

# **SharePoint 2010**

## Become an Expert

### FUNDAMENTALS

- 01 Introduction
- 03 Definitions History Expectations
- 07 Sites Lists
- 19 Pages, Web Parts, Themes
- 24 InfoPath Forms
- 29 Office Integration
- 33 Notifications and User Profiles
- 37 Security - Authorization
- 46 Site Administration Tools
- 52 Project Document Libraries and Sites
- 55 Advanced Information Management
- 67 Project Document Center Functionality
- 69 SharePoint Designer for Site Administrators

### ADMINISTRATION

- 77 Introduction to Administration
- 80 PowerShell
- 83 Site Collections as an Administrator
- 88 Web Application Management
- 93 IIS and SQL Server
- 99 User Profile Service Application
- 105 Search Service Application
- 110 Backup and Restore
- 116 Site Definitions and Features

### DEVELOPMENT

- 121 Development Overview
- 127 SharePoint API
- 137 Elevation and Impersonation
- 142 Client APIs
- 144 Simple Application Visual Studio 2010
- 151 Workflow Activity

Microsoft SharePoint 2010

## **Introduction**

### **Course Overview**

- **Introduction**
- **Target Audiences**
- **Course Structure**
- **Virtual Image Demo**

### Target Audiences

#### **Content Manager / Site Administrator**

SharePoint core features for collaboration, document management, and form management.

#### **System Administrator / Architect**

Planning, setting up, and managing server farm, web applications, databases, search, user profiles, and backup.

#### **Developer / Architect**

Using Visual Studio 2010 to customize SharePoint efficiently.

### Course Structure

- **Slides and printable workbook** – for you to take notes
- **Demos** – to watch concepts from slides in action
- **Labs** – to practice what you saw in demos
- **Free style exercises** – to solidify what you learned in labs
- **Other resources** – links to important SharePoint sites and a link to download the virtual image on which you can perform the labs

Microsoft SharePoint 2010

## **Definitions, History, Expectations**

### **What is SharePoint?**

**History**

**Difference in editions**

**Marketing definitions**

**Success with SharePoint**

### History 1995- 2010

**1995-2000** – During the internet boom many start ups created web-based portals, content management, and collaboration systems.

**2000** – Microsoft releases two products

- » SharePoint Portal Server 2001 for document management and enterprise search (Exchange backend)
- » SharePoint Team Services for collaboration (SQL backend)

**2003** – Microsoft releases version 2.0 products

- » Windows SharePoint Services (SQL and .NET)
- » Office SharePoint Portal Server 2003 (built on top of WSS)

**2006** – SharePoint 2007 (3.0) fixes many shortcomings of the 2003 product, introduces new features such as Business Data Catalog and InfoPath Form Services

**2010** – SharePoint 2010 (4.0). Improvements on technologies found in 2007 are welcome, as are some of the new features. Even for basic collaboration this is a superior product.

### Editions

**SharePoint Foundation 2010** – platform product that provides collaboration and document management for smaller teams.

- » This is similar to WSS in past releases

**SharePoint Server 2010** – With the foundation at its core, this version provides more functionality relating to the six terms surrounding the word SharePoint in the previous graphic.

- » This is similar to MOSS in the past. Just don't call them MOSS and WSS anymore. Call them SharePoint.

**SharePoint 2010 for Internet Sites** – A different licensing scheme used for sites where your target users are external and the sites will be web facing.

**SharePoint 2010 for Search** – Provides search functionality but not the other aspects of the product. Includes FAST search.

\* More SKUs exist, but these mentioned vary by more than just licensing terms.

### The SharePoint Wheel



### Defining the Wheel

**Sites** – a place to store data, do collaboration, and use the product to whatever extent you can.

**Communities** – Using SharePoint to create web based environments where people can interact. This relates to both internal and external use.

**Content** – Whether web pages or documents, SharePoint provides a mechanism to store, manage, handle versions, and attach metadata to them.

**Search** – SharePoint provides a robust search engine for finding your pages, documents, and other list items.

**Insights** – Making informed business decisions and collaborative effectiveness while enabling IT efficiency.

**Composites** – No-code collaborative solutions that allow businesses to meld control with flexibility.

### Where is the Insights presentation?

- The words defined are **broad-scoped marketing terms** hinting at possible use cases of the product. They bear only superficial resemblance to the nuts and bolts of the product whose understanding is of paramount importance when building solutions for your businesses.
- You could garner “Insights” from a mere list in certain cases, and in others gain them from a complicated web part page merging information from numerous external data sources.
- **There is no cure-all Insights module** for SharePoint. The same applies to other terms.

### SharePoint Success

- You don't have to (and likely shouldn't) implement all of SharePoint's capabilities for the product to be of great value to your organization. **Many successful SharePoint deployments are using less than half of its functionality.**
- **No feature will meet 100% of a user's demands.** It is important to understand SharePoint's limitations before committing to build a project around that feature. Generally, if a feature only meets 80% of the requirements it's too expensive to build the other 20%. You are better served finding an off the shelf work around or dropping that requirement.



Microsoft SharePoint 2010

## **Sites Lists**

### **Functional Overview**

**Site Collection**

**Site**

**List**

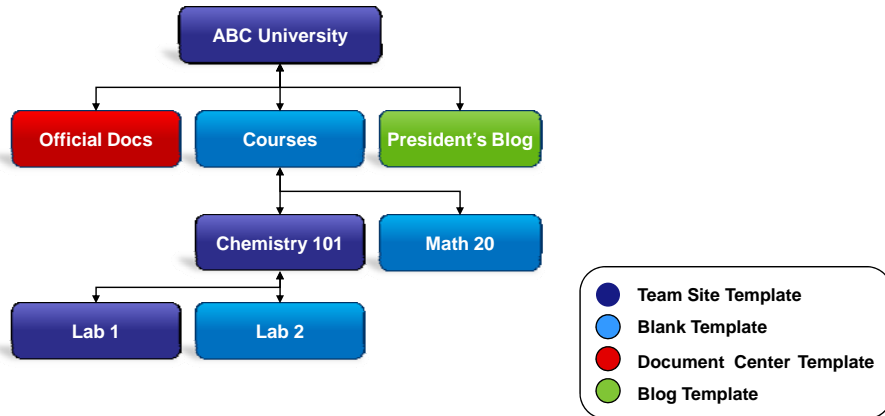
**Versioning**

**Draft Publishing Model**

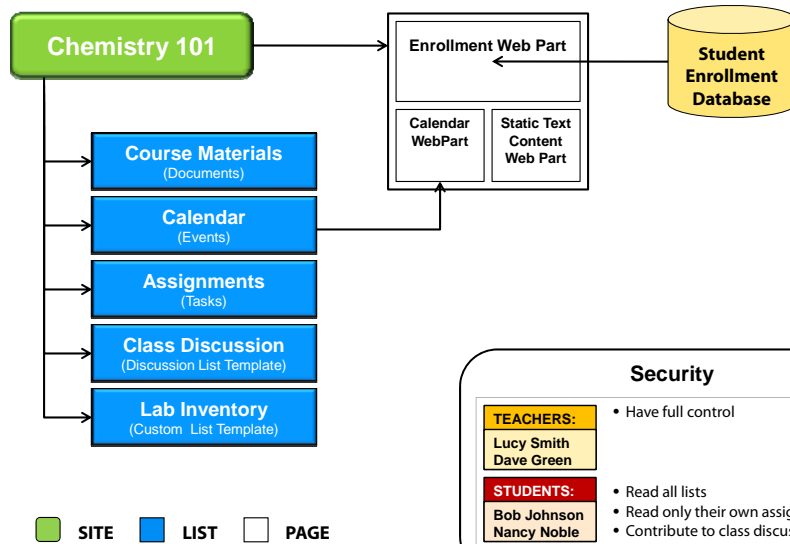
## Site Collection

**Site Collection** is a set of hierarchically organized sites with one top or root site and many sub sites on different levels.

**Site template** is an initial configuration of lists, pages, and features.



## Site



### Navigation in WSS

**Top Link Bar** – horizontal collection of links controlled by site content managers. Limited, no drop downs.

**Quick Launch Bar** – menu on the left controlled by site content managers

**Tree View** – folder view of SharePoint sites and lists. Shows all content to which the the user has access. No easy way to exclude content items except for permissions.

**Current Location** – provides the location and easy navigation in the site collection

**Breadcrumb** – useful for navigating within the site

### Navigation Customization

#### With SharePoint Designer, content managers can

- » Add **more levels** to Top Link bar and Quick Launch Bar
- » Set the default **number of levels to expand**
- » Make the **tree view frame wider** which is important when there are long list or folder names

#### When Publishing Feature is enabled

- » Top Link Bar becomes **Global Navigation** with one level of drop downs and easy organization
- » Quick Launch Bar becomes **Current Navigation**

### Demo Overview

#### **Demo:** Creating and Navigating Sites and Lists

1. Create several sites and explore site navigation
  - Courses - "Blank Site" template
  - Chemistry 101 - "Blank Site" template
  - Information Software Design - "Team Site" template
  - President's Blog - "Blog" Template
2. Create several lists and explore list navigation
  - Course Materials - "Document Library" template
  - Assignments - "Tasks" template
  - Course Directory - "Custom List" template

### Lab: Exercise 1

#### **Perform Lab:**

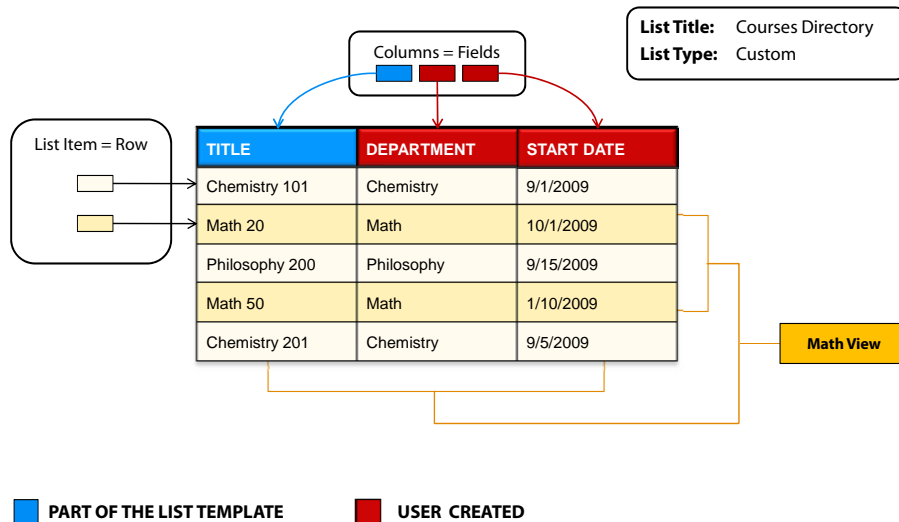
- **Exercise 1** from **Sites and Lists Lab**

