SPIF 1.1 for CMS 6.6 Update-5 or later

SharePoint Integration Framework Developers Cookbook

A Guide to Integrating Sitecore and SharePoint

Table of Contents

Chapter 1

Introduction

The SharePoint Integration Framework enables you to display SharePoint lists in real time on a Sitecore website. It includes 'out of the box' customizable sample controls for page level integration and a SharePoint Integration wizard for item level integration. Choose one of these approaches to integrate SharePoint content with Sitecore.

The Integration Framework also includes an API for developers who want to customize the framework. Depending on your business requirements, choose the approach that best suits your needs.

This cookbook is for Sitecore partners and developers and includes useful tips for developers with examples.

This document contains the following chapters:

- **Chapter 1 — Introduction**
  This introduction contains a description of the content, aims, and the intended audience of this cookbook.
- **Chapter 2 — The SharePoint Integration Framework**
  This chapter provides an overview of the SharePoint Integration Framework architecture and main component parts.
- **Chapter 3 — Security and Authentication**
  This chapter gives systematic instructions on how to configure Sitecore and SharePoint to use the SharePoint Integration Framework for the first time.
- **Chapter 4 — Page Level Integration**
  This chapter explains how to use the sample controls included in the SharePoint Integration Framework to integrate SharePoint and Sitecore content in real time.
- **Chapter 5 — Item Level Integration**
  This chapter explains how to use the SharePoint Integration wizard to integrate SharePoint items with Sitecore content items.
- **Chapter 6 — Integration Scenarios**
  This chapter includes several fictitious scenarios to demonstrate how to implement page or item level integration in a typical business context. Each walkthrough includes systematic instructions to guide you through the integration process.

Chapter 2

The SharePoint Integration Framework

This section is an introduction to the basic concepts and components used in the SharePoint Integration Framework.

This chapter contains the following sections:

- Overview
- Architecture

2.1 Overview

The SharePoint Integration Framework provides Sitecore developers with a flexible and customizable development framework to integrate SharePoint and Sitecore content.

The framework offers you three possible approaches to integration:

- Page level integration
- Item level integration
- API integration

Page Level Integration

Use renderings and sub layouts to integrate SharePoint content in real time. The sample controls in the SharePoint Integration Framework are examples that you can customize:

- SharePoint List
- SharePoint Announcements
- SharePoint Tasks
- SharePoint Search

For more information about each control, see Chapter 4: Page Level Integration

Item Level Integration

Use the SharePoint Integration wizard to:

- Create Sitecore content items bound to SharePoint list items.
- Create field mappings, configure options, such as bidirectional integration, and scheduled BLOB transfer.
- Integrate items in real time or set an interval between updates.

For more information about Item Level Integration, see Chapter 5: Item Level Integration

API Integration

The API contains the following class groupings:

- Object Model
  i) SharePoint objects
  ii) Connector (SharePoint web service wrappers)
- Integration Providers
  i) SharePoint Item Provider
  ii) SharePoint Provider
- Integration Pipelines

For more information about the API, see the *SPIF API Reference* document.

Some developer options using the SharePoint Integration Framework:

- Use or customize the sample controls to display your own SharePoint lists.
- Create your own controls to display standard or custom SharePoint list items.

- Use the Item Provider class to represent SharePoint lists as Sitecore content items.
- Use pipelines to customize item level integration.
- Integrate with custom SharePoint list types.
- Extend the framework using SharePoint Web services.

Note
In this document, the term *connector* refers to the `Sitecore.Sharepoint.ObjectModel.Connectors` class used by the integration module to connect to SharePoint Web services. You should not confuse this with the Sitecore *SharePoint Connector* module, which was an earlier way of integrating with SharePoint.

2.2 Architecture

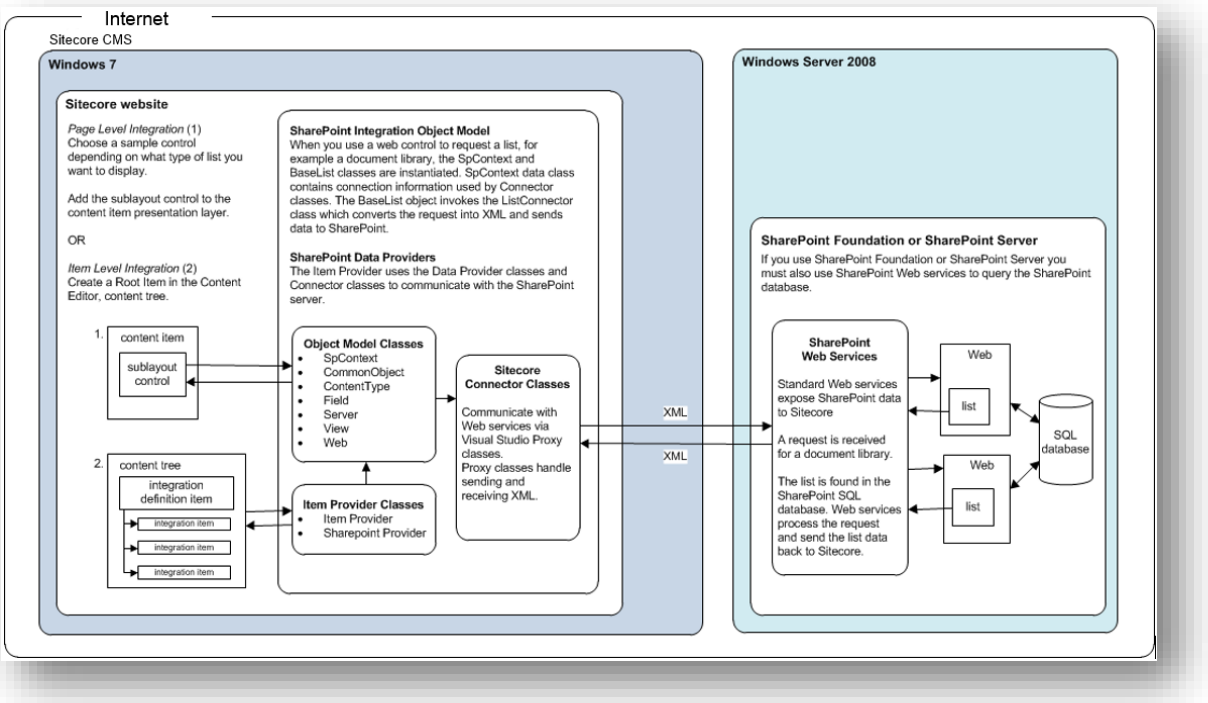
The SharePoint Integration Framework enables you to create a real-time connection between Sitecore and SharePoint. Integrate Sitecore content items and SharePoint lists using the Sitecore Integration Object Model.

There is a typical integration scenario:

- Integrate Sitecore CMS and SharePoint on a corporate extranet. Create Sitecore items bound to SharePoint list items in real time or specify how often items are updated. Use Sitecore functionality such as publishing and workflow.

  Recommended approach: item level integration

The following diagram gives a simple overview of SharePoint Integration, showing how the Integration Object Model interacts with SharePoint Web services using XML to transfer data.



Chapter 3

Security and Authentication

Before you install the SharePoint Integration Framework for the first time, it is important to understand some basic security and authentication concepts. You must also perform some additional configuration steps in SharePoint and Sitecore.

This chapter contains the following sections:

- Software Requirements
- Security and Authentication

3.1 Software Requirements

The SharePoint Integration Framework requires:

- WIF (Windows Identity Foundation) on the server with SPIF

The SharePoint Integration Framework supports:

- Sitecore CMS 6.6 Update-5 or later
- Windows SharePoint Services (WSS 3.0) or Microsoft Office SharePoint Server (MOSS) 2007
- SharePoint Server or SharePoint Foundation 2010
- SharePoint Server or SharePoint Foundation 2013
- SharePoint Online

To install the SharePoint Integration Framework, use the Sitecore Installation wizard. For more information about installation, see the *SharePoint Integration Framework Installation Guide* on the Sitecore Developers Network.

3.2 Security and Authentication

After you have installed the SharePoint Integration Framework, choose an authentication method and enable communication between Sitecore and SharePoint.

In page level integration, you continue to work directly with SharePoint lists so have continuous access to both the SharePoint server and the Sitecore server.

In item level integration, when you run the integration wizard or update items you have access to both Sitecore and SharePoint, otherwise you only work with Sitecore items.

Note
It is possible to set up page level Integration to use the system account to access SharePoint (for on-premises solutions).

3.2.1 Sitecore and User Authentication

To allow a Sitecore website to communicate with a SharePoint server you must first understand how Sitecore and the SharePoint Integration Framework handles authentication and then configure the appropriate permissions in IIS and the `sharepoint.config` file.

Use the Sitecore security layer to control security and permissions in Sitecore.

For more information on configuring Sitecore security, see the *Security Administrators Cookbook* on the Sitecore Developers Network.

Active Directory and Single Sign On for On-premises Solutions

Single sign on means that you only need to enter your credentials once to access Sitecore and SharePoint lists. To enable single sign on in your Sitecore installation, use the Active Directory common authentication layer and the Active Directory Integration module.

Install the following components:

- The Sitecore Active Directory Module.
- The sample integration code.

Both of these components are available to download from the Sitecore Developers Network.

For more information about the Active Directory Module, see the *Active Directory Module Administrators Guide* or *Active Directory Integration in Sitecore Intranet Portal* on the Sitecore Developers Network.

Note
To enable Windows authentication you must add the appropriate settings in IIS Manager. For more information on how to enable Windows authentication, see Configuring Impersonation and Delegation in Windows.

Page Level User Authentication

In page level integration, you work directly with SharePoint lists so must have access to both the SharePoint server and the Sitecore server.

There are several ways that the SharePoint Integration Framework sample controls authenticate users. The SharePoint Integration Object Model `SpContext` class handles authentication in the following ways:

- Prompts a user for login credentials.

- Uses credentials stored in the `sharepoint.config` file.
- Uses credentials from the currently logged in user (for on premise solutions).

If there are no user credentials in the `sharepoint.config` file, then it uses the credentials of the currently logged in user by default. This handled in the API by the `CredentialCache.DefaultNetworkCredentials` class.

Item Level User Authentication

In item level integration, access to SharePoint is required to create or to update integration items. After integration, you work directly with Sitecore items.

The Item Provider uses the `SpContext` class to handle authentication. In the Sharepoint integration wizard enter a SharePoint URL and then choose one of the following methods of authentication:

- Use default credentials stored in the `sharepoint.config` file.

  OR

- Enter alternative login credentials.

Note
In the item Provider, it is not appropriate to use credentials from the currently logged in user.

Configuring Authentication in the SharePoint Config File

Use the `sharepoint.config` file to configure SharePoint user authentication credentials. You can find these settings in the `sharepoint.config` file under the `<servers>` node.

Advantages of using the `sharepoint.config` file:

- Effective if a customer needs to display the same information to all front end users
- Most convenient way to set credentials in the ItemProvider and the wizard
- Possible to store different credentials for different SharePoint sites

Path to the `sharepoint.config` file: `website\app_config\include\`

Each server entity provides credential for a single SharePoint site. For example:

```
<server url="http://<sitename>" username="*****" password="*****" context="Provider"
sharepointOnline="false" />

<server url="http://<sitename>" username="*****" password="*****" context="Any"
sharepointOnline="false" />

<server url="http://<sitename>" username="*****" password="*****" context="Webcontrol"
sharepointOnline="true" />
```

When you edit these settings, use the following parameters:

Parameters	Value
`url`	Provide a URL to a SharePoint site.
`username`	Provide a username to connect to SharePoint. For example, `admin` or `Sharepoint\admin`.
`password`	Provide a password to connect to a SharePoint site.

Parameters	Value
`context`	Defines the context in which these credentials are used. Add one of these values: `Provider` – The credentials supplied only allow access to the Item Provider for item level integration. `Webcontrol` – The credentials supplied can only be used to retrieve information for web controls `Any` – The credentials supplied can be used for both the Item Provider and to render web controls.
`sharepointOnline`	Defines if this is a SharePoint Online server

3.2.2 SharePoint Security and Permissions

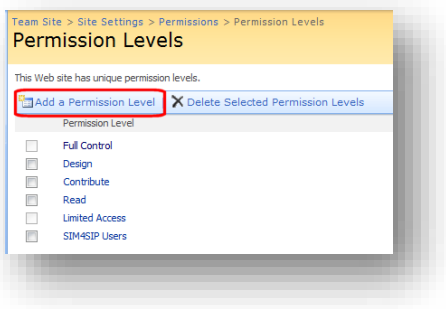
In the SharePoint Integration Framework we use Web services to connect to SharePoint lists. To allow integration of Sitecore and SharePoint content enable the following settings in SharePoint.

