public views of a list.
<input checked="" type="checkbox"/> Cancel Check-Out - Check-in a document without saving the current changes.
<input checked="" type="checkbox"/> Add Items - Add items to lists, add documents to document libraries, add Web discussion comments.
<input checked="" type="checkbox"/> Edit Items - Edit items in lists, edit documents in document libraries, edit Web discussion comments in documents, and customize Web Part Pages in document libraries.
<input checked="" type="checkbox"/> Delete Items - 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.

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

## 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 under 'Site Settings'. It displays a table with columns for 'Permission Level' and 'Description'. The levels listed are Full Control, Design, Contribute, Read, Limited Access, and My Custom Permission Level. Each level has a corresponding description of its permissions.

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

## People

- WSS V3 introduces people to visual identities associated with current site

The screenshot shows the 'People and Groups: All People' page under 'People and Groups'. It lists four individuals: Litwareinc Administrator, Mike Fitzmaurice, System Account, and Ted Pattison. Each entry includes a checkbox, a picture, the name, job title, and department.

	Name	About Me	Job Title	Department
<input type="checkbox"/>	Litwareinc Administrator			
<input type="checkbox"/>	Mike Fitzmaurice		Senior Vice President	Research and Development
<input type="checkbox"/>	System Account			
<input type="checkbox"/>	Ted Pattison		Senior Developer	Strategic IT

## Groups

- WSS introduces groups to visualize aggregate groups from many places
  - WSS-created groups
  - Active Directory groups
  - Local groups
  - Groups from a custom role provider

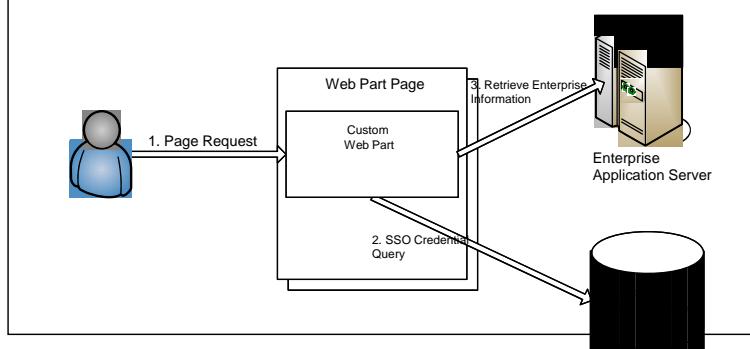
The screenshot shows the 'People and Groups: All Groups' page in SharePoint. The left navigation bar has 'Groups' selected, showing options like 'Team Site Members', 'Team Site Visitors', 'Team Site Owners', 'My Custom Group', and 'More...'. The main content area displays a table of groups with columns for 'GROUP', 'Edit', and 'About Me'. The groups listed are: LITWARESERVER1\my local windows group (description: Includes all users that need to access this site with a custom permission level), My Custom Group (description: This is the member group for the SharePoint site: Team Site: http://litwareserver1.), NT AUTHORITY\authenticated users (description: This is the owner group for the SharePoint site: Team Site: http://litwareserver1.), Team Site Members (description: This is the member group for the SharePoint site: Team Site: http://litwareserver1.), Team Site Owners (description: This is the owner group for the SharePoint site: Team Site: http://litwareserver1.), and Team Site Visitors (description: This is the visitor group for the SharePoint site: Team Site: http://litwareserver1.).

## Levels of rights assignment

- Access control to content configured at many levels
  - Site Collection
  - Site
  - List or Document Library
  - Item or Document
  - Child Site

## Single Sign-On

- Storage and Mapping of Credentials
  - Used for Accessing Enterprise Applications
  - Serves to reduces Number of Credential Prompts
  - Leveraged through custom Web Parts and BDC



## Summary

- User Authentication
  - Windows Authentication
  - Forms Authentication
- WSS Identities and Security Contexts
  - Application Pool Identity
  - The SHAREPOINT\System account
  - Escalation of Privilege
  - Delegating User Credentials
- The MOSS Single Sign-On Service (SSO)