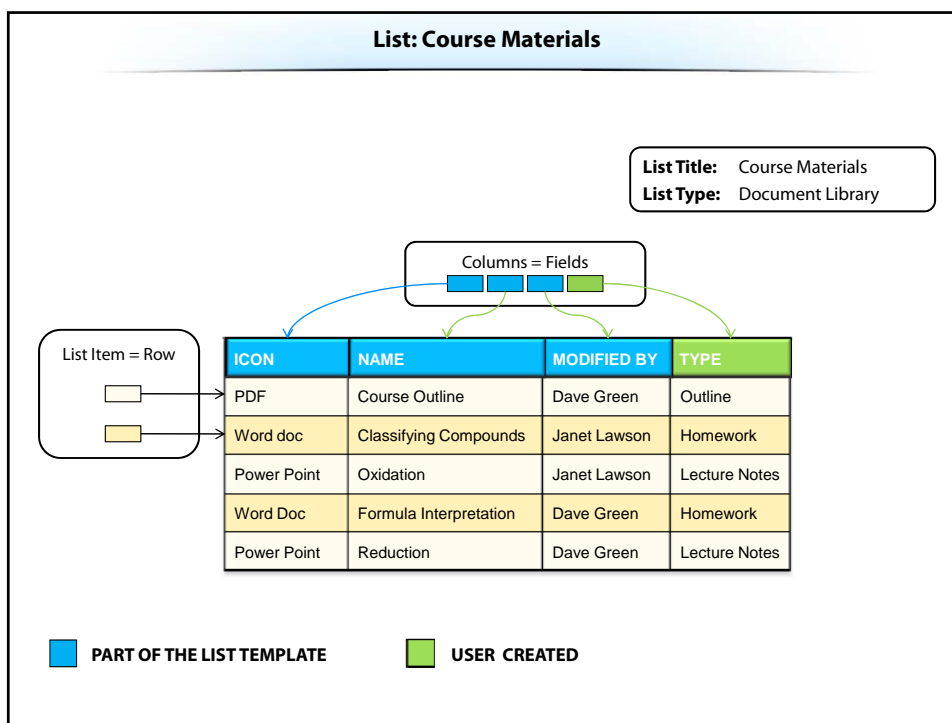
## List

### List = conceptually a table:

- » stores a collection of lists items (rows)
- » specifies columns (fields) that each list item will have. Initial set of columns determined by the list template. User can add more columns.
- » has configurable and customizable forms for data entry
- » has views that display subsets of list items using sorting, filtering and grouping
- » controls access based on user's permission levels

## List: Course Directory





**Demo Overview**

**Demo: Adding Columns and Items to Lists**

1. In **Course Directory** list:
  - Add Columns
    - Course Description”(Text)
    - Site URL (URL)
    - Department (Choice)
    - Start Date (Date) and End Date (Date)
    - Professor (Person)
    - Length (Calculated)
  - Create list items
  - Setup validation and navigation options
2. In **Course Materials** document library:
  - Add Materials Type (Choice) column
  - Create documents (list items)

### Demo Overview

#### Demo: Adding Views to Lists

1. In **Course Directory** list:

- Add Views
  - Chemistry Courses
  - Calendar
  - Gantt

2. In **Course Materials** document library:

- Add Homework view
- Setup per location views

### Lab: Exercises 2-3

#### Perform Lab:

- **Exercise 2 and 3** from **Sites and Lists Lab**

### Versions

- **Every list type supports versioning** with any change to the list item resulting in a new version
- **Versioning options are set per list**
- **Library lists** (documents, forms, etc) also provide
  - » **Check out** (lock) / **check in** (unlock) mechanism with optional required check out
  - » Draft/Publishing model using major and minor versions

### How New Versions Are Created

- **In check out/check in process**, one new version results regardless of how many times the user edits and saves the document.
  - » Current version: **3.0**
    - Check out, edit, and save
    - Edit and save
    - Edit, save, and check in
  - » Current version: **4.0**
- **Without check out**, a new version results every time user saves and edits the document
  - » Current version: **3.0**
    - Edit and save
    - Edit and save
    - Edit and save
  - » Current version: **6.0**

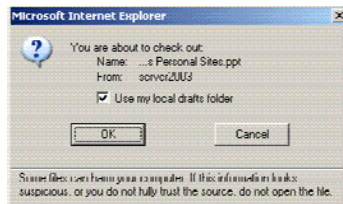


### Editing Conflicts and Required Check out

- **When a document is checked out**, one user controls that document.
- **Required check out** forces the user to check out a document before editing. One side effect is when a document uploaded to a document library through a windows explorer view, it still have to be checked in.
- **In non library lists**, explicit check out/check in is not available. If there are two users trying to edit the same item, the one of them will get editing conflict error.

### Library Versions: Local vs. Server

- During check out, a user can place a document on the local computer using local drafts check box. Otherwise a document will stay on the server.



- Ability to edit a document **offline** and **fast local saves** instead of saves to the server (that can take a long time) are the advantages of using local drafts.

### Demo: Versioning

1. Examine versioning options in a list and a document library
2. Edit an a list item
3. Check out, edit, and check in a document

### Versions: Publish and Draft

- **Major** (publish) and **Minor** (draft) version model closely matches the edit process in real world, where a person works on the document many times making a series of changes (draft versions that only editors can access) and finally publishes a major version (public version that everyone can see).
- In file system, draft/publish model requires two separate folders with different permissions. It is difficult for people to switch from this type of thinking. However, there are many advantages to keeping documents in one place.

### Versions: Content Approval and Draft Item Security

- **Content Approval specifies** whether changes to items should remain in a draft state until they have been approved.
- Approval takes place after a user publishes a major version of the document which remains unpublished until it's approved.

### Demo Overview

#### **Demo:** Versioning – Publish, Draft, and Content Approval

1. Enable major and minor versions in a document library
2. Go the process of editing a document
3. Turn content approval on and go through the approval process in a document library

**Lab: Exercise 4**

**Perform Lab:**

- **Exercise 4** from **Sites and Lists Lab**

Microsoft SharePoint 2010

## **Pages, Web Parts, Themes**

### **Pages, Web Parts, Themes**

#### **Types of Pages**

#### **Web Parts**

- » Properties
- » Connections
- » Web Part Gallery

#### **Site Customization**

- » Logos
- » Themes

### Page Types

**Web Part Page** – add or remove web parts in a predefined layout of web part zones.

**Wiki Page** – easily change layout and appearance, add or remove web parts, go through a publishing process controlled by **Site Pages** versioning settings.

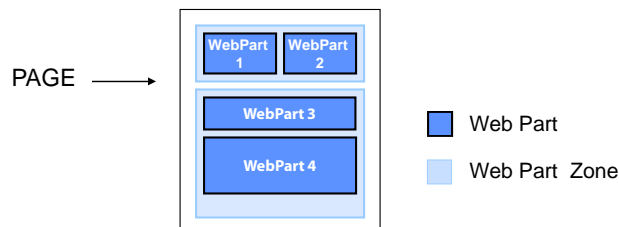
**Publishing Page** – for advanced content management needs. Designers define layouts shared by many publishing pages. Page can go through a publishing process controlled by **Pages** versioning settings.

**Application Page** – system page editable only by developers.

### Web Part Page and Web Parts

**Web Part Page** consist of web part zones arranged in a specific layout. There are a number web part layout templates available.

**Web Part** is a window/chrome that provides access to specific functionality. Web parts reside within web part zones.



### Web Part Details

- **Web parts have two types of properties:**
  - » **Base** - control appearance and behavior (title, height, width)
  - » **Custom** - specific to the functionality (zip code, view name, mailbox)
- **Properties can be:**
  - » **Shared** - changes affect all users
  - » **Personal** - changes only affect logged in user
- Web parts can connect and pass data or properties to other web parts

### Web Part Gallery

- **Web Part Gallery** is a document library that stores available web part definitions for the site collection, located at the root site.
- Users can add and modify web part definitions in order to control what appears on Add Web Part Page.

### Demo Overview

#### Demo: Web Parts

1. Examine different types of pages
2. Add web parts
3. Examine web part properties and connections
4. Examine Web Part Gallery list for the site collection

### Demo Overview

#### Demo: Site Image and Theme

1. Change the site image
2. Change the site theme



**Lab: Exercises 1- 4**

**Perform Lab:**

- **Exercises 1- 4** from **Pages, Web Parts, Themes Lab**

Microsoft SharePoint 2010

## **InfoPath Forms**

### **InfoPath Forms Overview**

#### **Form Design**

- » Fields / Columns
- » Visual Layout
- » Logic and Rules
- » Validation
- » External Data

#### **InfoPath Form Web Part**

#### **List forms vs. form library**

### Data Structure and Design

**Tip:** When building a new form, start with defining fields in the form or columns in the list.

**Tip:** Try to use specific controls for each field instead of generic text fields.

**Data design early is important** because in many cases once users fill out the form, it is difficult to get them to come back.