Allowing Sitecore Access to SharePoint Webs

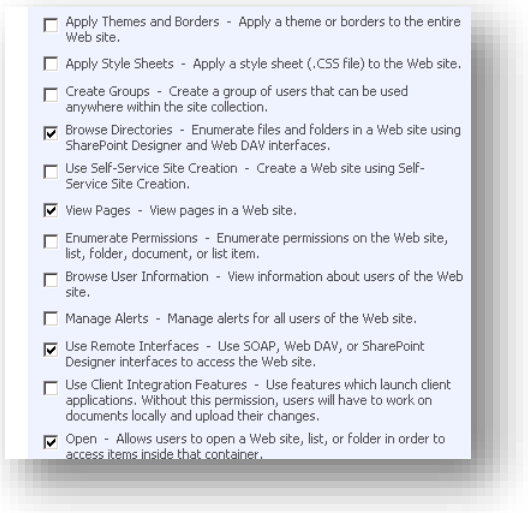
In your SharePoint Foundation website, enable read rights so that Sitecore can access the appropriate webs and sub webs. If you do not have an appropriate SharePoint site to use, then create a new web or sub web and add a new permission level.

To add permissions to a Web in SharePoint Foundation:

1. In the top level of your SharePoint website, select **Site Actions**, **Site Settings**, **Advanced Permissions**, **Settings**, **and Permission Levels**.
2. In **Permission Levels**, click **Add a Permission Level**.



3. Enter a suitable name for your permission level, for example *My Permissions* and select the following permissions.



4. Go back to **Permissions** and add a new **Group**. Give the group a name, such as *My Group.*
5. Select the newly created permissions level for this group**.** In this example, *My Permissions.*
6. Add users to the new *My Group*. For example, *NT AUTHORITY\authenticated users*. You can find these users automatically using Active Directory.

Configuration of Sitecore and SharePoint security is now complete.

3.2.3 Configuring Impersonation and Delegation in Windows

Note
This section describes impersonation and delegation in Windows for on-premises installations only.

To configure page level integration to display SharePoint list data for the currently logged in user in IIS, you must enable ASP.NET Impersonation and Windows Authentication.

There are two possible scenarios:

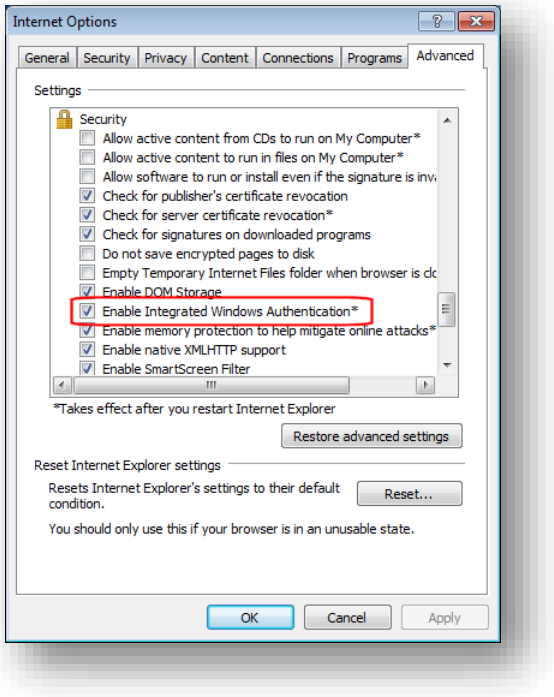
- The SharePoint server is on the same computer as your Sitecore website - enable impersonation.
- The SharePoint server and Sitecore Server are on different network computers - enable both impersonation and delegation.

Configuring Internet Explorer

How to configure Internet Explorer on client computers:

1. Use Internet Explorer to access your Sitecore CMS. Always enter the computer name in the browser and not the IP address. For example, *mycomputer.dk.sitecore.net*.

2. In the **Internet Options** dialog box, **Advanced** tab, select the **Security** option *Enable Integrated Windows Authentication*.



3. In Internet Explorer, **Internet Options**, add your Sitecore website to the *Local Intranet* or *Trusted sites* group.

   If you implement these steps correctly, Kerberos will authenticate both client and server.

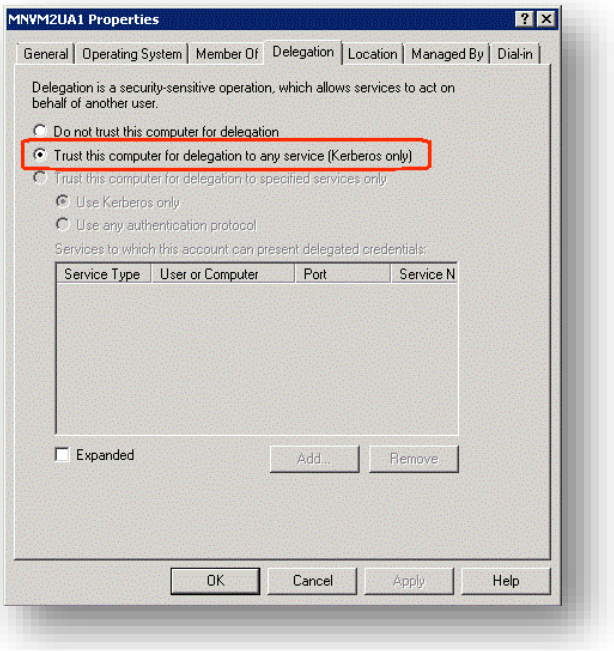
Configuring Active Directory

Your Sitecore CMS must be trusted for delegation.

To enable delegation in Active Directory:

1. Open **Active Directory Users and Computers**.
2. Navigate to your Sitecore server and open the **Properties** dialog box.

3. In the **Delegation** tab, select the *Trust this computer for delegation to any service (Kerberos only)* option.

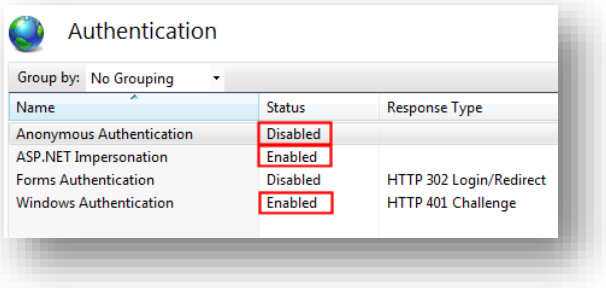


4. To save your changes and close this window click **OK**.

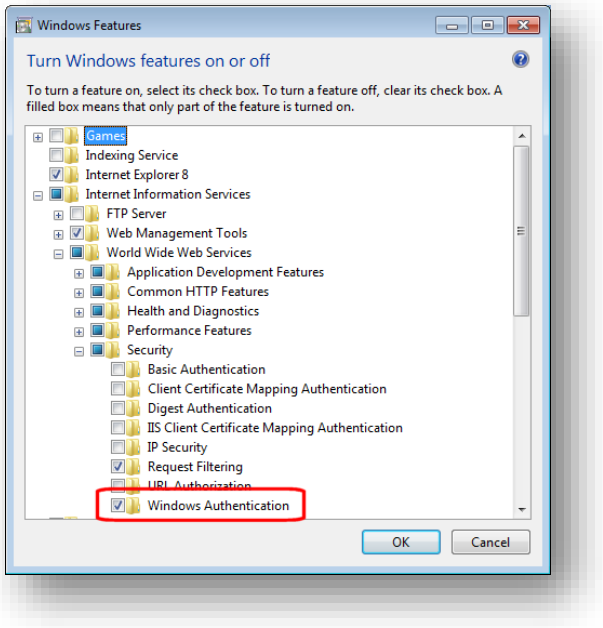
Configuring IIS

How to configure IIS for your SharePoint and Sitecore servers:

1. Open IIS Manager.
2. Select the website you want to configure.
3. Double click **Authentication**.
4. Set the following authentication settings:
   - Anonymous Authentication - *Disabled*
   - Windows Authentication - *Enabled*
   - ASP.NET Impersonation - *Enabled*



If Windows Authentication is not visible in the list of available options, add it by selecting the check box in the **Windows Features** dialog box.



Note
Enabling ASP.NET Impersonation may affect server performance.

Ensure that the Sitecore CMS supports Kerberos. This step is only necessary if the previous steps fail:

1. Click **Start**, click **Run**, type **cmd**, and then press **ENTER**.
2. Locate the directory that contains the *Adsutil.vbs* file. By default, this directory is *C:\Inetpub\Adminscripts*.
3. Use the following command to retrieve the current values for the **NTAuthenticationProviders** metabase property:

   cscript adsutil.vbs get w3svc/[*WebSiteNumber]*/root/NTAuthenticationProviders
4. The previous command must return:

   NTAuthenticationProviders : (STRING) "Negotiate,NTLM"

   Otherwise go to http://support.microsoft.com/kb/215383/en-us to solve the problem.

Additional SharePoint Server Configuration

1. Configure the SharePoint server to handle requests to the same URL as the server fully qualified domain name (FQDN). To do this you need to register the SharePoint server service principal name (SPN) in the Active Directory service

   To register the SharePoint SPN in the Active Directory service:

   a. First check that there is an appropriate mapping registered in SharePoint.

   b. In Central Administration, click *System settings* and then configure alternate access mappings.

   c. In IIS check that the SharePoint site has a binding to port 80 for all hosts.

2. Use the Network service account to run the SharePoint Site Pool. You can use other user accounts but whichever account you use, it must have access to Active Directory.

   To add a Network Service account:

   a. In Central Administration, click *Security*, then click *Configure Service Accounts*

   b. Select *Web Application Pool* for your site.

   c. Select *Network Service* as an account for this component.

3. Make sure that you have configured the default authentication provider correctly.

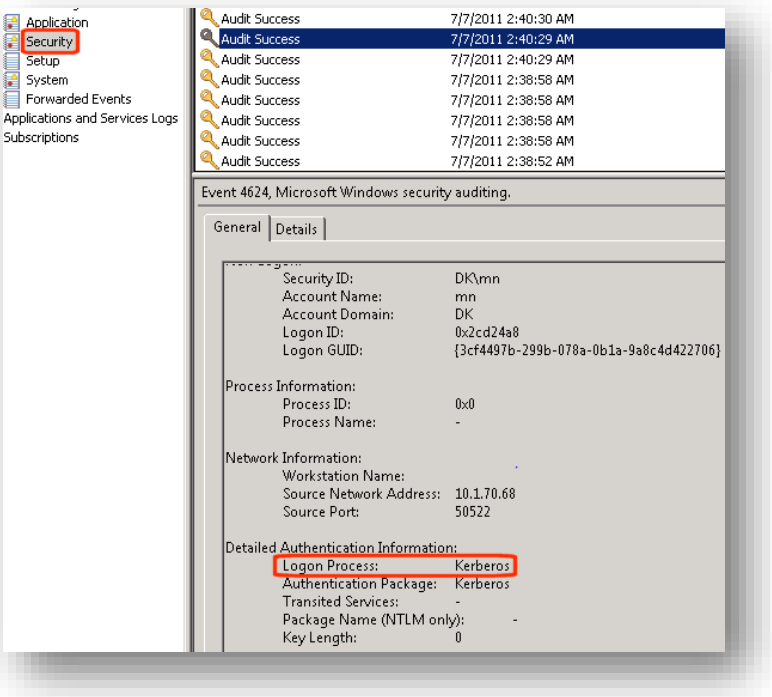
   To configure default authentication:

   a. In Central Administration, *Security*, click *Authentication Providers* and then click *Default*.

   b. Configure the following settings:

   - Authentication type = *Windows*
   - Anonymous access = *disabled*
   - IIS Authentication settings = *Integrated Windows Authentication (Negotiate Kerberos)*

When you have correctly configured the SharePoint server you can see the following entries in the Windows Security Log:



If you have configured the SharePoint server correctly the Logon Process = *Kerberos*.

