SharePoint 2007 Overview

Topics which will be covered…

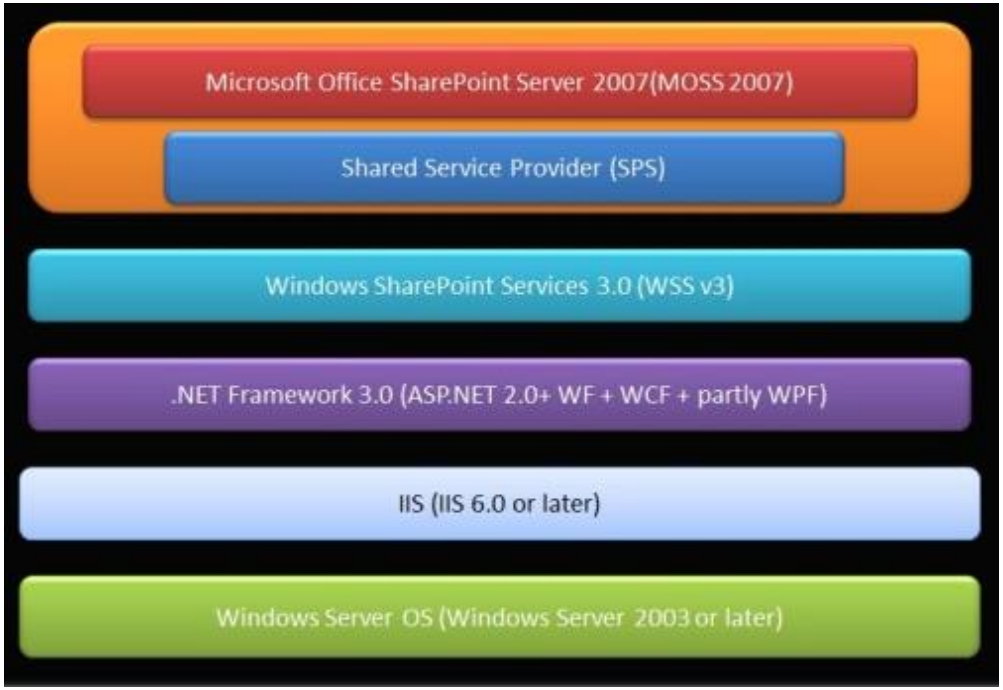
- What is SharePoint?
- Architecture
- Six Pillars of MOSS
- Explanations of Six Pillars
- Central Administration
- SharePoint API
  - Working with SharePoint Object Model
  - Working with SharePoint Web Services
- SharePoint Best Practices

What is SharePoint?

- SharePoint 2007 is a platform where user can **share Data**, **Collaborate** and people can also take this platform and customized platform as per their requirements.
- In Other words, it is a browser based platform provided by Microsoft that you can use for Collaboration as well as many other capabilities such as
    - Enterprise Content Management
    - Portal
    - Form Driven Business Processes
    - Business Intelligence
    - Search

SharePoint 2007 Architecture

- Windows SharePoint Services 3.0 comes equipped with Windows Server 2003 OS which provides the basis of SharePoint 2007.
- It requires ASP.NET 2.0, .NET Framework 3.0 along with IIS 6.0 or later along with other software prerequisites.
- The back-end database server computer must be running Microsoft SQL Server 2005 or Microsoft SQL Server 2000 with Service Pack 3 (SP3) or later.
- MOSS 2007 was built with SSP (Shared Service Provider) on top of WSS v3 which helped to improve and place addition to all existing WSS v3 features.



Microsoft Office SharePoint Server 2007(MOSS 2007)

Shared Service Provider (SPS)

Windows SharePoint Services 3.0 (WSS v3)

.NET Framework 3.0 (ASP.NET 2.0+ WF + WCF + partly WPF)

IIS (IIS 6.0 or later)