### Demo Overview

#### **Demo:** Lab Accident Form Data Design

1. Create a form for a list
2. Add following fields to the form
  1. Accident type and description
  2. Start time, location, lab manager
  3. Root cause and corrective actions
  4. Amount and type of chemicals spilled in Chemical Spill Option section
  5. How long did it take to put out fire in Fire Section
  6. Injury information

### Visual Layout

- **There are a number of layouts available**
  - » Two column layout is standard to SharePoint forms
  - » Four column layout can make forms shorter and more compact improving usability
- **SharePoint themes** enable designers to quickly select professional themes that work well with SharePoint

**Tip:** Group similar fields close together

### Logic

**Rules engine** comes with a number of pre built rules. Designers can create their own rules of different types:

- » **Data validation** – validate user input
- » **Formatting** – format inputs or layouts based on conditions. For example hide or show controls based on a value of a field.
- » **Actions** – set values of fields which can trigger formatting elsewhere

### Demo Overview

#### Demo: Lab Accident Form Layout Logic

1. Use Four Column layout
2. Show the following conditionally
  1. Chemical Spill Options - amount and type of chemicals spilled if accident type is chemical spill
  2. Fire Options - how long did it take to put out fire if accident type is fire
  3. Injury Options if description contain injury
3. Add validation logic to accident date and amount of chemical spilled

### External Data

#### SharePoint lists

- » On the current site or other sites

#### SQL Database

#### Web Services

- » SOAP web services such as SharePoint
- » RESP web services such as Bing Maps or Twitter

### Demo Overview

#### Demo: Lab Accident Form External Data

1. Make location field retrieve choices from a SharePoint list

### List vs. Library Forms

- Besides list columns, libraries also store form data as a file.
- With a large **complicated schema** (many columns), the designer can choose just a few of them to be list columns and the rest of the data will be in the file.
- **Digital Signatures** only work with files and not list items
- Sandboxed code

Microsoft SharePoint 2010  
**Office Integration**

**Office Integration**

**Outlook**

- » Calendars, Contacts, and Tasks

**Access**

- » Web Databases (SharePoint sites)

**SharePoint Workspace**

- » Offline Content

**Excel Services**

- » Business Intelligence Dashboards Using Web Parts and Filters

### Demo Overview

#### Demo: Outlook

1. Use Connect to Outlook functionality for Calendar, Contacts, and Tasks lists.

### Access Services

**Access Services** allows database designers to publish and synchronize databases with SharePoint sites. Tables become Lists and Forms become web based.

- Sites Administrators can create sites that use Access Database templates such as Assets or Contributions
- This functionality competes with using a combination of SharePoint lists and forms customized using InfoPath. We recommend using SharePoint + InfoPath unless you already have an investment in Access.



#### Demo Overview

#### **Demo:** Access Web Databases

1. Create a site using one of the access templates and examine how access tables and forms are rendered

#### Demo Overview

#### **Demo:** SharePoint Workspace

1. Examine ability to take content offline using SharePoint Workspace

## Excel Services and Dashboards

**Excel Services** allows authors to share workbooks through the browser and to control what part of the workbook is visible to the users and as well as what parameters users can modify.

**Business Intelligence Dashboard** is a web part page that typically contains excel services and filter web parts and allows users to see data in different ways.

**Data Connection Library** is a document library template that stores and manages Office Data Connection files to external data sources if Excel workbook needs them.

## Demo Overview

### Demo: Introduction to Excel Services

1. Examine Business Intelligence features
2. Create an excel pivot table and chart that uses external data and publish to SharePoint using Excel Services
3. Use Web Access web part to display the published excel file
4. Configure Filter web parts to change what's displayed

Microsoft SharePoint 2010

## **Notifications and User Profiles**

### **Staying up to Date and User Profiles**

**List Alerts and RSS Feeds**

**User Profiles**

**Personal Site / Content**

**Search**

### Notifications

- **Alerts** give users an ability to be notified of list item or list events either when they occur or at a later time.
  - » Users with proper permissions can create alerts for other users.
- Each list has an **RSS feed**. User can subscribe to RSS feeds and track lists and list items on different sites using an RSS reader.

### Demo Overview

#### Demo: Alerts and RSS Feeds

1. Create an alert for a list item and a list
2. Subscribe to RSS feeds for different lists

### Personal Site/User Profile Purpose

#### User Profile Stores

- » Person's Expertise
- » Contact, Organizational, and Social Networking Information

**Personal site** – a site collection dedicated for lists and documents of a particular user. This has always been a confusing point for most organizations. We recommend not deploying personal sites if there is not a clear need.

### Demo Overview

#### Demo: User Profile and Personal Content

1. Editing user profile information
2. Managing personal content

## Demo Overview

### Demo: Search

1. Use search to find documents
2. Use search to find people

Microsoft SharePoint 2010

## **Security - Authorization**

### **Security Overview**

- Permissions and Permission Levels
- Site Collection Administrator
- Authentication vs. Authorization
- What provides authentication
- Who can assign permissions
- SharePoint groups

### Permissions and Permission Levels

**Permission** gives user a right to perform an action on a resource. User interface shows only permitted actions.

example:

List permissions: Add Items, Delete Items, ...

Site permissions: View Usage Data, Create Subsites, ..

**Permission Level** is a collection of permissions

Default Permission Levels:

Read

Contribute

Design

Full Control

### Site Collection Administrator

**Site Collection Administrator** is a special user that has Full Control permission level to every site in the site collection plus more.

- » Site collection administrator can make any other user into site collection administrator
- » System administrators can designate site collection administrators



### Demo Overview

#### Demo: Permission Levels

1. Examine default permission levels and permissions that make them up at the root site
2. Examine Site Collection Administrator settings

### Authentication vs. Authorization

**Authentication** – the process of verifying user identity.

example:

Entering credentials during a log-on process into a computer or a website.

**Authorization** – determining the permissions level that an authenticated user has to a SharePoint resource such as site, list, folder, and list items.

example:

Can Dave Green create lists in Chemistry 101 site? Does Lucy Smith have “professor assistant” permission level in Humanities 156?

### Who provides authentication

#### **Intranet: Active Directory (AD)**

- » Stores users and groups
- » Authenticates them via Windows Authentication.
- » Most SharePoint site administrators do have permissions to create users in AD, they can only use existing ones.

#### **Public facing site or extranet: LDAP or SQL Database**

- » Stores users and groups
- » Authenticates them via Forms Authentication.
- » Typically SharePoint administrators can create users via SharePoint user interface.

### Who can grant permissions

Users that have "**Manage Permissions**" permission (part of Full Control permission level by default) can grant a permission level to any user or group from a user store (AD, SQL, LDAP)

### Permissions Inheritance

- SharePoint resource can inherit permissions from the parent resource or have unique permissions.
- **Parent Site > Site > List > Folder > List Item**
- Site can not inherit permissions from a site in a different site collection.

### Demo Overview

#### **Demo:** Granting Permissions to AD Users

1. Examine Active Directory users
2. Grant a permission level to a user from AD at the root site, sub site, list, folder, and item level.
3. Grant a permission level to a group from AD
4. Check user's permissions

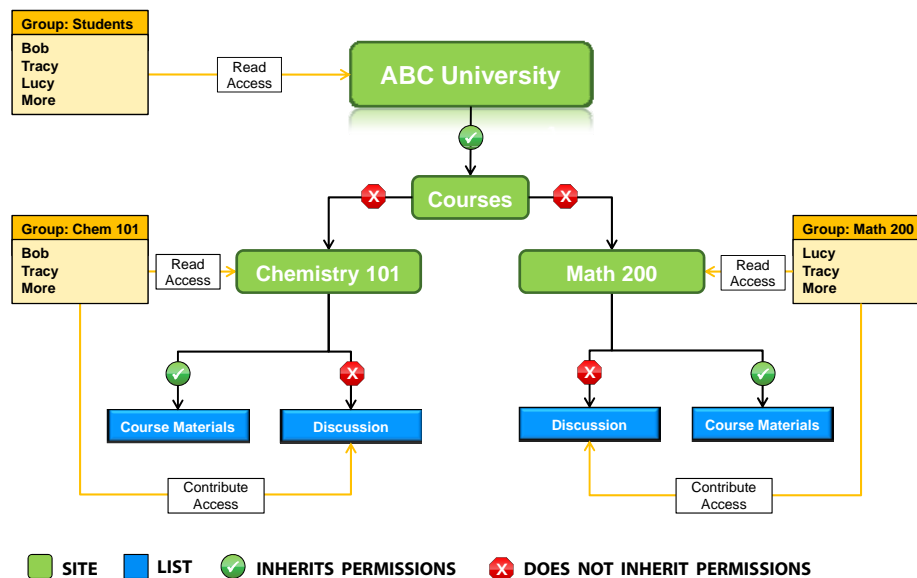
### Security Definition: SharePoint Group

**SharePoint Group** is collection of users and/or groups

- » Scope is one site collection
- » Users or groups come from a user store (AD, SQL, LDAP).
- » Cannot contain other SharePoint groups

Using SharePoint groups simplifies permission management as instead of granting a permission level to each user separately, one can add those users to a group, and then grant a permission level to that group.

### Security: Courses Directory



### Demo Overview

#### **Demo:** SharePoint Group Management

1. Create a new SharePoint group and add users to it
2. Assign permission level to the group at a site, list, folder, and item level.
3. Check group's permissions
4. Examine the setup of Visitors, Members, and Owners groups for a new site that does not inherit permissions

### Lab: Exercises 1-4

#### **Perform Lab:**

- **Exercises 1 – 4 from Security Lab**

### Security Notes

- Users with permissions to access specific lists and list items only will have “**Limited Access**” permission level on the site.
- Non document library lists have a setting to allow users to read and edit only their own items.
- Assigning a permission level to all authenticated users group (**NT Authority\Authenticated Users**) gives everyone in Active Directory access to a SharePoint resource

### User Information List

**User information list** contains all the users and groups in the site collection

- » Typically, other lists lookup values from user information list.

**example:**

“Assigned To” column in “Tasks” list or “Modified By” in “Document Library”.  
Changing display names of users in the user information list will change them for everything in the site collection.

### Demo Overview

#### Demo: Security Misc

1. Examine User Information List
2. Examine List specific permissions
3. Limited Access

Microsoft SharePoint 2010  
**Site Administration Tools**

**Site Collection Administration**

- Features
- Site Content and Structure Tool in Publishing Feature
- List and Site Templates
- Usage Reporting



## Features

**Feature** specifies some functionality

example of features:

- Navigation
- Document Library list
- Wiki Page

**Features have different scopes:**

- » Document library has a **Site** scope
- » List Template Gallery has a **Site Collection** scope

**Site Templates** activate certain features upon creation

## Features – Site Collection Administrator Role

**Developers Create Features**

**System Administrators**

- » Install and uninstall features on the server
- » Activate and deactivate features at the farm, web application, site collection, or site scope

**Site Collection Administrators**

- » Activate and deactivate features at site collection or site scope
- » Install features in solution sandbox

#### **Demo: Features**

##### **Demo: Features**

1. Activate and deactivate one of the site features
2. Activate and deactivate one of the site collection features
3. Examine what features are activated by different site templates

#### **Demo: Manage Content and Structure Tool**

##### **Demo: Manage Content and Structure Tool**

1. Move sites within site collection
2. Observe dependence on publishing infrastructure

### Site and List Templates

- Users save lists and sites as templates by using “**save as template**” action. **Security settings are not saved.**
- When saved, SharePoint stores the templates as files in **Solution** and **List Template** galleries of the root site. There is a size limited that system administrators can change.
- Templates are available through the site collection. Users can move them to other site collections by uploading to target galleries.