Kerberos Authentication

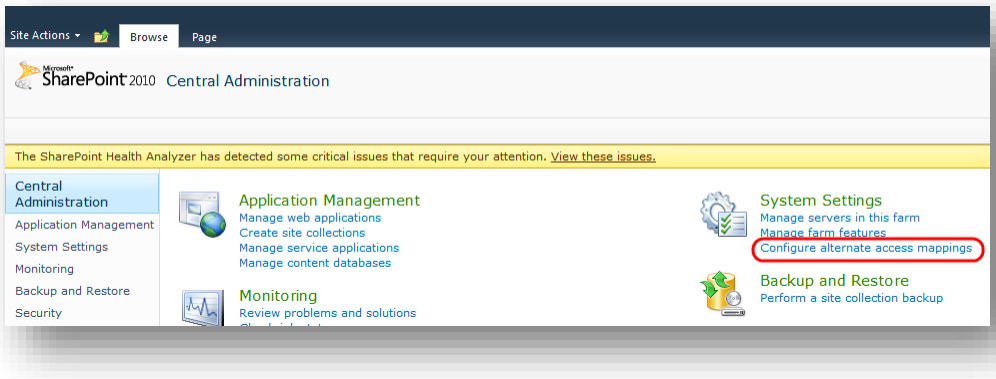
For the Kerberos authentication to work, you must implement one of the two following options:

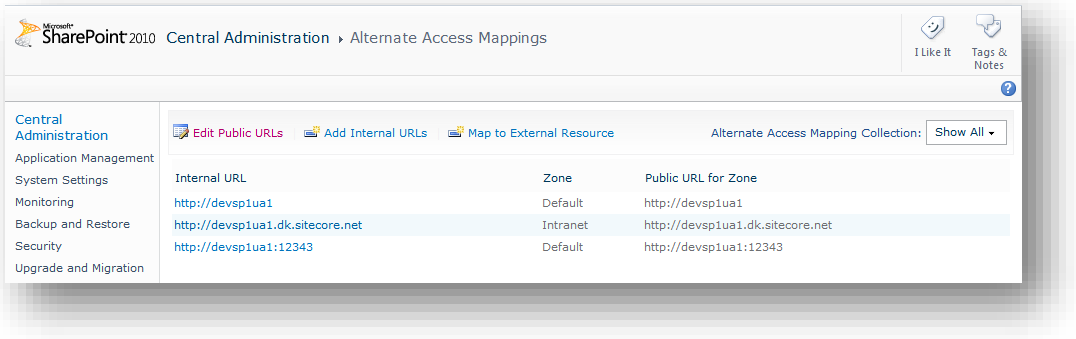
1. Configure the SharePoint server so that other applications access it using exactly the same URL as the server fully qualified domain name (FQDN).

2. If the previous option is not suitable for you, you must register the server service principal name (SPN) in the Active Directory service.

To use the first option:

1. Check, that there is an appropriate mapping registered in SharePoint.
2. On the SharePoint server, select **Central Administration**, **System settings**, **Configure alternate access mappings**.

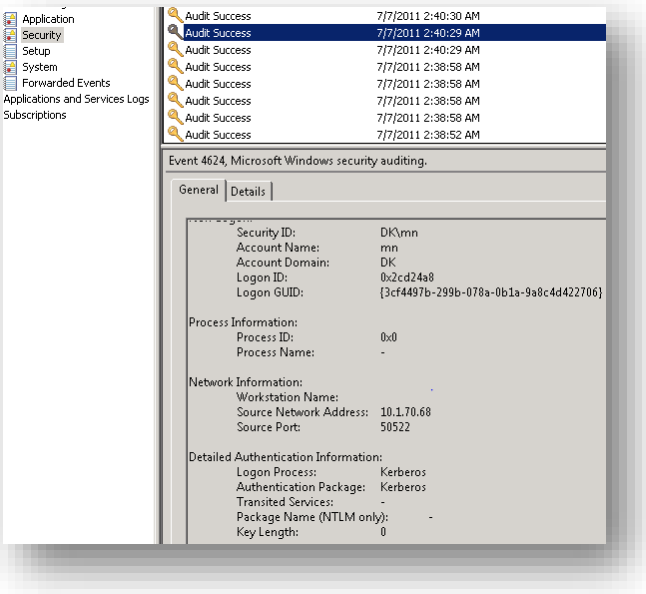




3. Check that SharePoint site in IIS has binding to 80-th port for all hosts.
   - The NETWORK SERVICE account should be used to run SharePoint Site Pool:
     a. Click **Central Administration**, **Security**, **Configure Service Accounts**.
     b. Select *Web Application Pool* for your site.
     c. Select NETWORK SERVICE as an account for this component.
   - Make sure that the default authentication provider is configured correctly:
     a. Click **Central Administration**, **Security**, **Authentication Providers**, **Default**.
     b. Set the following settings:

Setting	Value
Authentication type	Windows
Anonymous access	disabled
IIS Authentication settings	Integrated Windows Authentication (Negotiate Kerberos)

If the SharePoint server is set up correctly, you should see the following entries in the Windows Security log:



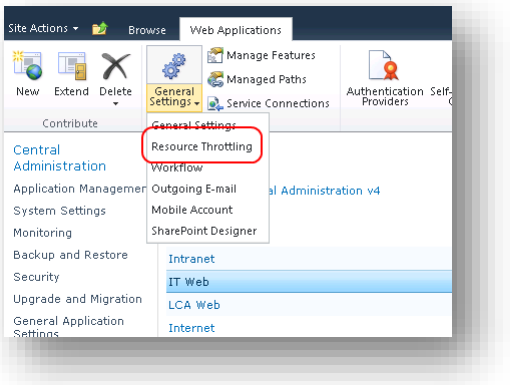
Important
Make sure that Logon Process is Kerberos.

List View Threshold

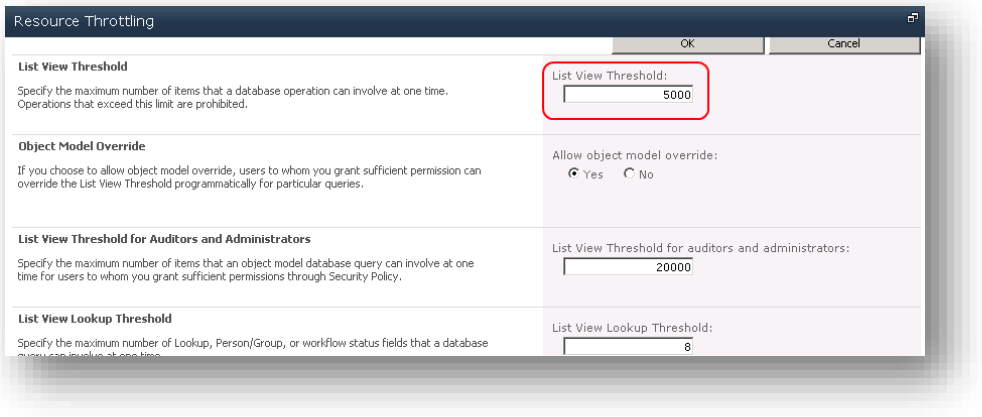
List View Threshold is an option in SharePoint Server 2010 and SharePoint Server 2013 that enables you to limit how users can load the server when working with large lists. It allows you to set a limit on how many rows of data can be retrieved for a list or library at any one time.

To change the *List View Threshold* and other resource throttling settings:

1. Navigate to **Central Administration**.
2. Click **Application Management**, **Manage Web Applications**.
3. Select the web application for which you want to change the LVT.



4. In the ribbon, click **General Settings** and then select **Resource Throttling**.



5. In the **List View Threshold** field, enter another value and click **OK**.

Important
Increasing List View Threshold by a large amount will increase the use of hardware resources on the server.

Note
The **List View Lookup Threshold** field limits the number of joins that a query can perform — the number of **Lookup**, **Person/Group**, or **Workflow Status** fields that are included in the query.

Chapter 4

Page Level Integration

This section provides an overview of page level integration. Page level integration enables you to embed SharePoint lists on a Sitecore website using SharePoint Integration Framework sample controls. Sitecore provides several example sample controls with the SharePoint Integration Framework. This section provides instructions on how to configure and use these controls.

This chapter contains the following sections:

- Components
- Using Sample Controls to Display SharePoint Lists
- Using SharePoint Search
- Configuring Sample Control Properties

4.1 Components

When you install the SharePoint Integration Framework, it adds some additional presentation components to your site:

- SharePoint Templates
- SharePoint Web controls

4.1.1 SharePoint Web Template

To locate the SharePoint Web template in the Sitecore content tree, navigate to:

`/sitecore/templates/Sharepoint/Page Level Integration/Sharepoint Web`

Template	Description
SharepointWeb	Standard template to use with SharePoint Integration sample controls. This template allows you to enter information about your source SharePoint site.

Note
You can also use other templates when you create Sitecore items to integrate with SharePoint. The advantage of using the SharePoint Web template is that you can enter information about the source SharePoint site such as Server and Web without having to enter it on the properties of each sample control.

4.1.2 SharePoint Sample Controls

Use the sample controls to display SharePoint lists, such as document libraries, announcements, and tasks in real time on a Sitecore website. All sample controls are the sublayout .ascx files. Some of the controls also contain C# code behind files.

In the Content Editor, Presentation tab, use Layout Details to add controls to the presentation layer and to configure properties for each control.

Sample control categories:

- **Multi List** — Generic grid control
  Use the *SharepointList.ascx* control to display any SharePoint list. This control is very versatile but quite complex and requires more advanced developer skills to customize.
- **Single List** — Basic sample control
  Use this type of control to point to a specific SharePoint list. This category is easy to implement and customize but more limited. For example, *AnnouncementsList.ascx*.

To locate these controls in the Sitecore content tree, navigate to:

`/sitecore/layout/Sublayouts/Sharepoint`

The following table describes the controls in the SharePoint sublayouts folder.

Sublayout Name	Description	Control Type
Sharepoint Announcements	Points to a single SharePoint announcements list. Can be placed anywhere on a site apart from the front page. This control can only display unexpired items.	Single List
Sharepoint List	Most versatile SharePoint control. Closely replicates SharePoint functionality, enabling you to display any kind of list, sort lists and display views. Customization requires more advanced developer skills.	Multi List
Sharepoint Search	Searches SharePoint using the SharePoint Search Web service.	Single List
Sharepoint Tasks	Points to a single SharePoint task list.	Single List

4.2 Using Sample Controls to Display SharePoint Lists

Use the sample controls provided in this module to quickly and easily connect to SharePoint and display any list on your Sitecore website.

Choose a suitable control depending on your business objectives and then in the Content Editor or Page Editor add the control to the presentation layer of an item.

Using the sample controls, there are three types of Sitecore item that you can integrate with SharePoint:

- Existing items — this is the quickest and easiest approach. You can add a SharePoint control to any Sitecore item regardless of template or layout. You must specify a SharePoint server and web on the control.
- New items — create a new content item based on the SharePoint Web template. Then add any SharePoint control to the item using Layout Details. Some sample controls also allow you to specify a SharePoint server and web.

The following sections outline each of these methods in more detail.

Note
The SharePoint Integration wizard offers flexible ways of displaying SharePoint data in Sitecore. For more information, see Chapter 5: Item Level Integration.

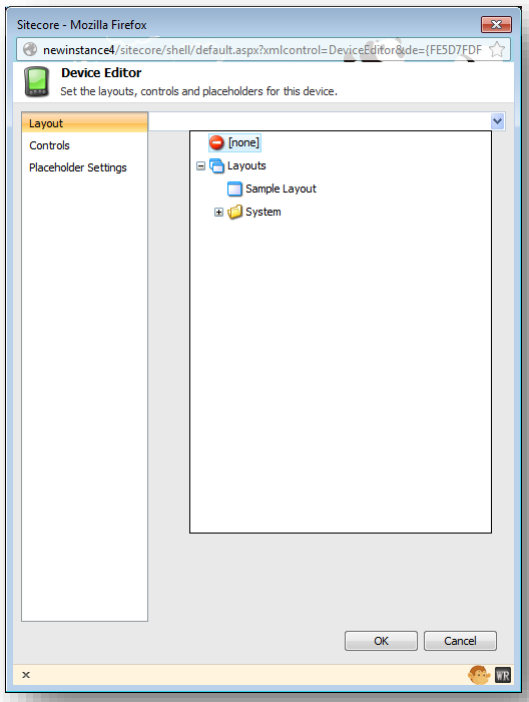
4.2.1 Displaying SharePoint Lists on an Existing Item

Choose a SharePoint Integration Framework sample control to add to an existing Sitecore content item. The SharePoint List control is the most flexible of the sample controls and most closely replicates SharePoint functionality.