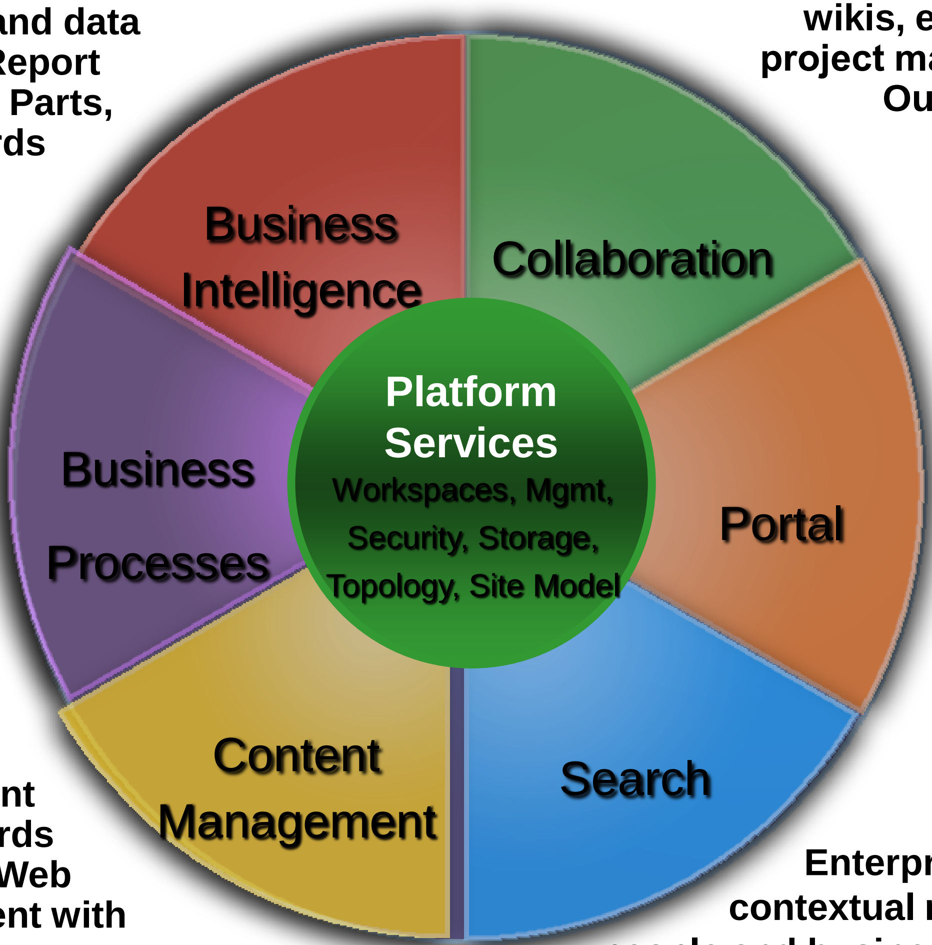
Windows Server OS (Windows Server 2003 or later)

Functional Architecture/ Six Pillars of MOSS

Server-based Excel spreadsheets and data visualization, Report Center, BI Web Parts, KPIs/Dashboards

Docs/tasks/calendars, blogs, wikis, e-mail integration, project management “lite”, Outlook integration, offline docs/lists

Rich and Web forms based front-ends, LOB actions, pluggable SSO



Enterprise Portal template, Site Directory, My Sites, social networking, privacy control

Integrated document management, records management, and Web content management with policies and workflow

Enterprise scalability, contextual relevance, rich people and business data search

Six Pillars of MOSS

1. Collaboration

By integrating Workspaces, Tasks, Forums, Surveys, Blogs, RSS and Wikis…..

2. Form Driven Business Processes

Microsoft has overhauled this aspect of SharePoint with XML driven InfoPath forms that are available on a variety of platforms….

3. Portal

A one stop site for everything enterprise-related.

Six Pillars of MOSS

Enterprise Content Management

- Document Management -> Records Management, Web Content Management, e-forms/workflow
- Compliance and multi-lingual scenarios
- Rich and seamless Office integration
- Key Technologies
  - Content Types
  - Integrated Workflow
  - Internet scale WCM
  - Records Management with policies and auditing
  - Integrated rights management

Six Pillars of MOSS

Enterprise Content Management

Document Management

Forms Automation

Records Management

Web Content Management

Six Pillars of MOSS

Document Management

- Scalable Document Libraries and Repositories
  - Extensible content types with rich metadata
  - Check-in/check-out with comprehensive versioning
  - Large document repositories and specialized library templates

- Workflow and Approval
  - Out of the box, serial and parallel workflows (e.g. “Document review/approval” and “Signature collection”)
  - Participate & initiate workflows from within preferred application
  - Customizable workflows built with SharePoint Designer or Visual Studio 2005 on the Windows Workflow Foundation