### Demo: List Template

#### Demo: List Template

1. Save “Assignments” list as a template
2. Create another list based on “Assignments” template within the same site collection

### Demo: Site Template

#### Demo: Site Template

1. Create a site using "Blank" template. Make changes to the site, and save the site as template including content
2. Create a site based on template within the same site collection

### Demo: Usage Reporting

#### Demo: Usage Reporting

1. Examine the usage reporting feature

**Lab: Exercises 1-3**

**Perform Lab:**

- **Exercises 1 – 3 from Administrator Tools Lab**

Microsoft SharePoint 2010

## **Project Document Libraries and Sites**

### **Project : Libraries and Lists**

#### **Research Documents**

- » Metadata creation
- » Metadata navigation
- » Permissions by folder
- » Views without folders

#### **Project Site Management**

- » Project directory
- » Project site template

#### **Media – Assets Library**

- » Images
- » videos

#### Demo Overview

##### **Demo:** Research Documents

1. Build a document library with columns that will classify research and provide navigation metadata
2. Setup permissions using groups by folder
3. Setup views that display information without folders

#### Demo Overview

##### **Demo:** Projects Area

1. Build a project site
2. Use the site a project template
3. Setup project directory

### Demo Overview

#### Demo: Assets Library

1. Examine Assets Library functionality for storing images and videos in SharePoint

### Free Style Lab Exercise

#### Free Style Lab Exercise

1. Perform all exercises from **Project: Libraries and Lists**



Microsoft SharePoint 2010

## **Advanced Information Management**

### **Advanced Information Management**

**Site Columns**

**Content Types**

**Information Management Policies**

**Workflows**

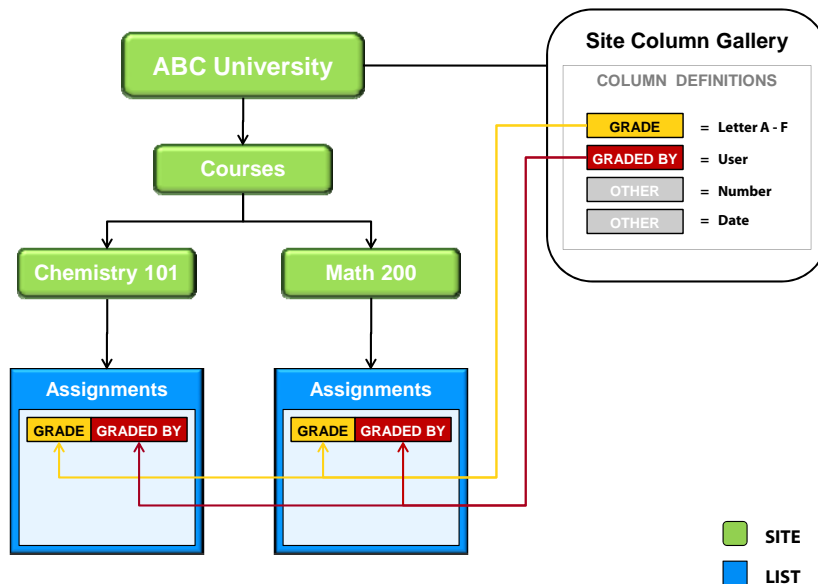
### Site Columns: Overview

**Site Columns** provide a way to reuse a column definition on one site or many sites.

**Site Columns Gallery** stores site columns (outside of any lists).

To make a list column, SharePoint makes a copy of the site column, stores it in the list, and retains a pointer to the original.

### Site Columns: Illustration



### Site Column Scopes and Updates

- Site columns are available at the site where they are created and sites below within the same site collection.
- **Updating a site column**, also updates all the list columns linked to that site column.
- **User can change list columns** without affecting the site column. **This is not recommended** as site column updates are not granular; they will override list columns.

### Demo Overview

#### Demo: Site Columns

1. Create a site column "Assignment Type" (Choice) at the root level site "Site columns" gallery.
2. Add "Assignment Type" site column to two "Assignments" lists (create another list if necessary)
3. Demonstrate updates and conflicts by modifying list column definitions and site collection column definition
4. Create a lookup site column and use it in several locations

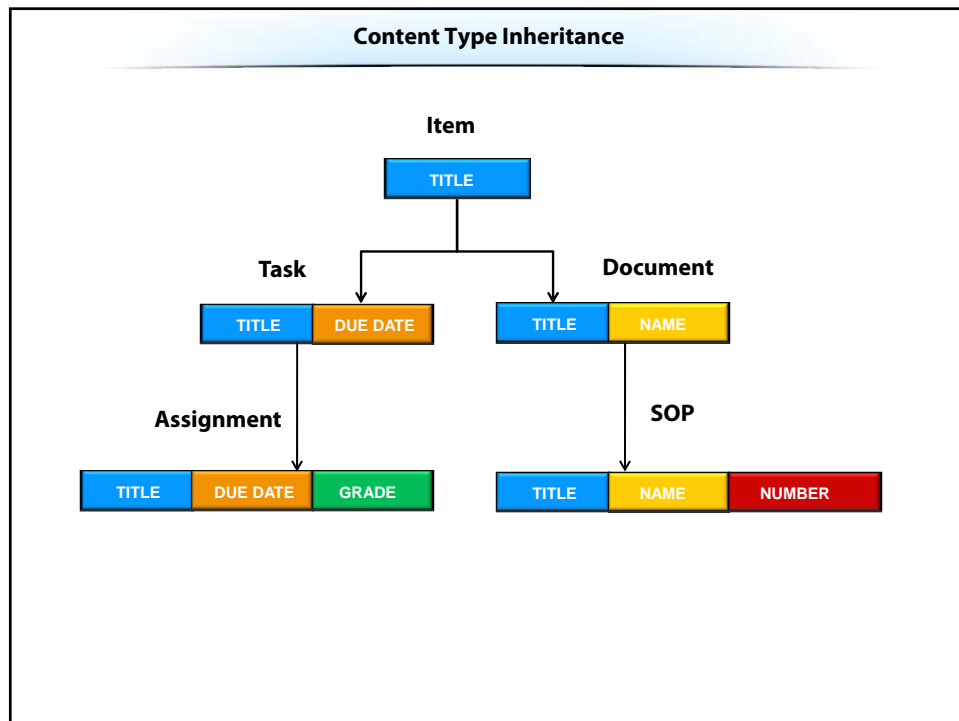
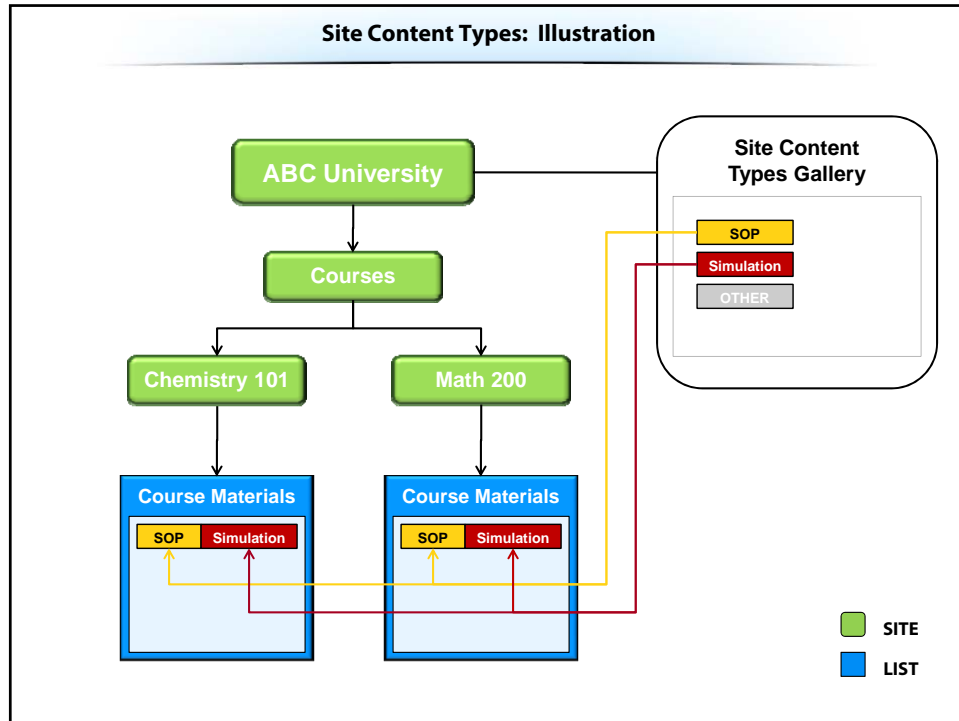
### Lab: Exercise 1

#### Perform Lab:

- **Exercise 1** from **Advanced Information Management Lab**

### Content Types

- **Content Type** is a collection of settings applied to content
  - » Set of Columns
  - » Forms
  - » Workflows
  - » Information Management Policies
- Site Content Types behave just like site columns except that they are stored in **Site Content Types Gallery**.
- Each list usually starts with one content type. Users can add more.



#### **Content Types Global Changes Use Case**

- Content types are useful if you are making changes through the entire site collection.
- Adding a site column called NDA Required to the Document content type would propagate to every single document library that uses it.

#### **Content Types One List Use Case**

- If you are planning to store two types of documents in a document library: Standard Operating Procedure and Equipment Manual
- Use content types if each document type requires different forms, workflows, or information management policies.
- Otherwise, use a column called "Materials Type" with two choices and tag each document. In many cases this is the simpler option.