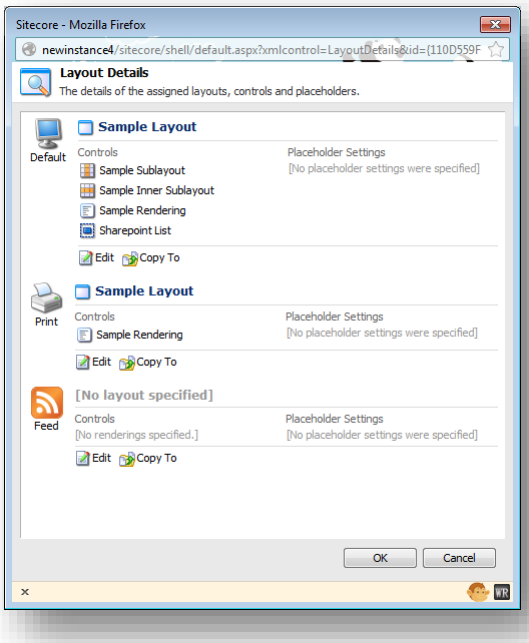
To add a SharePoint List control to a content item in Sitecore:

1. Open the Content Editor.
2. In the content tree, select a suitable item.
3. On the Sitecore ribbon, click **Presentation**, and in the **Layout** group click **Details**.
4. In the **Layout Details** dialog box, select the **Default** layout and click **Edit**.

5. In the **Device Editor**, select **Layout**, and then click *Sample Layout.*

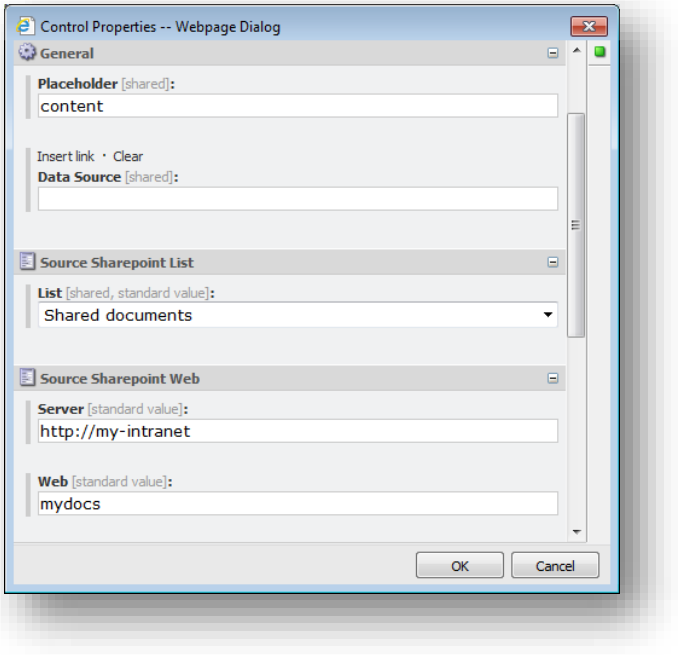


6. In the **Device Editor**, select **Controls** and then add a *SharePoint List* sublayout:
   `/sitecore/layout/Sublayouts/Sharepoint/Sharepoint List`



7. Set the following properties on the control:

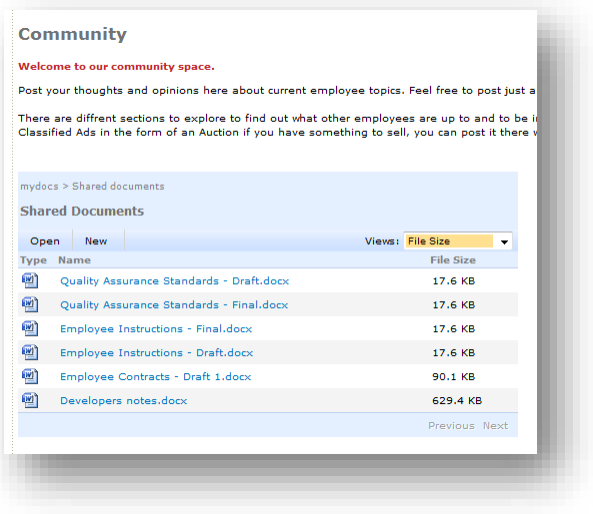
Property	Value
List	*<Name of SharePoint List>*, for example *Shared documents*
Server	*<Name of SharePoint Server>,* for example *http://my-intranet*
Web	*<Name of SharePoint Web>*, for example *mydocs*



Note
If the List field drop-down does not contain any options, then you can create new list definition items. For more information on creating list values, see Creating a List Value.

8. Save your changes.

9. Preview the item in Sitecore or in a browser. You can see the SharePoint document library embedded directly in a Sitecore website page.



4.2.2 Displaying SharePoint Lists on a New Item

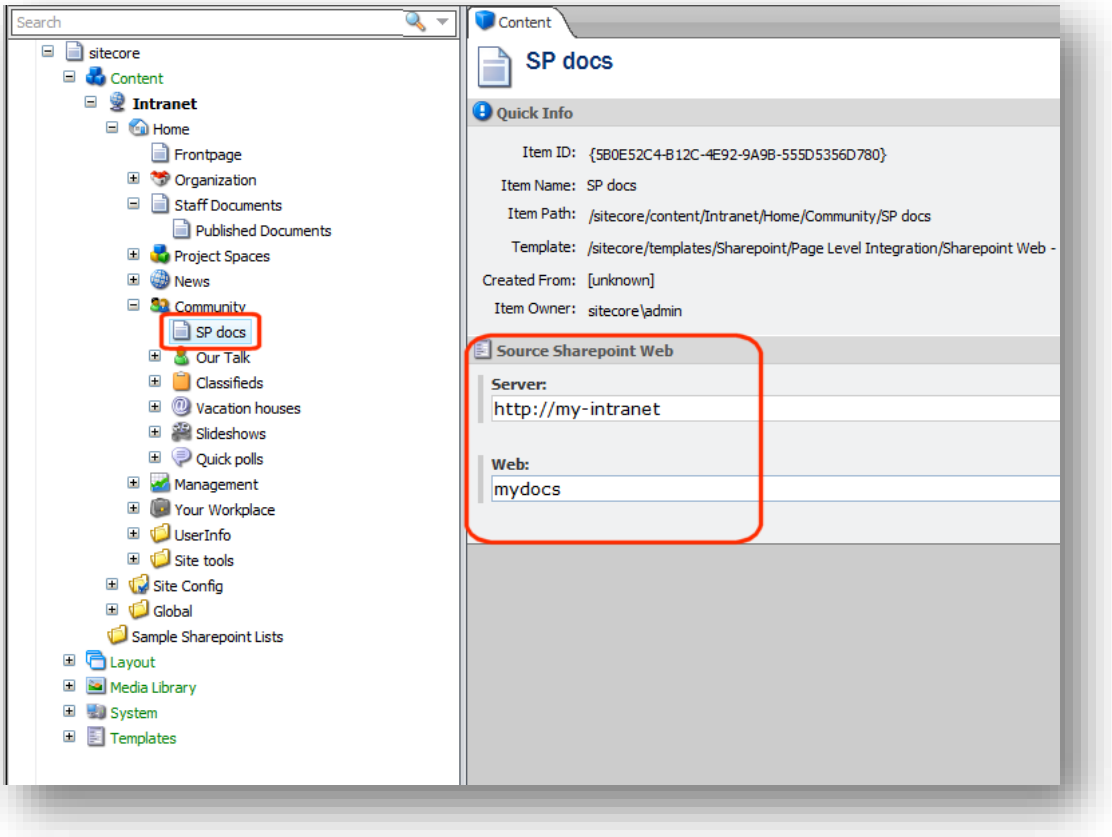
If you create a new content item based on the SharePoint Web template, you can use any of the sample controls to display SharePoint lists. In this example, create a new item under any item on your site.

To add a SharePoint List control to a new Sitecore content item:

1. In the Content Editor, select a suitable node in the content tree. Click **Insert from Template**.
2. Select the *SharepointWeb* template:

   `/sitecore/templates/Sharepoint/Page Level Integration/Sharepoint Web`
3. Give the item a suitable name, such as *SP docs*.
4. Enter the following values:

Field	Value
Server	*<Name of SharePoint Server>*, for example *http://my-intranet*
Web	*<Name of SharePoint Web>*, for example *mydocs*



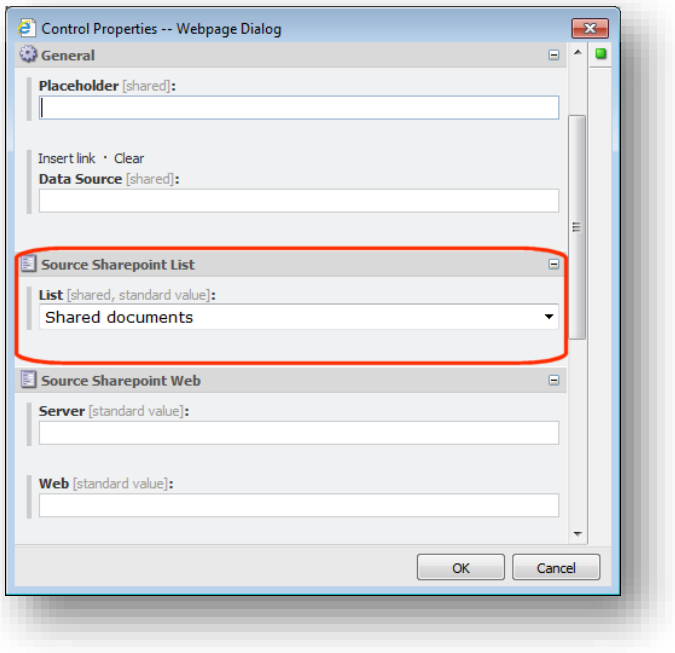
5. Save your changes.

Next, add a SharePoint List control and configure the presentation:

1. Select the new content item that you created.
2. In the Device Editor, click **Layout** and select the *Sample Layout*.
3. In Controls, add the *SharePoint List* control.

   `/sitecore/layout/Sublayouts/Sharepoint/Sharepoint List`
4. Select the *Sharepoint List* control and click **Edit**.

5. In the Control Properties dialog box, in the List field, select *Shared Documents*.



6. Preview your new content item in Sitecore or open a new browser window.

Note
It is not always necessary to complete the Server and Website fields in the control. You can also specify these paths on the content item. However, you must always complete the List field in the SharePoint List control.

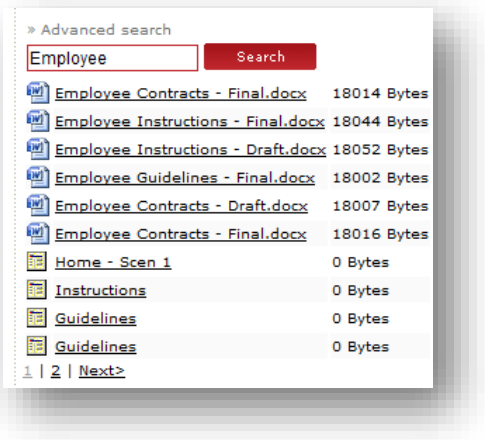
4.3 Using SharePoint Search

The SharePoint Integration Search control enables you to search SharePoint lists or for items in lists from a Sitecore website. The SharePoint Integration Framework sample search control is an example of how you can implement SharePoint search. Use this control as a quick and easy way to implement SharePoint search functionality on your Sitecore website.

You can add the sample search control to the presentation layer of any content item created with the SharePoint Web template. Add this control to the presentation layer in the same way as any of the other sample controls.

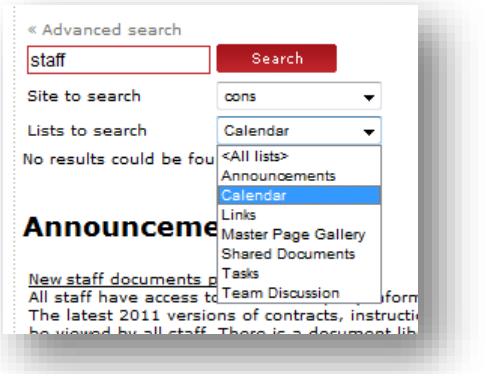
4.3.1 Standard Search

Standard search is the default view of the SharePoint Search control. You can enter a search term that is part of a list title and return a list SharePoint documents or list items.



4.3.2 Advanced Search

Advanced search enables you to make the focus of your search more specific. You can use two drop-down list controls to select a Sharepoint web, sub web or list.



4.3.3 SharePoint Search API Classes

You can find the SharePoint Search classes in the following location in the object model:

`Sitecore.Sharepoint.ObjectModel\Search`

There are three main search classes:

- Query
- SearchResult
- SearchResultCollection

4.3.4 Understanding the Search Control

Add a SharePoint Search control to any content item based on a Sharepoint Web template. The search control is a sublayout called `SharepointSearchControl.ascx`. Use the following path to locate it in the content tree:

`/sitecore/layout/Sublayouts/Sharepoint/Sharepoint Search`

An explanation of how the SharePoint Search sample control works.