Six Pillars of MOSS

Search

- Breaking through silos of information
- Find people and expertise
- Easily connect to backend business systems
- Key Technologies
  - Search improvements – relevancy, hit highlighting, query correction
  - Business Data Catalog
  - Customizable Search center
  - Rich people search

Six Pillars of MOSS

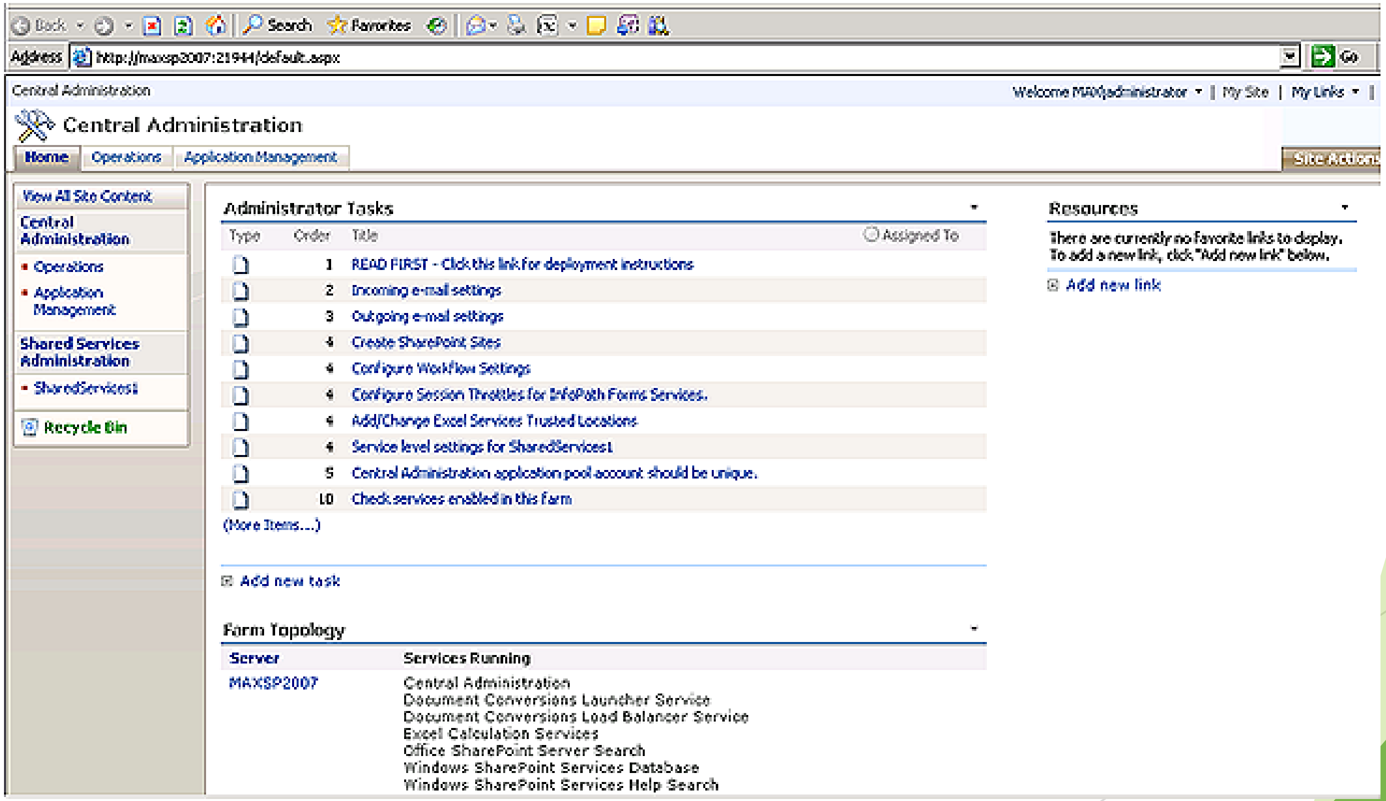
Business Intelligence

- The glue for “BI for the Masses”
- Solves information under load by connecting clients and servers
- Delivers in-context BI
- KPIs, Reports, Dashboards
- Key technologies
  - Excel Services
  - Report Center
  - KPIs and KPI Lists
  - Business Data Catalog

Central Administration

Central Administration Web Site

- You can use Central Administration to perform administration tasks for SharePoint Products and Technologies from a central location. Central Administration is divided into three parts: the **Home page**, the **Operations page** and the **Application Management page**.