#### Demo Overview

##### **Demo:** Assignment Content Type

1. Create a new content type "Assignment" that inherits from a "Task" content type.
2. Add "Grade" and "Graded By" columns.
3. Enable the "Assignment" content type in several lists.
4. Make changes to the content type

#### Demo Overview

##### **Demo:** SOP and Equipment Manual Content Type

1. Create a content type "SOP" based on "Document" with SOP Number and SOP Purpose columns
2. Create a content type "Equipment Material" based on "Document" with Manufacturer column
3. Test content types within a document library

### Information Management Policy

- SharePoint supports the following **information policy management tools**:
  - » Document Retention and Expiration Schedules
  - » Auditing of Document Events
  - » Labeling and Barcodes
- **Administrators can create a policy for**:
  - » Entire Site Collection
  - » Content Type
  - » List

### Demo Overview

#### **Demo:** Information Policy Management

1. Create a site collection information management policy
2. Attach the policy to a specific content type and then create content type specific information management policy.
3. Attach a site collection policy to a specific document library
4. Examine how to run audit reports



### Lab: Exercises 2-3

#### Perform Lab:

- **Exercises 2 - 3** from **Advanced Information Management Lab**

### Workflow

**Workflow** is a process around content - documents, lists items, or sites

example:

- Course syllabus must be reviewed and approved by chemistry department head
- Simulation needs approval of all chemistry professors

**Out of box**, a number of configurable workflow templates such as "Approval" and "Three State", provide core workflow functionality.

### Association

**During association**, user connects a workflow template to a list, site, or content type, and configures the following properties:

- » Association name
- » Tasks and history lists location
- » Manual or automatic start
- » Number of stages and type of execution at each stage: parallel (tasks assigned to all users at once) or serial (one user gets a task at a time)
- » Default values for users that will be involved
- » Completion conditions

**After association**, users can initiate (start) workflows on list items or sites.

### Demo Overview

#### **Demo:** Workflow for a List

1. Add a workflow called "Course Materials Approval" to "Course Materials" list with the following settings
  - a. User starts manually
  - b. Serial order with specific approvers
  - c. Tasks are due 2 days after assignment
2. Initiate "Course Materials Approval" workflow on a specific document within a document library
3. Complete workflow tasks
4. Office Integration

### Demo Overview

#### Demo: Workflow for SOP Content Type

1. Add a workflow called "SOP Approval" to SOP content type with the following settings
  - a. User starts manually
  - b. Parallel order with specific approvers from each department
2. Create a multi-stage workflow for SOP content type

### Other Workflow Options

- Using **SharePoint Designer**, user can quickly create workflows complete with different types of logic and actions.
- Using **Visual Studio**, developers can create new workflow templates (**very difficult and time consuming**) or activities for SharePoint Designer (**not as difficult as the entire workflow template**)
- **Third party developers** such as Nintex provide an alternative SharePoint Designer or Visual Studio development.

**Lab: Exercises 4-5**

**Perform Lab:**

- **Exercises 4 - 5 from Advanced Information Management Lab**

Microsoft SharePoint 2010

## **Project Document Center Functionality**

### **Project Document Center**

#### **Examine document center default functionality**

- » Link to Document content type
- » Document Sets
  - Versioning
  - Workflows
- » Document IDs
- » Content Navigation
- » Site Collection Features

### Demo Overview

#### Demo: Document Center

1. Examine Document Center Functionality

### Free Style Exercise

#### Free Style Exercise

1. Examine Document Center Functionality

Microsoft SharePoint 2010  
**SharePoint Designer**

**SharePoint Designer 2010**

**Editing Look and Feel**

- » Master and content pages
- » List forms

**Data Sources and Views**

**Business Data Services**

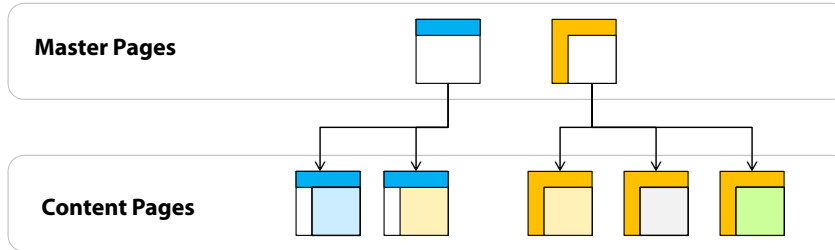
- » External lists and content types
- » Editing forms with InfoPath

**Workflows**

- » Terminology and types
- » Editing forms with InfoPath

### Master and Content Pages

Designers can factor common parts of **content pages** (.aspx files) into a **master page** (.master file located in master page gallery)



### Customized vs. Not Customized

- SharePoint retrieves **Un-customized pages** from site definitions that are stored on the web server file system.
- Once **customized**, SharePoint saves and retrieves the pages from the content database instead of the site definition. Thus, subsequent changes made to the side definition do not affect customized pages.
- **Revert to site definition** – means revert to un-customized page



### List Forms

- SharePoint designer **Custom List Form** provides an **alternative** to InfoPath.
- Modifying forms in Designer requires good knowledge of HTML and CSS. Therefore **we recommend using InfoPath**.

### Demo Overview

#### **Demo:** Editing Look and Feel

1. Navigating through a site using SharePoint Designer
2. Edit content pages including list forms
3. Edit master page in the master page gallery

## Data Sources and Views

**Data Sources** organizes data that can be used in the site

- » SQL databases
- » Business Data Services External Lists
- » SOAP or REST Web services
- » SharePoint lists on the current site and other sites
- » XML files

**Data View** web part shows sorted, filtered, conditionally formatted data from a data source. Data views can communicate through web part connections.

## Demo Overview

### Demo: Data Views and Data Sources

1. Adding Data Sources
  - » List on a current site and on a different site (connect to another data source library)
  - » SQL database
2. Creating Data Views from data sources
  - » Properties
  - » Conditional formatting
  - » Web part connections

### Data Source Catalog

- **Business Data Services** provides access to information that resides outside of SharePoint.
- With Designer, users can create **external content types and lists** and customize add/edit forms using InfoPath.
- External lists do not support certain features such as alerts or workflows

### Demo Overview

#### **Demo:** External Content Types and Lists

1. Create an external content type
2. Create an external list based on external content type
3. Use InfoPath to edit list forms

### Workflows

- For business users a workflow defines business **process around content**
- For developers a workflow defines **application logic around data**
- SharePoint Designer makes  
**business users  $\approx$  developers**

### Workflow Terminology

**Event** is what starts a workflow.

- » User creates or edits an item
- » User explicitly starts a workflow

**Action** is a something that workflow can do.

- » Create, edit, delete list items in any list
- » Lookup information
- » Assign tasks
- » Do a calculation
- » Wait
- » Send email
- » And much more

### Workflow Terminology

**Condition** controls whether an action is performed.

- » If column value is equal to a predefined value, then update an item in another list

**Conditional Branch** controls the flow of actions.

- » If user provides a certain value then performs one set of action else performs a different set of actions

**Step** is a combination of conditions, branches, and actions. Workflow consists of steps.

**Variables** can store data between actions.

### Workflow Types

**List** workflow is associated with a list. Users start it at the item level

**Site** workflow is associated with a site, and can perform actions on any list. Users start it from site actions menu.

**Reusable** workflow is associated with a content type.

### Demo Overview

#### **Demo:** Designer Workflow

1. Create a workflow with multiple steps, many different actions, and conditional branches.

### Lab: Exercises 1-5

#### **Perform Lab:**

- **Exercises 1- 5** from **SharePoint Designer Lab**

Microsoft SharePoint 2010

## **Introduction to Administration**

### **Introduction to Administration**

- **What to expect**
- **What we will learn**
  - » PowerShell
  - » Site Collections and Web Applications
  - » IIS and SQL Server
  - » Service Applications
  - » Site Definitions and Features
  - » Changes from 2007
- **Where to go from here**

### Great Expectations

- **Our goal** is to learn how the various pieces of the farm interoperate, and what those pieces are
  - » You should end with a solid grasp of the architecture, and will understand enough to properly research the specifics for your environment
- This is not intended to be a reference manual
  - » Extensive documentation is available on TechNet

### Topics we will cover

- **PowerShell**
  - » STSADM has been deprecated
- **Site Collections and Web Applications**
  - » In the first part of the class we learned about sites and end user functionality
  - » Now we learn where and how the sites are stored
- **IIS and SQL Server**
  - » Though direct interaction with these components should be minimized, it is important to understand their role in SharePoint



### Topics we will cover – Part 2

- **Service Applications**
  - » The Shared Service Provider has been split into separate service applications for each service
- **Site Definitions and Features**
  - » Where they're stored, basic structure, new abilities
- **Changes from 2007**
  - » Throughout the course we will be pointing out things that are different in 2010

### What Next?

- With full understanding of this material, you should be ready to configure and administer SharePoint
- While every deployment has issues at some point, you should be prepared to research and repair those that you run into

Microsoft SharePoint 2010

## **PowerShell**

### **PowerShell**

**Stsadm?**

**PowerShell introduction**

**Basic Examples of using PowerShell**

### Goodbye stsadm

**Stsadm** was the way to administer SharePoint 2003 and 2007 from the command line

- » Problem
  - New process for every task affected scripting speed
  - Extensibility – adding features was difficult
- » Solution
  - Use PowerShell cmdlets

Support for stsadm will disappear eventually, so learn PowerShell

### Cmdlets

- **PowerShell** is based on the idea of cmdlets
  - » Pronounced “command-let”
  - » Conceptually: an application launched within the same PowerShell process
  - » You can launch other applications from within PowerShell (stsadm, for instance), but you lose the benefit of not needing to spawn new processes
- **Cmdlets** take the form of “verb-noun”
  - » For instance: get-help, new-spweb

### Demo Introduction

#### Demo: Launch and use PowerShell

1. Launch PowerShell for SharePoint Administration
2. Get-help
3. Make a new site using PowerShell
4. Get a list of all the site collections that exist in the farm

### Exercises

We will use **PowerShell** in other parts of the Administration course

Microsoft SharePoint 2010

## **Site Collections as an Administrator**

### **Site Collections as an Administrator**

- **What a Site Collection is**
  - » Functional description
- **Why to make multiple Site Collections**
  - » Separation of Data
  - » Backup Considerations
- **How to create Site Collections**
  - » Central Administration
  - » Self-Service
  - » Command Line

### Site Collection Description

- **Container**
  - » A site collection holds one top level site and whatever sub-sites are defined below it
  - » All data within the sites are guaranteed to be in the same content database as the site collection itself
- **Has Features**
  - » Features enabled at the site collection scope are available to all entities contained within
  - » While site features never depend on features activated at other sites (including parents), they may depend on site collection features