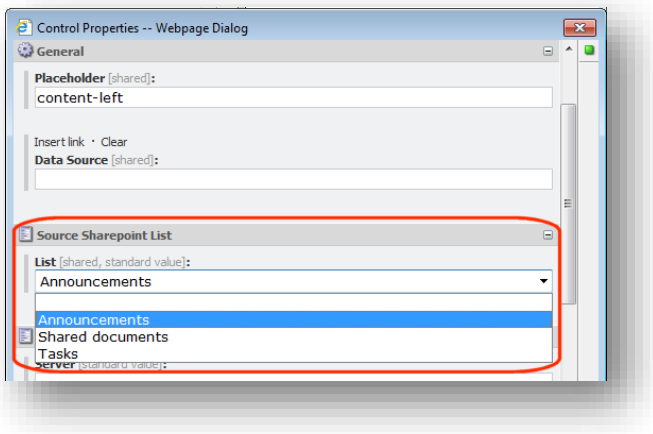
- When you enter a text string in the search box, the `Query` class encapsulates the parameters of the search request. For example, search text and any values selected in the advanced search drop-down controls.
- The `Query` class encapsulates the `ToSearchString` method that converts the current search parameters to a format that the SharePoint Search Web service understands.
- The `Server` object encapsulates the `Search` method and has a parameter called Query type. This represents the search parameters and it returns results of the type `SearchResultCollection`. This method converts the query and sends it to SharePoint as XML using the SharePoint Search Web service.
- SharePoint applies its own search technology to the query and searches its SQL database.
- When SharePoint finds some results, it sends them back to Sitecore as XML.
- The `SearchResult` class represents one record of search results found. The `SearchResultCollection` class represents all records found as search results.
- These classes parse the XML that SharePoint returns as search results and saves them as key value pairs.
- The `SharepointSearchControl.ascx` sublayout formats the search results and embeds them in the Search control on your web site.

4.4 Configuring Sample Control Properties

In the Content Editor, you can configure more SharePoint Integration control property options. For example, you can customize the drop-down options that are available in the control properties dialog box.

The SharePoint List control can display any list. However, the available lists that Sitecore displays must first be configured using list values.

You can create new list values if you want to display a SharePoint list with a different name. In the following example, *MyList* does not appear in the drop-down list.



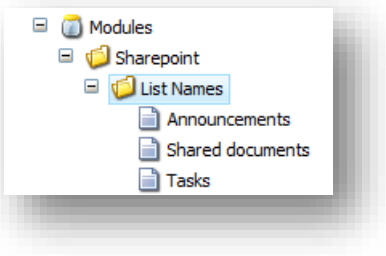
To add *MyList* to the drop-down options, create a new list item. If you also use the generic SharePoint List sublayout control you can point to any type of list in this way.

Note
If you create your own custom controls, you must also configure custom list items.

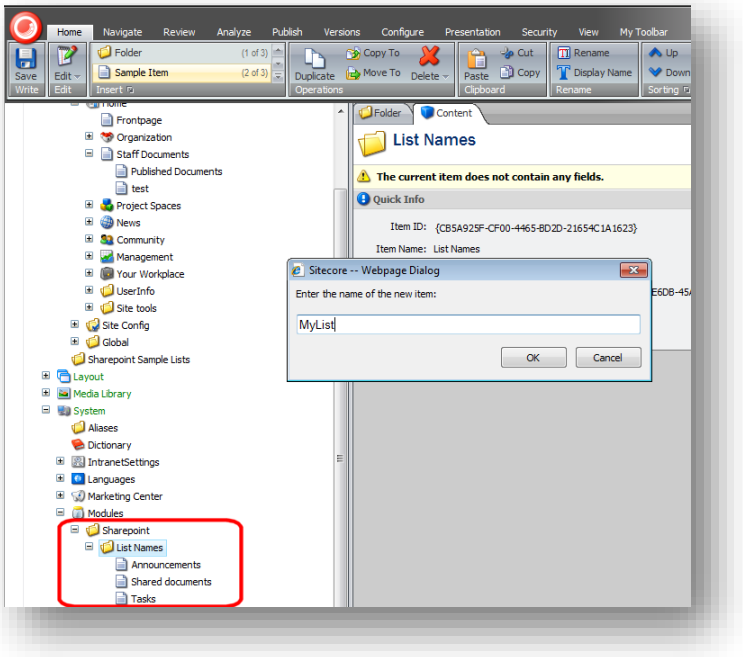
4.4.1 Creating a List Value

To create a new control list value:

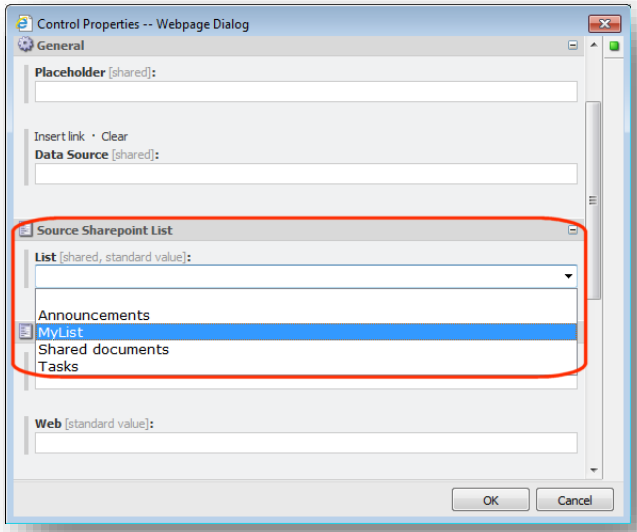
1. In the Content Editor, navigate to the **List Names** definition item.
   `/sitecore/system/Modules/Sharepoint/List Names`



2. Create a new item based on the *Sample* template. Enter a name for your list item in the **Name** field. The name you choose must match the actual name of the Sharepoint list.



3. In the content tree, select a content item and add a **SharePoint List** control.
4. In **Layout Details**, open the **Control Properties** dialog box for this control. Enter a name in the **Website** field and a path in the **Server** field for the SharePoint site you want to display the control.



Notice that *MyList* now appears in the drop-down list of options.

Chapter 5

Item Level Integration

Item level integration offers you more full featured SharePoint integration options. This section provides instructions on how to configure and map integration items using the SharePoint Integration wizard or by directly editing the XML in the SharePoint integration definition item.

This chapter includes:

- Overview
- Components
- Options and Settings
- Using the SharePoint Integration Wizard
- Editing the XML in a SharePoint Integration

5.1 Overview

Item level integration uses a wizard and the Item Provider class to integrate SharePoint lists with Sitecore in real time as common content items or as Media Library items. The SharePoint Integration wizard creates a bidirectional relationship between SharePoint and Sitecore. This means changes to SharePoint lists appear in Sitecore and changes to Sitecore integration items appear straight away in SharePoint. You can configure settings in the wizard to enable or disable this functionality.

The SharePoint Integration wizard enables you to create mappings between SharePoint lists and Sitecore items and then saves all configuration settings as XML to a field in the SharePoint integration definition item. The Item Provider class uses the XML configuration information to integrate SharePoint list data with Sitecore.

Item level integration enables you to work with SharePoint lists completely in Sitecore, a method suitable for publishing SharePoint content to a corporate extranet. In the Content Editor, you can view SharePoint lists in real-time and specify how often integrated content is updated.

Benefits of Item Level Integration:

- Integrate SharePoint and Sitecore content in real time.
- Store SharePoint lists in the content tree or the Media Library.
- The SharePoint Integration wizard simplifies configuration and field mappings.
- Set an expiration interval to update lists and optimize performance.
- Apply Sitecore functionality to SharePoint lists, such as publishing and workflow.
- Use scheduled BLOB transfer to import BLOBs data from SharePoint.

5.2 Components

This section includes an explanation of each of the components, options and settings available in item level integration.

5.2.1 SharePoint Integration Definition Item

To integrate SharePoint lists with the Sitecore content tree or the Media Library use the SharePoint Integration wizard to create a SharePoint integration definition item based on the *Sharepoint Integration Configuration* template. You can find the *Sharepoint Integration Configuration* template at the following location:

```
/sitecore/templates/Sharepoint/Item Level Integration/Sharepoint Integration Configuration
```

SharePoint integration definition items contain the following fields:

Field Name	Description
`IsIntegrationItem`	This field contains a check box. It is selected if the Sitecore item contains items integrated with SharePoint.
`BidirectionalLink`	Select this check box if you want updates to come from both SharePoint and Sitecore.
`IntegrationConfigData`	This field contains the XML field mappings and other configuration information needed to retrieve list data from SharePoint. Use the SharePoint Integration wizard to configure these settings. You must first enable raw values to display this data.
`ScheduledBLOBTransfer`	Select the check box in this field if you want to run a scheduled task to import SharePoint lists at a pre-defined time. Use the Tasks node in the Content Editor, content tree to configure a scheduled task.

5.2.2 The SharePoint Integration Wizard

Use the SharePoint Integration wizard to configure the following settings:

- Security
- Server
- Web
- List
- View
- Expiration Interval
- Scheduled BLOB Transfer
- Item limit
- Is SharePoint Online
- Field Mappings. For example, title field, modified field, body field.

Once you have completed the wizard, the SharePoint integration definition item contains your settings. In the wizard, you also have the option to save your settings to a new template.

5.3 Options and Settings

This section contains more detailed information about how to configure item level integration using the SharePoint Integration wizard.

5.3.1 Authentication

The item provider can handle SharePoint login credentials in two ways:

- Users can enter credentials in the fields provided in the wizard.
- You can specify user credentials in the `sitecore.config` file

If you enter invalid credentials, you get an error message on the progress page of the wizard.

Note
SharePoint Online check box should be checked in case of connection to SharePoint Online.

5.3.2 Creating SharePoint Integration Mappings

Use the wizard to create field mappings between SharePoint list items and Sitecore content items. The wizard saves mappings and other configuration settings as XML in the *IntegrationConfigData* field.

Example of some typical field mapping tags:

```
        <FieldMapping>
              <Source>ows_Body</Source>
              <Target>Body</Target>
            </FieldMapping>
```

The `Source` tag refers to the SharePoint field that you want to map.

```
<Source>ows_Body</Source>
```

The Target tag refers to the Sitecore field that you want to map the SharePoint field to.

```
<Target>Body</Target>
```

5.3.3 Performance Tuning

This section describes settings that enable you to fine tune the Item Provider for better performance.

Scheduled BLOB Transfer

If you want to import a very large list item, such as an image or video file stored as a BLOB in a SharePoint document library, then you can use a BLOB transfer schedule to download the file at a pre-defined time as a scheduled task.

Configure the task in Sitecore and schedule it to run at a quiet time when it is less likely to have a negative impact on performance. You can create a Sitecore package that contains pre-defined commands and scheduled items.

In the Content Editor, content tree use *Tasks* to create the scheduled task. There are two settings to configure:

- Commands
- Schedules

To activate a scheduled BLOB transfer, select the *Scheduled BLOB Transfer* check box in the SharePoint Integration wizard.

Note
You can only use functionality for updating BLOB document list items.

Expiration Interval

The expiration interval is the minimum amount of time between requests to the SharePoint server for updated list information. You define the expiration interval in seconds. For example, an expiration interval of 3600 seconds requests updates from the SharePoint server once an hour.

During the time between expiration intervals there are no requests to the SharePoint server to update the children of a root item unless you change an item and then save your changes.

For example, if you make a change to a Sitecore integration item this triggers an update to the corresponding SharePoint list.

This means that the connection to the server remains constant with only minimal interruptions to process updates. You can adjust the expiration interval to optimize performance.

Set the expiration interval using the SharePoint Integration wizard or by editing the XML in the *IntegrationConfigData* field.

5.3.4 Updating SharePoint Lists from Sitecore

After you have used the SharePoint Integration wizard to import SharePoint lists, you can update, edit and delete list items from Sitecore or SharePoint. The Item Provider creates a real time, bidirectional relationship.

For example, if you integrate an announcements list, you can edit the title and body text of the announcement in Sitecore and see the changes immediately in SharePoint. This is because when there is a `GetItem` call for a specific Sitecore item the update is immediate.

Note
If you set the expiration interval too low, you may create a condition where SharePoint updates the item before you can save your changes. This makes it impossible to see recent changes reflected in the Sitecore item.