Administrator Tasks

Type	Order	Title	Assigned To
	1	READ FIRST - Click this link for deployment instructions	
	2	Incoming e-mail settings	
	3	Outgoing e-mail settings	
	4	Create SharePoint Sites	
	4	Configure Workflow Settings	
	4	Configure Session Throttles for InfoPath Forms Services.	
	4	Add/Change Excel Services Trusted Locations	
	4	Service level settings for SharedServices1	
	5	Central Administration application pool account should be unique.	
	10	Check services enabled in this farm	

(More Items...)

Add new task

Resources

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

Add new link

Farm Topology

Server	Services Running
MAXSP2007	Central Administration Document Conversions Launcher Service Document Conversions Load Balancer Service Excel Calculation Services Office SharePoint Server Search Windows SharePoint Services Database Windows SharePoint Services Help Search

Central Administration Web Site

In Short……….

- Central Administration is the place where you will be performing/configuring almost all your administrative tasks for a SharePoint environment.
- From starting services to backing up and restoring your SharePoint servers will be done through this page.
- There are tasks such as Topology and Services, Security Configuration, Logging and Reporting, Upgrade and Migration etc. found under the **Operations** Tab that can be used to perform several tasks as required.
- There are tasks such as SharePoint Web Application Management, SharePoint Site Management, Search, InfoPath Forms Services etc. found under the **Application Management** Tab that can be used to perform several tasks as required.
- In addition to the above, the Central Administration will be hosting the Shared Services Provider settings as well. these are a common set of settings which will be shared across your SharePoint.

Programming with SharePoint 2007

Programming with SharePoint 2007

Programming with SharePoint exposes Workarounds as per deployment of code concerns. Depending upon the requirement itself the code may reside on the server side which is being denoted as the
SharePoint Server Object Model
or if we need to access SharePoint server and its objects from the client end which will result in creating calls to specific **web services** and using its exposed web methods.

- **SharePoint Server Object Model**

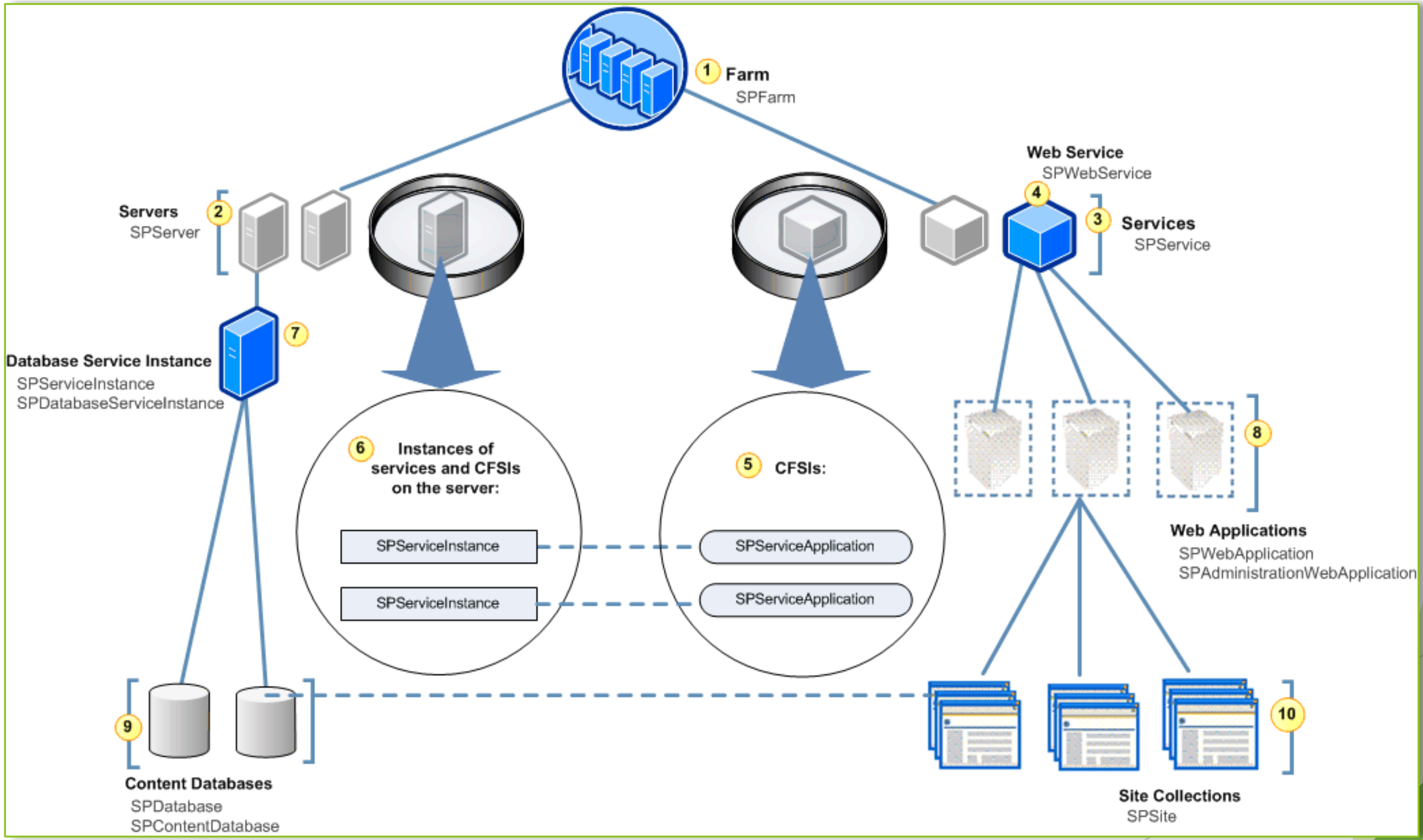
  Microsoft SharePoint Foundation offers a highly structured server-side object model that makes it easy to access objects that represent the various aspects of a SharePoint Web site. From higher-level objects, you can drill down through the object hierarchy to obtain the object that contains the members you need to use in your code.