### Managed Paths

#### **Managed paths**

- » Only one site collection made within a web application (more on web applications later) can live at the root URL
- » Additional site collections within the same web app must be named and appear under a managed path such as **sites**
- » A managed path can be thought of as a placeholder (so that no sub-site under the root is created there) under which you place additional site collections

### When to make a new Site Collection

- **Data segregation**
  - » Permissions never inherit across site collections, so accidental access granting is less likely
  - » This typically makes sense for different departments in an organization
- **More granular backup and restore – requires distinct content database**
  - » Restores are faster when the data set to be restored is smaller
  - » If a content database gets corrupted only part of the deployment is lost and in need of restoration
  - » If different sites require different backup schedules, they should to be in separate content databases

### Creation Options

	Pros	Cons
<b>Central Administration</b>	<ul style="list-style-type: none"><li>• Simple interface</li></ul>	<ul style="list-style-type: none"><li>• Content database assigned automatically</li><li>• Must choose a site template</li></ul>
<b>Self Service</b>	<ul style="list-style-type: none"><li>• Less work for system administrators</li></ul>	<ul style="list-style-type: none"><li>• Governance</li></ul>
<b>Power Shell</b>	<ul style="list-style-type: none"><li>• Choose a specific content database</li><li>• No need to specify a site template</li><li>• Scriptable</li></ul>	<ul style="list-style-type: none"><li>• Direct server access</li><li>• Minor learning curve</li></ul>

#### Demo Introduction

##### Demo: Create Site Collection

1. Make a new managed path for department site collections
2. Make a Music site collection under the new managed path
3. Test the new site collection
4. Add a link back to <http://abcuniversity>

#### Demo Introduction

##### Demo: Create Site Collection

1. Enable Self-Service Site Collection Creation in Central Administration
2. Make a Biology Department site collection under the department managed path
3. Test the new site collection



### Demo Introduction

#### Demo: Create Site Collection

1. Create a Astronomy Department site collection using PowerShell
2. Test the new site collection

### Exercises

#### Perform Lab:

- All exercises from **Site Collection as an Administrator Lab**

Microsoft SharePoint 2010

## **Web Application Management**

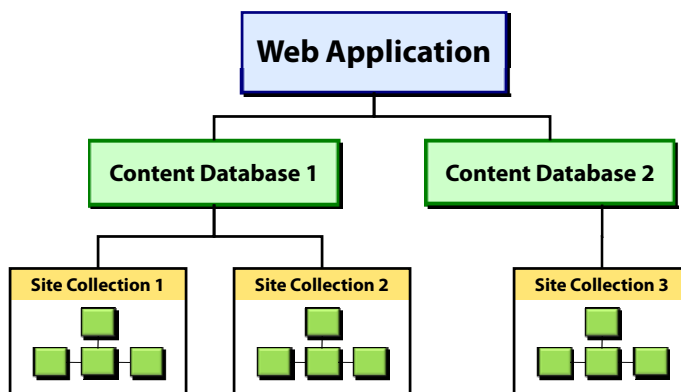
### **Web Application Management**

- **What a Web Application is**
  - » Functional description
- **Alternate Access Mappings**
  - » Zones
  - » Different URLs for the same content
- **Creating and extending Web Applications**

### Descriptions

- **Web Application**
  - » Exists in both SharePoint and IIS
  - » Has content databases which house site collections and site data
  - » Handles the URL and authentication information
  - » Allows features and other settings to be activated or selected at the web application level
- **Content Database**
  - » A set of tables in SQL that holds site collections and the content within them

### Illustration: Web Applications



### Creating and Extending Web Applications

- **Why to make a new web application**
  - » Distinct URL and authentication mechanism for new content (distinct web site)
- **Why to extend a web application**
  - » Different URL for the same content
- **Information to specify**
  - » Authentication type
  - » New or Shared Application Pool
  - » Port / Host Header
  - » Database server/Database name
  - » Service Applications the new Web App will be using

### Alternate Access Mappings

**Zone** - grouping of URLs so that links between web applications can be maintained properly

- » A web application can have one or more zones with different authentication mechanisms
- » **Zone names** - Default, Intranet, Extranet, Internet.
- **For a web application, a zone has:**
  - » **One Public URL** – rendered to the client as links on the pages.
  - » **One or many Internal URLs** - not returned, but may be received by the server either through a redirect or load balancer.
- **Proper configuration of AAM is important**

### Example: Alternate Access Mappings

- **Zone: Default**
  - » Authentication: Windows
  - » Public URL: abcuniversity
  - » Internal URLs: web1, web2 web3
- **Zone: Internet**
  - » Authentication: Forms with Anonymous
  - » Public URL: www.abcuniversity.com
  - » Internal URLs: web1.abcuniversity.com, web2.abcuniversity.com, web3.abcuniversity.com

### Authentication Mechanisms

- **Claims based**
  - » Allows for multiple providers within the same Web Application
  - » Forms Authentication
- **Windows**
  - » Active directory or local accounts (single server)
- **Basic**
  - » Clear text passwords sent over the network

### Demo Introduction

#### **Demo:** SharePoint Web Application Management

1. Create a new web application
2. Extend an existing web application
3. Add a content database to a web application
4. Create a web application with a host header without having to extend it

### Lab: SharePoint Web Application Management

#### **Perform Lab:**

- All exercises from **SharePoint Web Application Management**

Microsoft SharePoint 2010  
**IIS and SQL Server**

**IIS and SQL Server Investigated**

- SharePoint uses both Internet Information Services (IIS) and SQL Server
- What IIS does and when you should interact with it
- What SQL Server does and when you should interact with it

## IIS

- **What IIS knows about**
  - » Application pools
  - » Web applications
  - » Virtual directory mappings to the local disks
- **When you have to directly interact with IIS**
  - » Setting up a certificate for SSL

All management of application pools and IIS web sites **should be handled through Central Administration or the API** (including PowerShell)

## Application Pages Notes

**Application page** is a web page that is available across every site in SharePoint through \_layouts path:

- » [/\\_layouts/viewlsts.aspx](#)
- » [/courses/chem101/\\_layouts/viewlsts.aspx](#)
- IIS maps the [\\_layouts](#) directory to the [hive/template/layouts](#) folder in the local file system
- Application pages are not editable in SharePoint Designer
- Developers may create their own application pages to provide additional functionality across all sites



### Safe Page Review

**Safe pages** are web pages available **only** in sites where they are defined

- » IIS knows nothing about safe pages
- » Safe pages are editable in SharePoint Designer
- » Designers may modify safe pages
- » Safe pages can have web parts (safe controls) on them

### Demo Introduction

#### **Demo:** IIS Server Inspection

1. Examine a website in IIS. Notice that sites and sub-sites are not stored there.
2. Find “\_layouts” virtual directory
3. Examine the application pool settings
4. Examine Service Account Settings in Central Administration

### SQL Server

**SQL Server** provides storage for information in SharePoint sites

- **What is not stored in SQL Server**
  - » Application Pages, Site Definitions
    - Anything else in the hive
  - » Search Index
- **What is stored in SQL Server**
  - » All Content
  - » Farm Configuration Information
  - » Simply – everything not mentioned above as specifically not being housed there

### SQL Server Database Types

- **Configuration Database** – info about web farm configuration
- **Content Database (Site)** – sites, lists, and documents
- **Profile Database** – user profiles and audience targeting
- **Search Services database** – search configuration info
- **Single Sign On (SSO)** – credentials to access other line of business apps

### Content Databases Examined

- **What is stored in Content Databases**
  - » Sites
  - » Site Collections
  - » Lists
  - » Content
- **When you should modify SQL Server directly**
  - » Never. Do not change anything
  - » Read-Only access is OK, as some information is most easily and quickly gotten through a direct query

All interaction with Content Databases should be done through the API or the user interface

### Demo Introduction

#### **Demo:** Content Database Structure

1. Examine content database tables.
2. Create a new content database for a web application using the interface

## Freestyle: Content Databases

### Free Style Lab Exercise

1. Create a new content database for a web application using the interface
2. Now do it using PowerShell instead of the interface

Microsoft SharePoint 2010

## **User Profile Service Application**

### **User Profile Service Application**

- **User Profiles**
  - » Properties, Audiences
- **Synchronization with AD**
- **Organization Profiles**
- **My Sites**

## Profiles

**Profiles** consist of properties such as name, state, manager, etc

- » SharePoint stores profile information in the profiles database
- » Active directory is the source for some of the profile properties, others may be mapped if desired

## Synchronization

- **Synchronization** with Active Directory can be done by choosing which containers you would like synchronized
  - » In the Containers region you select those to be imported
- Synchronization with BCS objects can be set as well

### Demo Introduction

#### Demo: User Profile Management

1. Create user profiles manually
2. Create a new profile property
3. Walk through a connection to Active Directory

### Audiences - Overview

- Audiences allow content managers to target content to specific people based on the criteria in user profiles or group membership. Audiences are **not a substitute for security**.

example:

**Oklahoma IT**

State = OK and Department = IT

- Content managers can target pages, web parts, and lists to an audience

## Demo Introduction

### Demo: Audiences

1. Create and compile a new audience
2. Use the audience to target a web part
3. Use the audience to target a page

## Organization Profiles

**Organization Profiles** are like User Profiles, but for teams or groups within your organization instead of individuals

- » Conceptually they're the same thing, but for different entities
- » Can define a parent or individual to which the organization reports
- » Can be helpful when deducing who is in charge of a certain team



## Demo Introduction

### Demo: Organization Profiles

1. Create a new Organization Profile for the Math Team that reports to Dave Green.

## My Sites

**My Sites** are a personal space for end users

- » Each My Site is a site collection of which the user is an administrator of.