Important
All items contain some fields that it is not possible to update, such as *AssignedTo*, *Created* and *LinkTitle*.

In your integration item, if you want to create a new SharePoint item or list from Sitecore without recreating the integration item, you can take two approaches:

Item Level Integration - If you have used the SharePoint Integration wizard to create integration items you can create new items from Sitecore if the items are documents or items in a document library. In the Content Editor go to the SharePoint integration definition item you created using the wizard (this item must already map to a SharePoint document library). Insert an item from a template, which contains a BLOB field. You can also add items to a SharePoint document library using the Sitecore Upload Files (Advanced) button.

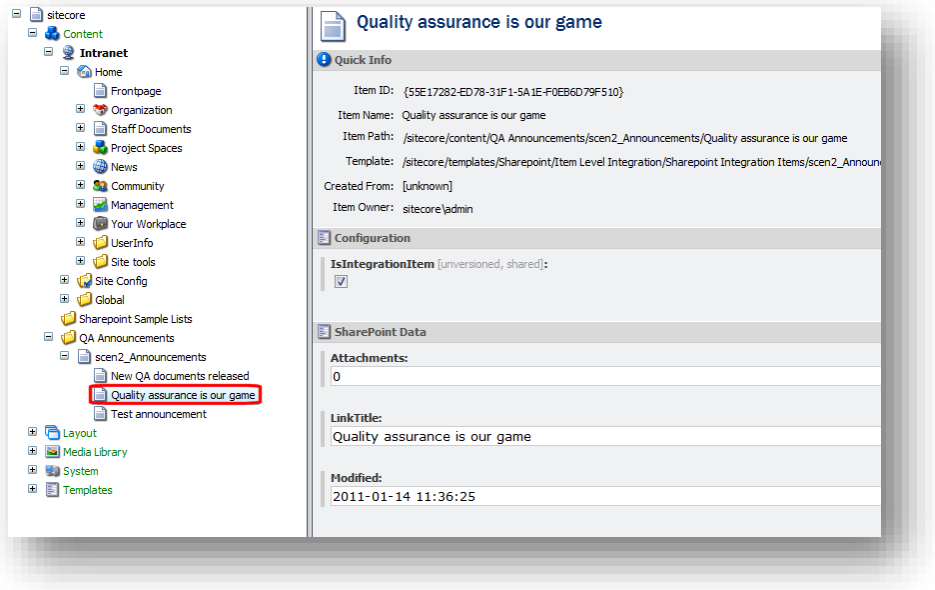
Page Level Integration – The SharePoint List and Document List sample controls have built in actions, such as *Open* and *New* that you can use to create a new item from Sitecore. If you click *New* this opens a SharePoint page where you can create the item. When you click *OK*, the item appears in both SharePoint and Sitecore.

Note
Although you can create any type of list item from SharePoint, it is not always possible to do the same from Sitecore. However, it may be possible to use the SharePoint Integration Framework API to extend this standard functionality.

5.3.5 Presentation Options

The Item Provider represents SharePoint list items of a specified list as content items or media items in the Sitecore content tree.

A SharePoint announcement list item represented in the Sitecore content tree:



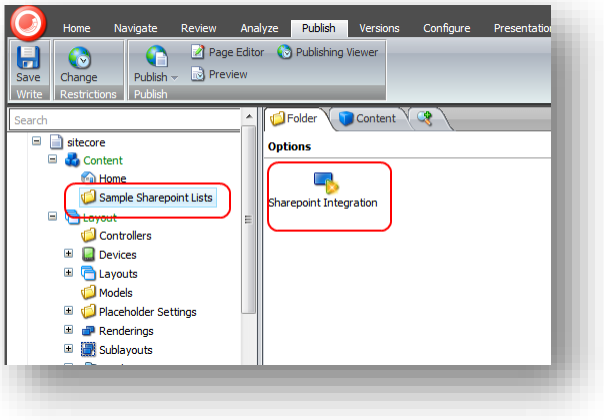
This gives you more flexibility and a wide variety of presentation options. You can use any existing rendering or sub layout, if it is suitable or you can create your own custom controls.

You can also use other Sitecore functionality, such as publishing, workflow and versioning to manage your integrated content.

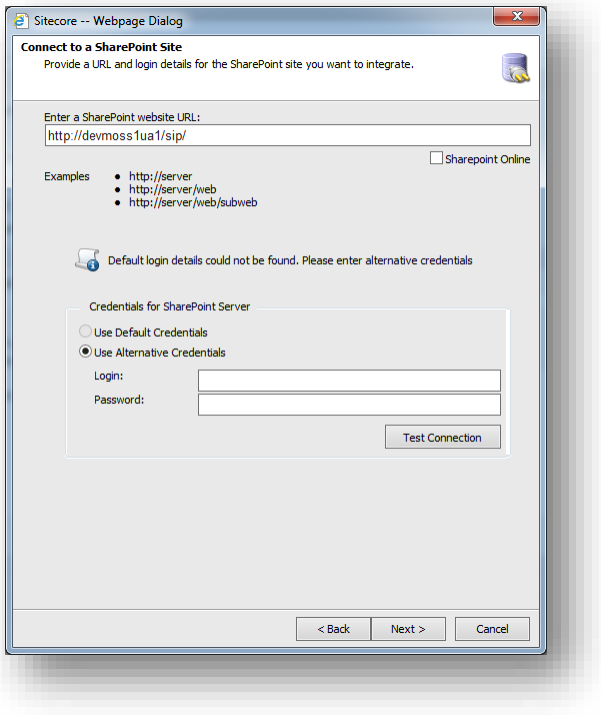
5.4 Using the SharePoint Integration Wizard

To integrate a SharePoint list with Sitecore using the SharePoint Integration wizard:

1. In Sitecore, open the Content Editor.
2. In the content tree, select the *SharepointSampleLists* folder or create a folder of your own to store your integration items. Integration items have the flexibility to be stored anywhere in the content tree.



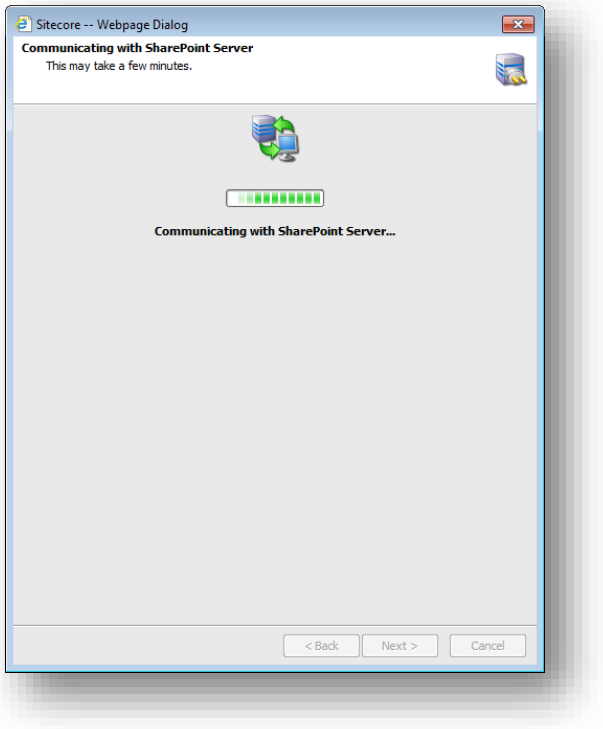
3. In the **Options** tab, click **Sharepoint Integration** and the **Create a SharePoint Integration Item** wizard.
4. In the **Connect to a SharePoint Site** dialog box, configure the connection details.



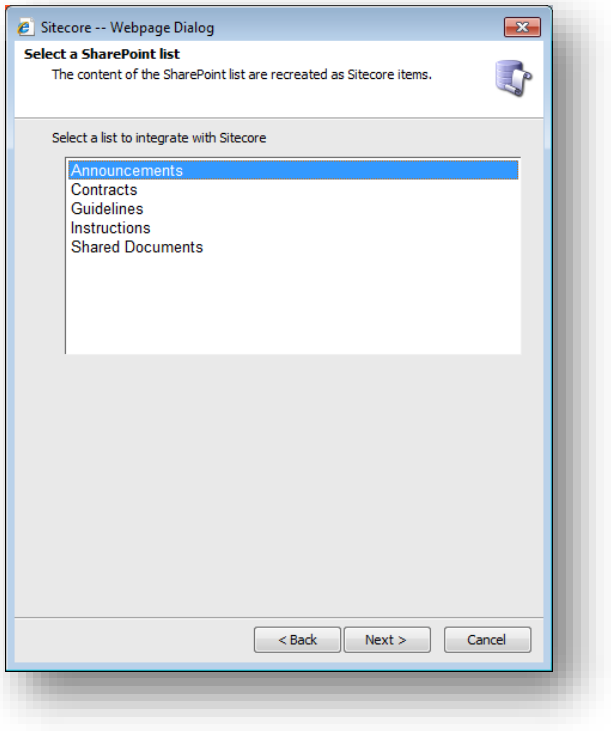
- Enter the address of a SharePoint site or server in the field provided.
  Enter a URL in the following format:

```
http://server
http://server/web
http://server/web/subweb
```

- Check the SharePoint Online checkbox if the target SharePoint server is SharePoint Online.
- The wizard searches in the `sharepoint.config` file for the URL and user credentials. If credentials can be found then move to the next page of the wizard. If the Wizard cannot find the URL or default user credentials then you see a message requesting alternative credentials.
- Click **Test** to test the connection to SharePoint.

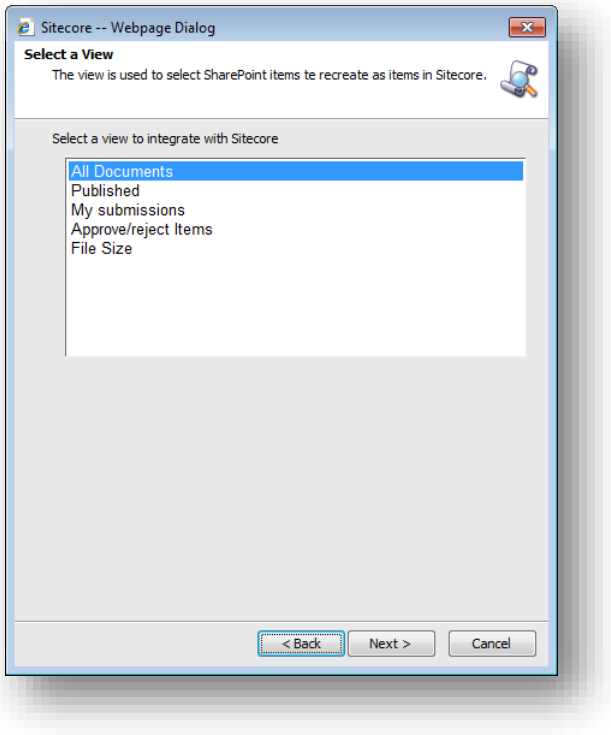


5. The **Select a SharePoint List** page displays all the lists available on the specified website. Select the type of list that you want to import. For example, *Announcements*.



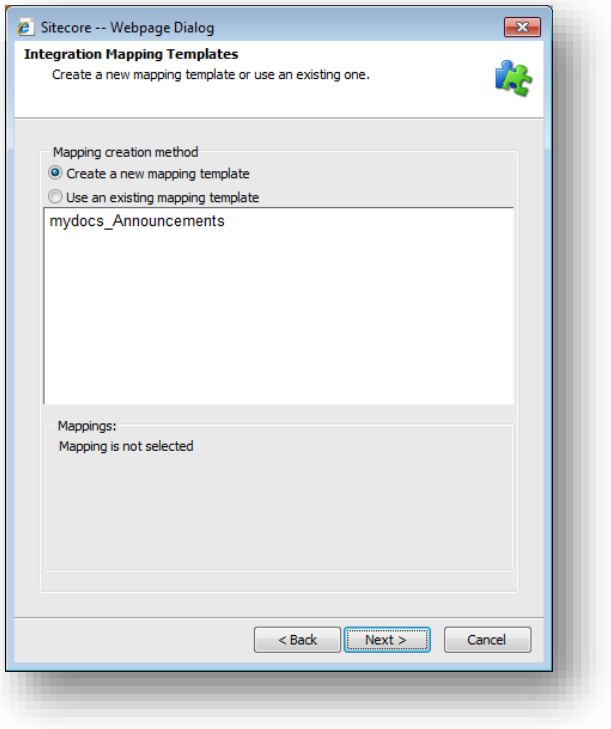
6. In the **Select a View** page select the SharePoint view that you want to integrate. This reduces the number of fields you need to map between SharePoint and Sitecore and defines which list items

to display.