Programming with SharePoint 2007

References:-

We need mainly two dlls to be referenced for accessing WSS Objects.

- Microsoft.SharePoint
- Microsoft.SharePoint.Administration



The following diagram shows the SharePoint Foundation server architecture in relation to the collections and objects of the Microsoft.SharePoint.Administration

Programming with SharePoint 2007

- The SPFarm object is the highest object within the SharePoint Foundation object model hierarchy. The Servers property gets a collection representing all the servers in the deployment, and the Services property gets a collection representing all the services.

- Each SPServer object represents a physical server computer. The ServiceInstances property provides access to the set of individual service instances that run on the individual computer.

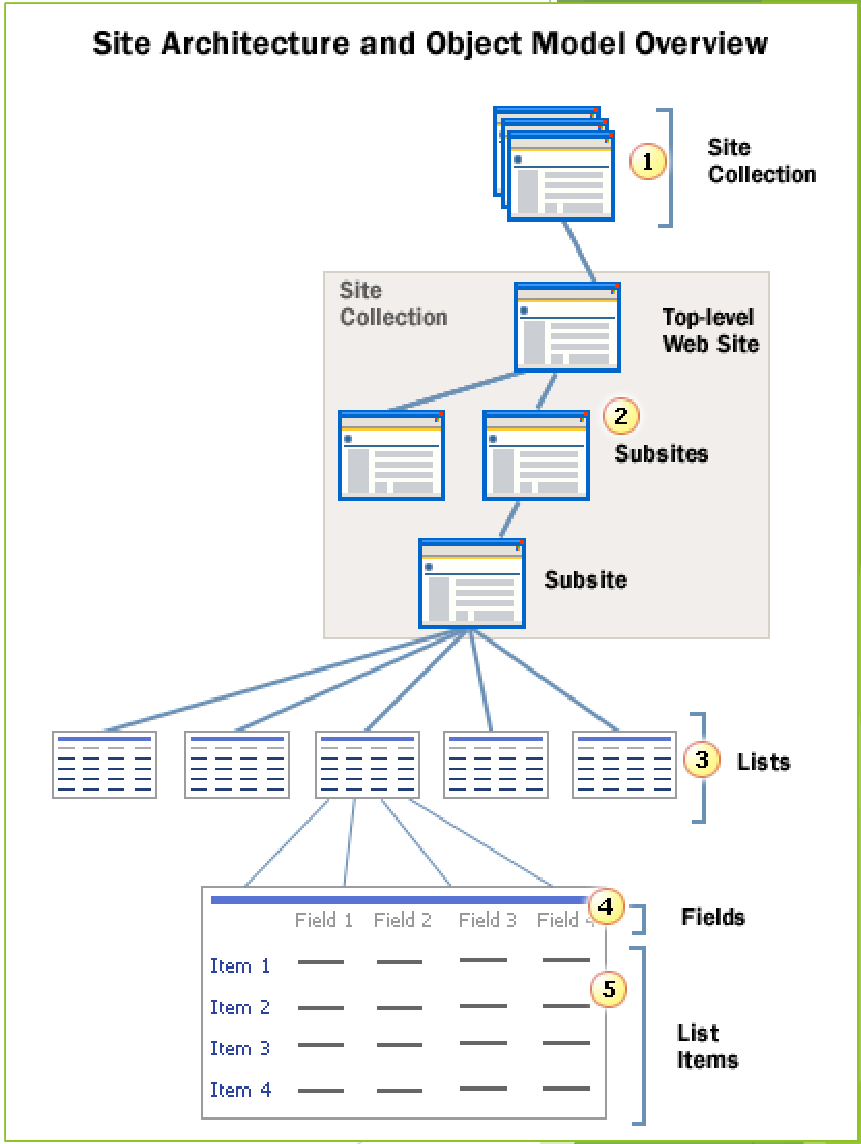
- Each SPService object represents a logical service installed in the server farm. Derived types of the SPService class include, for example, objects for Windows services, such as the timer service, search, the database service, etc. and also objects for Web services, such as the basic content publishing Web service which supports the Web applications.