### Typical use cases:

- » sharing of documents not directly related to projects housed by SharePoint
- » Repository for documents to be backed up
- » custom personal portals
- » test bed for new ideas

### Demo Introduction

#### Demo: My Sites

1. Find the My Site for the logged in User
2. Examine the possible My Site settings in Central Administration

Microsoft SharePoint 2010  
**Search Service Application**

**Search Service Application**

**Search Architecture**

**Content Sources and Crawl Rules**

**Properties**

**FAST**

### Search Architecture

- **Web Server** - uses web parts to show search results
- **Query Server** - executes user searches against the search index
- **Search index** - a specialized data structure on the file system optimized for quickly finding information.
  - » Index can be partitioned across multiple Query Servers
- **Crawl Server** - crawls content and builds the search index
  - » Need not be a web front end
- **SQL Server** - hosts search related databases
  - » Property DB
  - » Crawl DB
  - » Admin DB

### Content Sources and Crawl Rules

- **Content Source** - a collection of start addresses to crawl
  - » Addresses can exist inside or outside of SharePoint
  - » Business Connectivity Services sources can be crawled as well
- **Crawl Rules** - disallow certain content from being crawled
  - » Use a different account to crawl certain content
    - An account with fewer permissions to avoid crawling minor versions
    - More permissions to index content that the default account has no access to

## Demo Introduction

### Demo: Content Sources and Crawl Rules

1. Add a file share content source
2. Add a crawl rule to not crawl Chemistry 101
3. Perform a full crawl and test results

## Search Properties

- **Crawled Properties**

- » Column or file property information (metadata) discovered by the crawl engine
- » cannot be targeted specifically for search

- **Managed Properties**

- » Crawled properties that have been promoted
- » Are searchable specifically

example:

Query: AssignmentType:homework  
Results: list items whose Assignment Type is homework

### Search Scopes

- **What they do**
  - » Improve Relevance
- **How they work**
  - » Append constraints to the query
  - » Force certain properties to match certain values

example:

Homework – Return results where AssignmentType:homework

### Demo Introduction

#### **Demo:** Search Properties

1. Add AssignmentType as a managed Property
2. Perform a full crawl
3. Add a new search scope for Homework

### **FAST Overview**

- **Intended Use**
  - » Very large organizations and companies who monetize end user search on the internet
- **Features (not exhaustive)**
  - » Improved metadata modeling
  - » Intelligent relationships between terms
  - » Thumbnail previews for certain document types
  - » Personalized search experiences
    - Different person, different results

Microsoft SharePoint 2010  
**Backup and Restore**

**Backup and Restore**

- **Importing and Exporting**
- **Backing up through Central Administration**
- **Backing up from PowerShell**
- **Restoration**
- **Unattached Content Database Restore**



## Export/Import

### Export and Import Scenarios (not exhaustive)

- » Move a site to an entirely new farm
- » Move a site into a new site collection
- » Take a site completely offline and archive it

## Demo Introduction

### Demo: Exporting and importing a site

1. Export a particular site from abcuniversity
2. Import it into the alumni web application
3. Check to ensure that the operation succeeded

### What to Back Up?

- **Content Databases**
  - » Most important
  - » Contains all of the information in SharePoint sites
- **Other Databases**
- **Configuration information in the file system**
- **Search Index Partitions**
  - » They live on the file system, so would not be covered by a SQL backup of Content Databases

### Backup and Restore Methods

Method	Scope	Comments
<b>SharePoint backup tool</b> - most comprehensive, and can be automated with PowerShell	SQL Database, Search Index, web application information	Configuration information in the file system not backed up. Provides the simplest interface for restoring.
<b>SQL Server Backup and Restore</b>	SQL Databases	Search index and configuration information in the file system not backed up. Works well in enterprise environments with SQL backup solution already in place and where SQL Administrators are responsible for backup
<b>SQL Server Mirroring</b>	SQL Databases	SQL Server feature that copies databases to a different SQL server in real time. Configuration information in the file system and search index not backed up.
<b>PowerShell Site Collection Backup</b>	Site collection	Only data for one site collection is backed up. If customizations in the file system are missing, the site will not work properly.

#### Demo Introduction

#### **Demo:** Backing up through Central Administration

1. Back up a portion of your environment in Central Administration

#### Demo Introduction

#### **Demo:** Backing up through PowerShell

1. Back up the whole farm using PowerShell

## Restoration

- **Complete**
  - » Restores everything from the backup
- **Partial**
  - » You may choose to restore any component down to the content database level that is included in a backup

## Demo Introduction

### **Demo:** Restoring a content database

1. Locate a content database from the backup and restore it through the interface.

### Unattached Content Database Restore

- **New in 2010**
  - » Previously you had to use third party tools to get granularity finer than the content database
- **Options**
  - » Browse a content database without attaching it to a web application
  - » Export a specific site or list from an unattached database

### Demo Introduction

#### **Demo:** Restore entities finer than a content database

1. Restore a specific list item
2. Export a site from an unattached content database

Microsoft SharePoint 2010

## **Site Definitions and Features**

### **Site Definitions and Features**

- **The 14 folder**
- **Site Definition definition**
- **Features**
- **Also in the hive**

### The Hive

- **Location**
  - » C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14
  - » Exists identically on every server in the farm
- **Folders of interest**
  - » Template\layouts – holds application pages
  - » Template\features
  - » Template\SiteTemplates
  - » Others are important in specific cases
- **Do not manually edit the files in these folders**

### Site Definition

- **Functionally equivalent to a site template**
  - » Defines a home page, lists and libraries, and features to activate
  - » Exist in the SiteTemplates folder under Template
- **Difference between a Site Definition and a Site Template?**
  - » You'd have to upload the \*.wsp file when selecting the root level site to use it there.
  - » Site Definitions would be available by default

## Demo Introduction

### Demo: Site Definitions

1. Locate the Hive
2. Find the SiteTemplates directory
3. Examine a site definition

## Features

- **Definition**
  - » A set of XML files defining some functionality in SharePoint
  - » You can create your own, buy from third parties, etc
- **Examples**
  - » Document Library
  - » Tasks List
  - » Content Types
  - » Publishing
- **Location**
  - » Template\features folder in the hive



## Demo Introduction

### Demo: Features

1. Find the features folder
2. Examine several features

## Installation

- Third parties or developers will deliver a Solution Package
  - » Depending if it's sandboxed or not, we can choose to install the features and site definitions (and application pages, etc) to either these folders, or just make them available to specific site collections
- You will use PowerShell to add the solution package to the solution store
  - » You are then free to use either the command line or interface to deploy

## Demo Introduction

### Demo: Solutions

1. Deploy a solution package to the farm

## Administration Conclusion

### Administration Conclusion

Microsoft SharePoint 2010

## **Development Overview**

### **Development Overview**

- Goals
- State of Development
- Development Components
- SharePoint API
- Types of Applications
- Application Integration
- Lists Manipulation and Location

### Goals

- Examine the most important components of the SharePoint API
- Build a solution using Visual Studio 2010 templates that utilizes different development components
- Realize the scope of SharePoint development

### State of Development

**The How To** – with full Visual Studio 2010 integration, SharePoint development is now straight forward. Creating and deploying components such as features and web parts has been streamlined.

**The Why and the Design** – what type of applications does it make sense to develop in SharePoint? Where to store the data and the best way to access it? This portion is still the difficult part of SharePoint development. It is easy to underestimate the scope of a project and not get much of it.

### Visual Development Components

- **Applications Page** – ASP.NET page accessible across all site collections.
- **Web Part** – ASP.NET server control with properties. Developers can include as part of the site template or users can within safe pages.
- **User Control** – a part of the page that can be used within ASP.NET pages or web parts.
- **Visual Web Part** – web part with a user control built-in. This is a proffered way to develop web parts.

### Event Development Components

- **Event Receiver** – code that runs before or after site and list events
- **Workflow Activity** – code that executes a single activity within a workflow. This is the simpler way to extend workflow functionality
- **Workflow Templates** – sequential and state machine workflow templates. This is the expensive way to extend workflow functionality.

### Configuration Development Components

- **List, Content Type, Site Definitions** – xml files that specify functionality using CAML and XSLT
- **Feature** – a collection of files that provides some functionality; includes an xml manifest file that describes the feature
- **Solution** – a collection of various SharePoint components; includes an xml manifest file that describes the solution

### SharePoint API

- SharePoint provides an extensive API for virtually all of its features.
- Developers can access the API many different ways:
  - » Server side through NET components such as web parts, event receivers, or application pages
  - » Client side through managed client object model or javascript client object model

### Types of Applications

- **Horizontal Application** – improves existing SharePoint functionality applicable to anyone who uses SharePoint. Example: Extra Usage Report or Workflow features.
- **Vertical Application** – builds on of SharePoint functionality to make SharePoint more applicable for a certain industry. Example: FDA Part 11 compliance for pharmaceutical or medical device makers.
- **Project specific** – customization that can't be accomplished out of box that are needed for specific project (maybe one or one extranet). Example. Custom registration process for an extranet or Foreign currency management for Treasury department.
- **Scripts** – one time or recurring tasks that need to be automated. This can be done in PowerShell.

### Application Integration

#### Existing applications:

- » If the application is working and has its data store outside of SharePoint, in most cases, it makes sense to leave as it.
- » SharePoint can provide entrance points to the application via links and Page Viewer web parts.
- » New components can be developed to retrieve data from existing data store to display in SharePoint either via custom development, BDC, or SharePoint Designer.

#### New applications:

- » the most important decision point is where to store the data which will discuss next. The user interface and business logic can be done with application pages, user controls, and web parts.

### List Storage

- Developers usually create applications where all database tables are on the same level.
- When storing data in SharePoint lists (conceptual tables), developers must decide at level within a site collection or within which site collection to store data.
- The decision on where to store the data affects how easy it is manage this data through the API (aggregation, joins, deletion, synchronization).

### List Storage

- **Site:**
  - » workflows store data within each site (workflow tasks and history lists). This makes sense since it is easy to control permissions for this data and take advantage of the list functionality. Aggregation works within one site collection, but much more difficult across site collections.
- **Site Collection:**
  - » web part gallery store data in one list for the entire site collection. This makes sense if the information is shared across the site collection.
- **Outside of site collections:**
  - » audit data (SPChange) stores data directly in one table within the content database. Does not take advantage of the list functionality. Easy to aggregate anywhere.



Microsoft SharePoint 2010

## **SharePoint API**

### **SharePoint API**

- How to access SharePoint API
- Working with SharePoint Foundation API
  - » Site collection organization
  - » List manipulation and data retrieval
  - » Security
  - » Administration
- Working with SharePoint Server API
  - » User profiles
  - » Search
- MSDN resources and SharePoint SDK

### Ways to test the API

- Application pages are an easy way to test the API
- Client object model has mostly the same classes and methods as the server site API and can be tested in console applications or **ECMAScript** in application pages.

### Where to Place Application Pages

- Directly in **\_layouts** folder (**Local\_Drive:\program files\common files microsoft shared\web server extensions\14\template\layouts**) as ASPX page with inline code or scrip element
- In **\_layouts/<sub\_folder>**, either as ASPX page, ASP.NET web site, or ASP web application (This is what VS2010 SharePoint templates will do by default)
- Another virtual directory within a web application. However, the resulting pages will not have characteristics of **\_layouts** application pages.