Selecting a view enables you to display list items that contain columns with clauses, such as *where by* or conditions such as *Approval Status = Approved*.

7. In the **Integration Mapping Templates** page configure the mappings between the SharePoint list you specified and Sitecore.



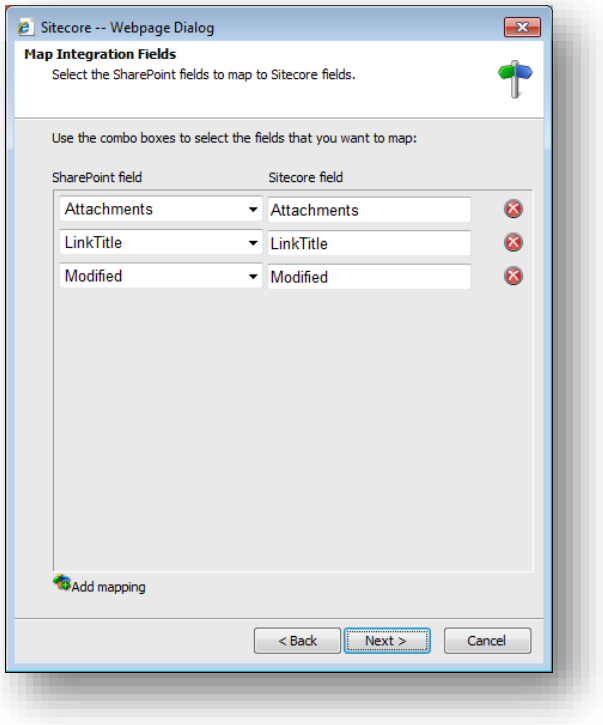
There are two options:

Create a new mapping template	Specify which fields you want to include in a new mapping template
Use existing mapping template	If you select this option, the wizard displays existing templates containing SharePoint mappings. Select the mapping template you want to use. The wizard displays a preview of the mapping. You can edit the selected mapping in the next step

Note
If you use an existing mapping template and add or update any Sitecore field in the next step Sitecore creates a new template.

8. The **Map Integration Fields** dialog box shows all the available field mappings in the SharePoint list and the corresponding Sitecore fields. You can also use this page to add or remove

mappings.

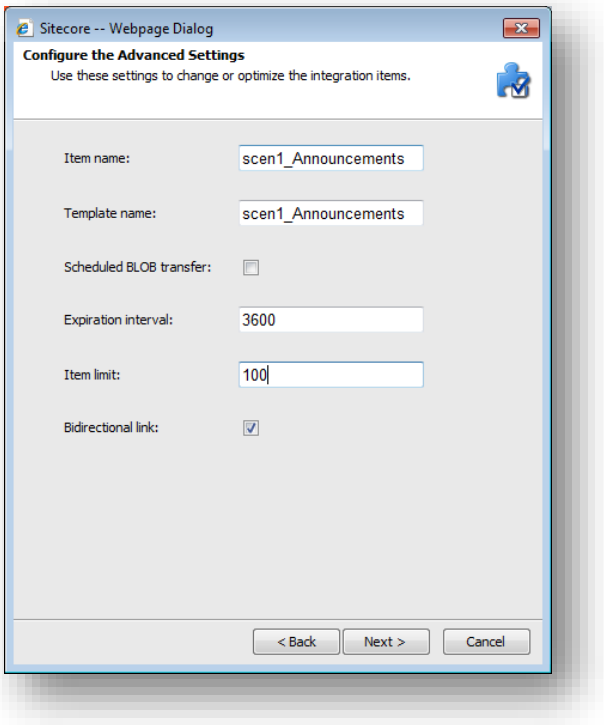


If you select an existing mapping template, you also get the option to edit the mappings.

An existing mapping template already contains specific field mappings and so does not display every field available in SharePoint.

9. In the **Configure the Advanced Settings** dialog box, you can add the following settings:

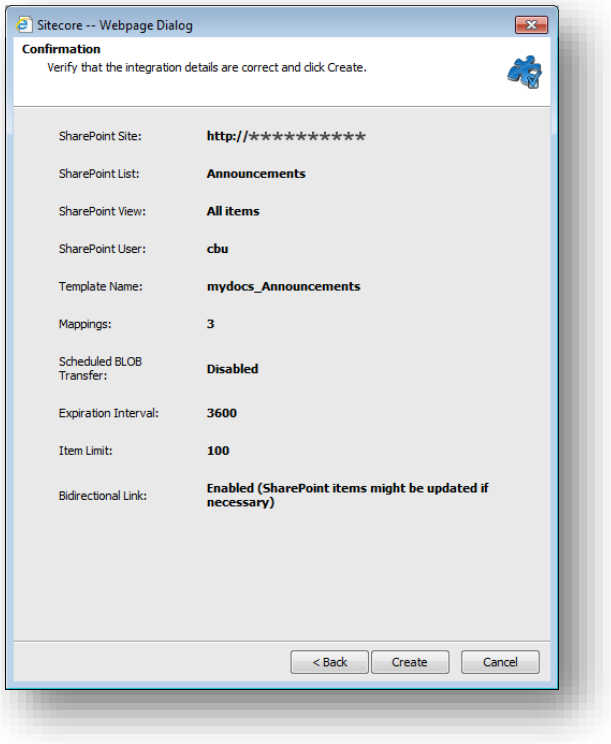
Setting	Value
Item Name	Enter an item name
Template Name	Enter a template name
Scheduled BLOB Transfer	Specify a time to import items from SharePoint. Select or clear the check box. For example, if you want to integrate a large image or movie BLOB file, it may be more efficient to import this file at a specific time to reduce the effect on performance
Expiration Interval	Real-time connection remains but updates from SharePoint only occur periodically. For example, if you enter 3600 seconds as the expiration interval, the real-time connection remains and updates come from SharePoint every hour or if you specifically request an item. Setting an expiration interval helps the Item Provider to work more efficiently and means that the SharePoint server does not have to deal with constant update requests.



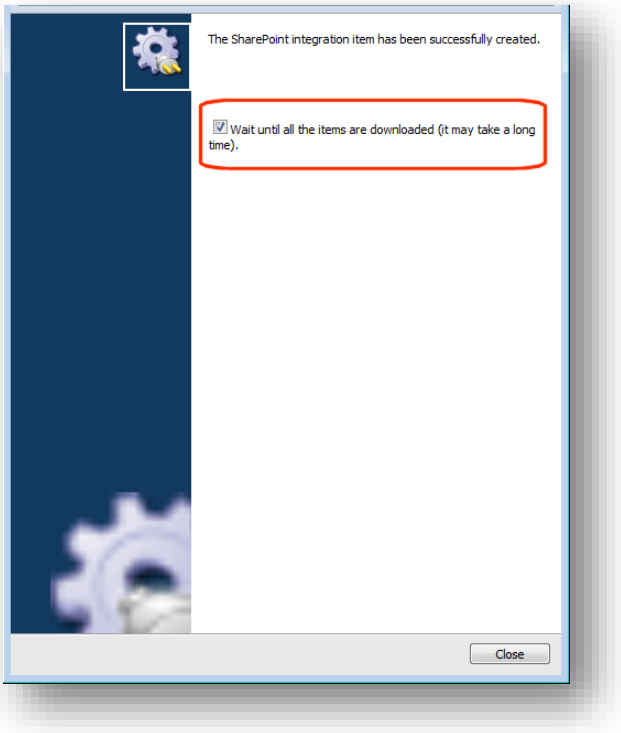
- **Item Limit** – Set an item limit. The default value = 100. Setting an item limit = 0 means that all SharePoint items for the selected list are integrated.
- **Bidirectional** – Specify whether the integration item should copy changes from both Sitecore and SharePoint to the mapped fields or only from SharePoint to Sitecore.
    - Enabled = copy changes from both Sitecore and SharePoint to the mapped fields
    - Disabled = only copy changes from SharePoint to Sitecore.

  The default value is *Enabled*.

10. In the **Confirmation** dialog box, you can review all the wizard integration details you have configured and click **Create**.



11. The final page of the wizard gives you the option to wait until you have downloaded all items that you want to integrate. This option is selected by default.



5.5 Editing the XML in a SharePoint Integration Definition Item

Mappings are XML nodes that contain information about the SharePoint lists you want to integrate, such as server, web, list, view, expiration interval plus the specific SharePoint and Sitecore fields you want to map.

The SharePoint Integration wizard provides you with a simple interface to create integration mappings. If you want to edit the XML directly, this section describes the purpose of each node and provides some examples.

A complete extract of the XML code in the *IntegrationConfigData* field:

```
        <IntegrationConfigData>
                <Server>http://my-intranet</Server>
         <Web>/mydocs</Web>
         <List>Shared Documents</List>
         <View>{B952EC0B-6E5F-4B32-9389-6521B215DAEC}</View>
         <ItemLimit>100</ItemLimit>
         <ExpirationInterval>3600</ExpirationInterval>
         <TemplateID>{276AEF08-F80A-4810-82F1-8F22F7964FB8</TemplateID>
         <FieldMappings>
          <FieldMapping>
           <Source>ows_Editor</Source>
           <Target>Editor</Target>
          </FieldMapping>
          <FieldMapping>
           <Source>ows_Modified</Source>
           <Target>Modified</Target>
          </FieldMapping>
          <FieldMapping>
           <Source>ows_LinkFilename</Source>
           <Target>LinkFilename</Target>
          </FieldMapping>
          <FieldMapping>
           <Source>ows_DocIcon</Source>
           <Target>DocIcon</Target>
          </FieldMapping>
         </FieldMappings>
         <SharepointOnline>False</SharepointOnline>
        </IntegrationConfigData>
```

To see the full XML structure in the root item created by the SharePoint Integration wizard first enable raw values in the Content Editor. Any user credentials contained in the XML are encrypted.

We do not recommend that you duplicate integration items but as an alternative to using the wizard, you can also create Sitecore integration items programmatically using the API.

Example code to create a SharePoint integration definition item programmatically:

```
        private void CreateIntegrationItem()
        {
          using (new SecurityDisabler())
          {
            string rootItemPath = "/sitecore/content/Sample Sharepoint Lists"; // Specify your
own root item path.
            var master = Factory.GetDatabase("master");
            Item rootItem = master.GetItem(rootItemPath);

            // Provide valid server name and list name.
            // Template ID will be replaced later, so you can just use ID.NewID here.
            var configData = new IntegrationConfigData("http://yourservername", "MyList",
Sitecore.Data.ID.NewID.ToString())
            {
              BidirectionalLink = true,
              ExpirationInterval = 3600,
```

```
              Web = "/myWebName",
              ItemLimit = 100,
              ScheduledBlobTransfer = false, // or 'true' for async BLOB transfer.
              View = "{59ade12c-dc93-4d53-85ff-e74daa8650c3}" // You should get this View ID from
SP server.
            };

            configData.SetCredentials("SPUserName", "SPUserPassword"); // Provide valid
credentials
            // Add custom field mappings, and do not forget to check which prefix your SP uses
for internal field representation:
            configData.FieldMappings.Add(new IntegrationConfigData.FieldMapping("ows_Field1",
"Field1"));
            configData.FieldMappings.Add(new IntegrationConfigData.FieldMapping("ows Field2",
"Field2"));
            // etc.

            // Create a template:
            Item templatesRootItem = master.GetItem(TemplateIDs.IntegrationItemTemplatesRoot);
            TemplateItem template = master.Templates.CreateTemplate("MyListTemplate",
templatesRootItem); // You may specify any template name here.
            TemplateSectionItem section = template.AddSection("SharePoint Data"); // Hardcoded
name, please do not change.
            foreach (var mapping in configData.FieldMappings)
            {
              section.AddField(mapping.Target);
            }
            using (new EditContext(template.InnerItem))
            {
              template.InnerItem[Sitecore.FieldIDs.BaseTemplate] =
TemplateIDs.IntegrationBase.ToString();
            }

            configData.TemplateID = template.ID.ToString(); // Here we set the real TemplateID,
substituting fake ID.NewID.

            var listItem = rootItem.Add("MyList", new TemplateID(TemplateIDs.IntegrationConfig));
// You may specify any convenient list name here.
            IntegrationConfigDataProvider.SaveToItem(configData, listItem);
            using (new EditContext(listItem))
            {
              listItem.Fields[FieldIDs.IsIntegrationItem].Value = "1";
            }

            ProcessIntegrationItemsOptions defaultOptions =
ProcessIntegrationItemsOptions.DefaultOptions;
            defaultOptions.Recursive = true;
            defaultOptions.AsyncIntegration = false; // or 'true' for scheduled synchronization.
            SharepointProvider.ProcessTree(listItem, defaultOptions);
          }
        }
```

Note
Duplication of a SharePoint integration definition item is not good practice so we do not recommend it. It is possible to duplicate a single integration definition item but not if the item has sub items. Also, if the definition item contains user credentials duplication will not work.