- An SPWebService object provides access to configuration settings for a specific logical service or application. The WebApplications property gets the collection of Web applications that run the service.
- If the service implements the Service Application Framework of SharePoint Foundation, then it can be split into multiple configured farm-scoped instantiations (CFSIs). Each of these provides the functionality of the service but each has its own individual permission and provisioning settings.
- Each instance of a service, or a CFSI, that is running on a specific server is represented by an SPServiceInstance object.

- An SPDatabaseServiceInstance object represents a single instance of a database service running on the database server computer. The SPDatabaseServiceInstance class derives from the SPServiceInstance class and thus inherits the Service property, which provides access to the service or application that the instance implements. The Databases property gets the collection of content databases used in the service.
- Each SPWebApplication object represents a Web application hosted in an Internet Information Services (IIS) Web site. The SPWebApplication object provides access to credentials and other farm-wide application settings. The Sites property gets the collection of site collections within the Web application, and the ContentDatabases property gets the collection of content databases used in the Web application.
- An SPContentDatabase object inherits from the SPDatabase class and represents a database that contains user data for a Web application. The Sites property gets the collection of site collections for which the content database stores data, and the WebApplication property gets the parent Web application.
- An SPSiteCollection object represents the collection of site collections within the Web application.

Programming with SharePoint 2007

- Each SPSite object, despite its singular name, represents a set of logically related SPWeb objects (see below). Such a set is commonly called a "site collection".
- The AllWebs property provides access to the SPWebCollection object that represents the collection of all Web sites within the site collection, including the top-level site. The SPSite.OpenWeb method of the SPSite class returns a specific Web site.
- Each site collection includes any number of SPWeb objects, and each object has members that can be used to manage a site, including its template and theme, as well as to access files and folders on the site. The Webs property returns an SPWebCollection object that represents all the subsites of a specified site, and the Lists property returns an SPListCollection object that represents all the lists in the site.
- Each SPList object has members that are used to manage the list or access items in the list. The GetItems method can be used to perform queries that return specific items.
- The Fields property returns an SPFieldCollection object that represents all the fields, or columns, in the list, and the Items property returns an SPListItemCollection object that represents all the items, or rows, in the list.
- Each SPField object has members that contain settings for the field.
- Each SPListItem object represents a single row in the list.