### Demo Introduction

#### Demo: Accessing the SharePoint API

1. Create application pages to access SharePoint api
2. Change web.config to show errors

### Site Collection Organization

In this section we will cover the basics of working with site collections:

- » Getting lists for the website
- » Getting all websites in a site collection

### Use Dispose To Free Up Memory

In order to avoid retaining the objects in memory.

#### Use

```
foreach(SPSite site in sites)
{
    //code
    site.Dispose();
}
```

#### Or

```
for(i = 0; i < sites.Count; i++)
{
    using(SPSite site = sites[i])
    {
        //code
    }
}
```

### Demo Introduction

#### Demo: Site Organization Classes

1. Use SPContext, SPSite, SPWeb, SPWebCollection, SPListCollection, and SPList classes

### List Manipulation

In this section we cover how to create new data:

- » Create a list
- » Create columns in the list
- » Modify a view for the list
- » Add new list item for the list

### Demo Introduction

#### **Demo:** List Data Creation

1. Use SPList, SPField, SPView, and SPListItem classes

### List Data Retrieval

In this section we cover how to retrieve list data:

- » Use SPQuery and CAML
- » Use SPSiteDataQuery
- » Use inline code to generate JQuery HTML

### Demo Introduction

#### **Demo:** List Data Retrieval

1. Use SPQuery, SPSiteDataQuery, and SPGridclasses

## Security

In this section we cover how to work with security features:

- » Checking if the website has unique permissions
- » Listing all domain users and groups
- » Listing all SharePoint groups
- » Listing all Permissions levels
- » Setting unique permissions for a folder

## Demo Introduction

### Demo: Security Classes

1. Use SPGroup, SPUser, SPRoleAssignment, and SPRoleDefinition classes

## Administration

In this section we cover how to work with Administration objects:

- » Listing all SharePoint web applications
- » Listing all content databases for a web application
- » Listing all site collections for a content database

## Demo Introduction

### **Demo:** Administration Classes

1. Use SPWebService, SPFarm, SPWebApplication, and SPContentDatabase classes



#### Demo Introduction

#### Demo: User Profile Classes

1. Use UserProfileManager and UserProfile classes from Microsoft.Office.Server.UserProfiles namespaces

#### Demo Introduction

#### Demo: Search Classes

1. Use FullTextSqlQuery class from Microsoft.Office.Server.Search.Query namespace

### Demo Introduction

#### **Demo:** MSDN Resources and SharePoint SDK

1. Examine MSDN
2. Examine SharePoint SDK

### Exercise

#### **Perform Lab:**

- All exercises from **SharePoint API Development**

Microsoft SharePoint 2010

## **Elevation and Impersonation**

### **Elevation and Impersonation in SharePoint**

Developers use **elevation** when an application needs to perform a task for which the current user does not have permissions

**example:**

- accessing a master list on the WSS site on which the user might not be a member
- creating a list when a user only has reader privileges
- using windows authentication to access SQL database for which the currently logged in user does not have permissions

### Two Methods of Elevation

```
SPSecurity.RunWithElevatedPrivileges(delegate()  
{  
    using (SPSite site = new SPSite(web.Site.ID))  
    {  
        // do things assuming the permission of the "system account";  
    }  
});
```

**OR**

```
SPUser user = web.AllUsers["@SHAREPOINTSYSTEM"];  
SPUserToken userToken = user.UserToken;  
using (SPSite site = new SPSite(web.Url, userToken))  
{  
    // do things assuming the permission of the "system account";  
}
```

### Demo Introduction

#### Demo: Elevation

1. Use both methods of elevation

### Authentication Models for Accessing External Data

**Trusted Subsystem** - the application (middle tier) authenticates with fixed identity

- » Offers database connection pooling.
- » Is less complex.
- » The group that owns and manages the back end gives access to one account that they manage.

**Impersonation and Delegation** - the application (middle tier) impersonates the client and authenticates to back-end on client's behalf

- » To enable auditing at the back end.
- » If there is per-user authorization at the back end.

### Connection String Options: SQL Authentication

#### SQL Authentication:

server=training; uid=sa; pwd=Pilot; database=ABCPurchasing

**Advantage:** easy to use, no special requirements.

**Disadvantage:** username and password are clear text

### Connection String Options: Windows Authentication

#### Windows Authentication

**example:**

Integrated Security = SSPI; server=training; database = ABCPurchasing

**Advantage:** username and password are not clear text.

**Disadvantage:** if application runs as a currently logged in user, that user must have access to DB.

**Windows Authentication with impersonation** of the application pool user

**Advantage:** uses application pool account to access db.

**Disadvantage:** no significant disadvantages

### Application Pool Account Impersonation Details

```
using System.Security.Principal

// revert to self
WindowsImpersonationContext wic = WindowsIdentity.Impersonate(IntPtr.Zero);
try
{
    // perform db operations
}
finally
{
    wic.Undo(); // resume impersonating
}
```

### Demo Introduction

#### **Demo:** Web Part To Access DB without SP Template

1. Create a class library project
2. Change the class to extend a webpart
3. Deploy the web part to web application bin folder  
(\Inetpub\wwwroot\wss\VirtualDirectories\<PortNumber>\bin)  
and update web.config safe controls sections
4. Add a web part to the site

### Links

- How to implement impersonation in ASP.NET application:  
<http://support.microsoft.com/?id=306158>
- ASP.NET Impersonation:  
[http://msdn.microsoft.com/en-us/library/xh507fc5\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/xh507fc5(VS.85).aspx)

Microsoft SharePoint 2010

## **Client APIs**

### **Client API**

- **Client API**
  - » Interact with SharePoint from script in the browser
  - » Interact with SharePoint from managed code on computers that do not have SharePoint installed.
- **Details**
  - » Subset of the server side API
  - » Class names are mostly the same



### Demo Introduction

#### Demo: Console Client

1. Build a console application to test out client API

### Demo Introduction

#### Demo: ECMAScript Client

1. Build an application page with ECMAScript Client

Microsoft SharePoint 2010

## **Simple Application Visual Studio 2010**

### **Simple Application with Visual Studio 2010**

- Development Process
- Sandboxed vs. Farm Solutions
- Application Pages and User Controls
- Visual Web Parts
- List and Site definitions
- Event Receivers

### Development process

- **Create a Visual Studio SharePoint project** – there are many different types, choose the type of deployment (sandboxed or farm), and choose a site to be used for testing
- **Make changes** to code and xml manifest and schema files
- **Build** – Visual Studio compiles all the code files into DLLs
- **Deploy** – by default Visual Studio does the following
  - » creates a solution from all the files in the project
  - » deploy this solution to SharePoint (retract if necessary first)
  - » recycle the application pool
  - » activate features

### Sandboxed vs. Farm Solutions

**Sandboxed** – code deployed to a specific site collection

- » does not need Farm Administrator approval
- » if code causes problems, it is isolated within a site collection
- » certain components such as application pages can not be included – anything that maps to HIVE

**Farm** – deployed farm wide

- » needs Farm Administrator to install
- » no limitations
- » can bring down the entire server if code causes problems

### Application Pages and User Controls Deployment Notes

- **Applications Page**
  - » project DLL goes to GAC (Global Assembly Cache)
  - » aspx files go to `Hive\template\layouts\<project_name>`
- **User Control**
  - » project DLL goes to GAC
  - » ascx files to `Hive\template\controltemplates \<project_name>`  
can be used in application page or visual web part
- Cannot be deployed as sandboxed
- Notice that project DLL deployment to GAC requires an Application Pool reset
- Post build events that copy files from the project folder to appropriate folders in the HIVE work for ASPX files only (not code behind) as long as VS project variable is replaced

### Demo Introduction

#### Demo: Application Pages and User Controls

1. Create an empty SharePoint project in VS2010
2. Create an application page and create a user control (map layouts and controltemplates folders).
3. Examine feature and solution files generated
4. Examine SharePoint Project Deployment settings and deploy the solution
5. Examine the results of deployment
6. Change Post Build events

### Demo Introduction

#### Demo: Custom UI Actions for Application Pages

1. Add a custom UI Action to List Settings
2. Add a custom UI Action to the Documents Ribbon

### Web Parts Deployment Notes

- Web Part Project DLL goes to GAC – requires application pool recycle
- For Visual Web Parts, User controls go to  
`Hive\templates\controltemplates\<project_name>`
- Web parts with user controls cannot be deployed as sandboxed
- Deployment changes web application web.config <SafeControls> section

```
<Configuration>
  <SharePoint>
    <SafeControls>
      <SafeControl Assembly="HelloWorld" Namespace="HelloWorld" TypeName="*" />
    </SafeControls>
  </SharePoint>
</Configuration>
```

### Demo Introduction

#### Demo: Visual Web Parts

1. Add a Visual Web Part to SharePoint Project and make changes to the user control
2. Deploy the project and observe the results
3. Change Post Build events for user control
4. Access external data from a web part using impersonation within a user control

### List and Site Definitions

- List, List Instance and Content Type definitions can be sandboxed
- Where are they deployed I don't see anything in features folder.
- Site Definitions are deployed to `HIVE\template\SiteTemplates` folder and can not be sandboxed

### Demo Introduction

#### Demo: List Definition

1. Add a list definition with custom columns
2. Update application pages to reflect our list definition
3. Deploy the solution and examine the results

### Event Receivers

- Deployed to GAC – requires application pool recycle
- Connected to lists, content types, or sites via:
  - » Feature definitions
  - » API calls
- Execute before or after an event occurs
- Run in the context of the originating user. Elevation and impersonation can take place.

### Demo Introduction

#### Demo: Event Receivers

1. Add several different event receivers
2. Deploy the solution and examine results

### Exercise

#### Perform Lab:

- All exercises from **Audit Application Using SharePoint Templates**



Microsoft SharePoint 2010

## **Workflow Activity**

### **Workflow Activity**

- **Workflow Activity** is a component that enables a user to perform a specific task in a SharePoint workflow
- The advantage of building workflow activities over the entire workflow template is that we can easily use them in SharePoint Designer.

### Demo: Introduction

#### Demo: Workflow Activity

1. Build a workflow activity in Visual Studio
2. Deploy it
3. Use in Designer