Programming with SharePoint 2007

- Code Samples

Using SPSite, SPWeb and SPList Objects

```
using(SPSite oSiteCollection = new SPSite("http://Server_Name"))
{
    using(SPWeb oWebsite = oSiteCollection.OpenWeb("Website_URL"))
    {
        using(SPWeb oWebsiteRoot = oSiteCollection.RootWeb)
        {
           ...YOUR CODE GOES HERE
        }
    }
}
```

```
SPSite oSiteCollection = SPContext.Current.Site;
using(SPWebCollection collWebs = oSiteCollection.AllWebs)
{
    foreach (SPWeb oWebsite in collWebs)
    {
        SPListCollection collSiteLists = oWebsite.Lists;
        foreach (SPList oList in collSiteLists)
        {
            Label1.Text += SPEncode.HtmlEncode(oWebsite.Name) + " :: " +
                SPEncode.HtmlEncode(oList.Title) + "<BR>";
        }
        oWebsite.Dispose();
    }
}
```

Programming with SharePoint 2007

- Code Samples

Using SPQuery Object

```
SPWeb oWebsiteRoot = SPContext.Current.Site.RootWeb
SPList oList = oWebsiteRoot.Lists["Tasks"];

SPQuery oQuery = new SPQuery();
oQuery.Query = "<Where><Eq><FieldRef Name='Status'/>" +
    "<Value Type='Text'>Completed</Value></Eq></Where>";
SPListItemCollection collListItems = oList.GetItems(oQuery);

foreach (SPListItem oListItem in collListItems)
{
    Response.Write(SPEncode.HtmlEncode(oListItem["Title"].ToString()) +
        "<BR>");
}
```

Programming with SharePoint 2007

- Code Samples

```
SPSite oSite = new SPSite("http://sigr8-1b:1000/sites/MyTestSite");
        SPWeb oWeb = oSite.OpenWeb();
        SPList oList = oWeb.Lists["MyCustomList"];

        SPListItem oListItem = oList.Items.GetItemById(1);

        SPListItemVersionCollection oVersionInfo = oListItem.Versions;

        int i=0;

        foreach (SPListItemVersion oVersion in oVersionInfo)
        {
          if (oVersion.Level == SPFileLevel.Published)
          {
                // here we retrieve all metadata properties
                // for this item with the current approved version.
                // after taking these values we can export them
                Console.WriteLine(oListItem.Versions[i]["Address"].ToString());
                Console.WriteLine(oListItem.Versions[i]["Title"].ToString());
                Console.ReadLine();
                return;
             // after getting the latest approved version
             // we are exiting the code
          }
          i++;
        }
```

Working with Web Services

Programming with SharePoint 2007

- Code Samples

```
Web_Reference_Folder.Lists listService = new Web_Reference_Folder.Lists();
listService.Credentials= System.Net.CredentialCache.DefaultCredentials;

XmlDocument xmlDoc = new System.Xml.XmlDocument();

XmlNode ndQuery = xmlDoc.CreateNode(XmlNodeType.Element,"Query","");
XmlNode ndViewFields =
    xmlDoc.CreateNode(XmlNodeType.Element,"ViewFields","");
XmlNode ndQueryOptions =
    xmlDoc.CreateNode(XmlNodeType.Element,"QueryOptions","");

ndQueryOptions.InnerXml =
    "<IncludeMandatoryColumns>FALSE</IncludeMandatoryColumns>" +
    "<DateInUtc>TRUE</DateInUtc>";
ndViewFields.InnerXml = "<FieldRef Name='Field1' />
    <FieldRef Name='Field2'/>";
ndQuery.InnerXml = "<Where><And><Gt><FieldRef Name='Field1'/>" +
    "<Value Type='Number'>5000</Value></Gt><Gt><FieldRef Name='Field2'/>" +
    "<Value Type=
        'DateTime'>2003-07-03T00:00:00</Value></Gt></And></Where>";
try
{
    XmlNode ndListItems =
        listService.GetListItems("List_Name", null, ndQuery,
        ndViewFields, null, ndQueryOptions, null);
    MessageBox.Show(ndListItems.OuterXml);
}
```

SharePoint Best Practices

Programming with SharePoint 2007

Best Practices

- While working with SharePoint object model, most of developers will use SPWeb, SPSite, SPList objects intensively. Using "**Using**" Clause for exposing IDisposable interface that Disposes SPSite or SPWeb object instances automatically.

- Using SPSecurity.RunWithElevatedPrivileges Delegate function to avoid any security issue inside the code and to run the code block with **Application Pool's Credentials**.

- Using SPQuery instead of using foreach loop to get SPListItemCollection.

- Also using the RowLimit Property of SPQuery object to return a specific number of List Items instead of pulling All the List Items.

Thanks...................

Techhelpgo.wordpress.com
Contact Me at-
Sunayan.sarkar2014@gmail.com
Mobile:- +91-7204097535