XML Nodes	Description and Example Parameters
<Server>	URL of SharePoint server to integrate with: <Server>http://my-intranet</Server>
<Web>	Name of SharePoint site to integrate: <Web>/mydocs</Web>
<List>	GUID of SharePoint List to integrate: <List>{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}</List>

XML Nodes	Description and Example Parameters
<View>	GUID of SharePoint view. <View>{B952EC0B-6E5F-4B32-9389-6521B215DAEC}</View>
<ItemLimit>	Set an item limit. <ItemLimit>100</ItemLimit>
<ExpirationInterval>	Interval between updates from the SharePoint server, set in seconds: <ExpirationInterval>100</ExpirationInterval>
<SsharepointOnline>	Defines whether the target SharePoint server is SharePoint Online.
<TemplateID>	GUID of the Sitecore template to use with integrated SharePoint list items: <TemplateID>{E24D5DB7-F665-435B-AC8D-79D65B38403A}</TemplateID>
<FieldMappings>	You can specify multiple field mappings between SharePoint and Sitecore items inside the <FieldMappings> node: <FieldMappings> <FieldMapping> <Source>ows Title</Source> <Target>Title</Target> </FieldMapping>
<FieldMapping>	Each <FieldMapping> node specifies a single SharePoint field mapped to a single Sitecore field: <FieldMapping> <Source>ows Title</Source> <Target>Title</Target> </FieldMapping>
<Source>	This specifies the name of the source SharePoint list item field: <Source>ows_Title</Source>
<Target>	This specifies the name of the target Sitecore item field: <Target>Title</Target>

Note
Use Notepad or another text editor to edit the XML file and then paste it back into the *IntegrationConfigData* field and click Save.

Chapter 6

Integration Scenarios

This section contains scenarios that demonstrate different approaches to implementing the SharePoint Integration Framework with the Sitecore in a typical business context.

This chapter contains the following scenarios:

- Page Level Integration
- Item Level Integration on a Sitecore Extranet

6.1 Page Level Integration

Use the SharePoint Integration Framework to add sample controls to a website.

About these Walkthroughs

The walkthroughs in this section demonstrate how to:

- Implement an Announcements List control
- Implement a SharePoint List control

Story

A facilities management organization has a small group of staff who use MOSS to maintain a large number of documents. The documents are typically guidelines and instructions. Draft and final versions of the documents are stored in a SharePoint document library. When a document becomes final they would like it to publish it to the company Intranet.

The same organization also has several SharePoint document libraries containing other documents such as contracts. They would like to use SharePoint to store these documents and make them available to staff on the company Intranet. They would also like to be able to post announcements on the Intranet informing staff of any news related to the publication of company documents.

Personas

Jane - SharePoint Editor/Contributor

Jane is responsible for creating and editing multiple Word documents stored in several SharePoint document libraries. Documents can be in two possible states; draft or final. If a document is in its final state, Jane uses SharePoint workflow to review and publish the document to the company Intranet. Once she has approved a document for publication, it automatically appears in the *Published* view of the SharePoint document library.

Peter – Sitecore/SharePoint Developer

Peter is the company .NET developer. He is an experienced Sitecore C# developer and knows the fundamentals of MOSS. He must implement an integration solution that allows Jane to achieve all her objectives.

Prerequisites

You need the following items to complete this walkthrough:

- A SharePoint Server
- Webs – One or more SharePoint websites
- An Announcements list with sample announcements.
- One or more document libraries – For example, Guidelines, Instructions and Contracts
- Sample draft and final Word documents

6.1.1 Implementing an Announcements List Control

Introduction

Jane wants to be able to post announcements related to newly published staff documents. She asks Peter (SharePoint developer) to make it possible to display announcements for guidelines and instructions on the staff Intranet.

In the SharePoint Integration Framework, the Announcement List and Task List controls are the simplest category of sample controls to implement. In this task, Peter uses the Announcements List control to display staff announcements related to published documents on the company Intranet site.

Creating a Sitecore Site Section and Sub Section

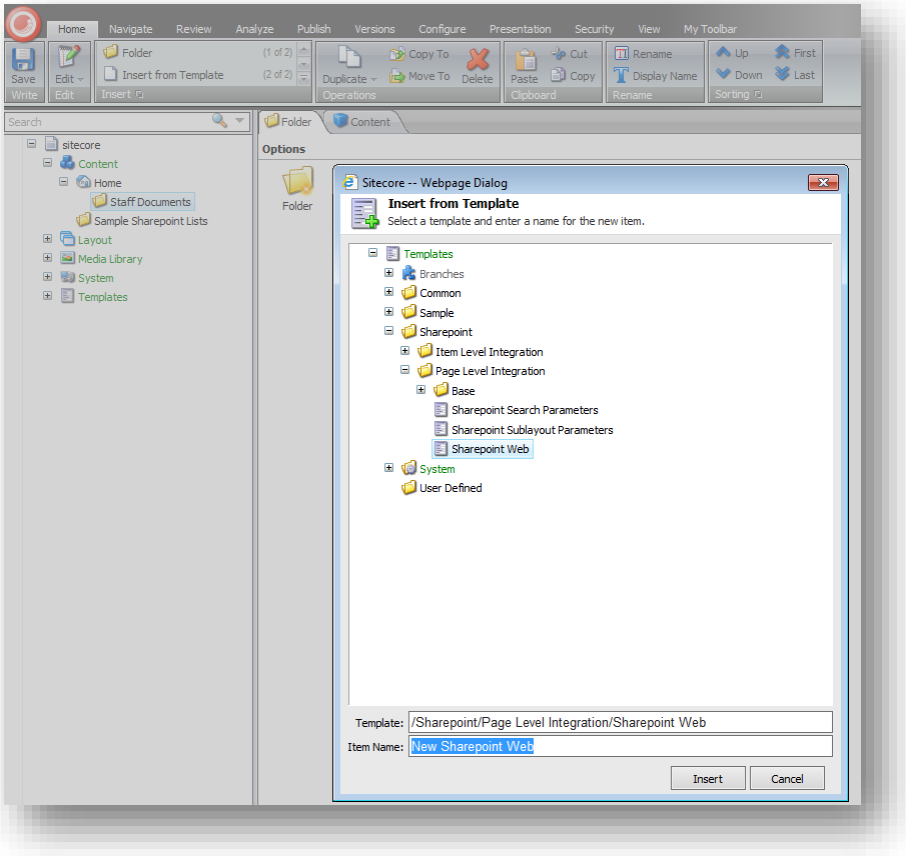
Peter must first create a new site section and a sub section to display the SharePoint lists.

To create a site section:

1. Open the Content Editor and in the content tree select **Home**.
2. On the ribbon, click the **Home** tab. In the **Insert** group, select the *Folder* template and name the new site section *Staff Documents*.

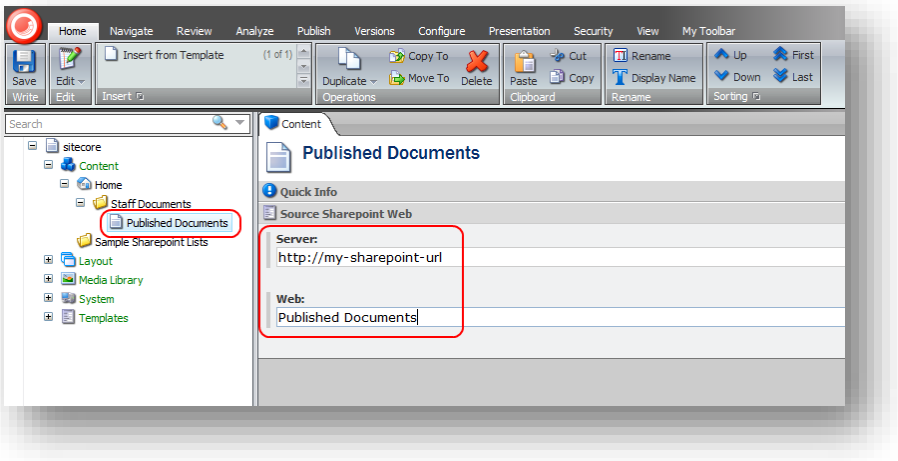
To create a sub section:

1. Select *Staff Documents*. Create a new content item from the *SharePoint Web* template. Insert an item from the *SharePoint Web* template.



2. Name the sub section *Published Documents*.

3. In the **Web** field of the new item, enter *Published Documents* as the name of the SharePoint website. This must be the actual name of your SharePoint Web not the display name.
4. In the **Server** field, enter the URL of the SharePoint server you want to connect. For example, *http://my-sharepoint-url*



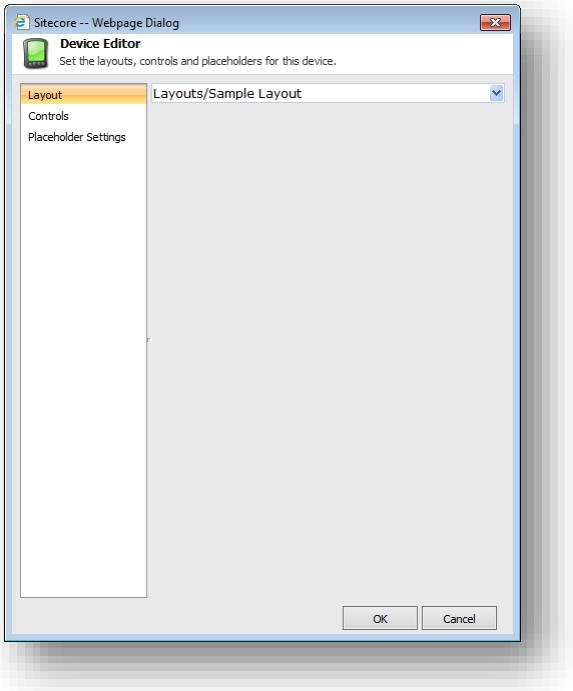
Adding the Announcements List Control

To display SharePoint announcement lists on the *Published Documents* item, Peter must add a control to the layout details. In Layout Details, he uses the *Sample Layout* layout, and then adds the *Announcements List* control to the presentation layer.

To add an Announcements List control:

1. In the Content Editor, content tree select the *Published Documents* content item.
2. On the ribbon, click the **Presentation** tab.
3. Click **Details** to open the **Layout Details** dialog box.
4. In the **Layout Details** dialog box, under the **Default** layout details click **Edit**.

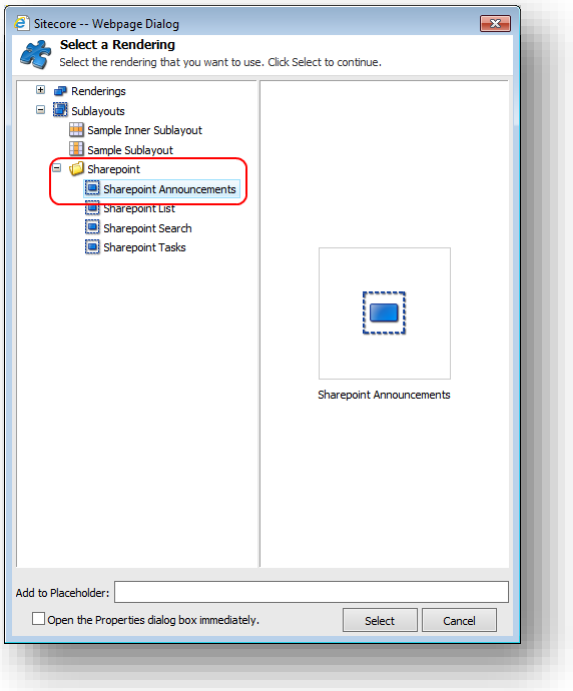
5. In the **Device Editor**, select **Layout** and then select *Sample Layout*.



6. Select **Controls** and add the *Sharepoint Announcements* list sublayout.
7. In the **Device Editor** dialog box, select **Controls** and click **Add**.

   Use the following path to locate the correct sublayout:

   `Sublayouts/Sharepoint/Sharepoint Announcements`



8. To save your changes, click **OK** and then click **OK** again in all open presentation dialogs.
9. On the ribbon, click **Save**.

Configuring the Announcements List Control

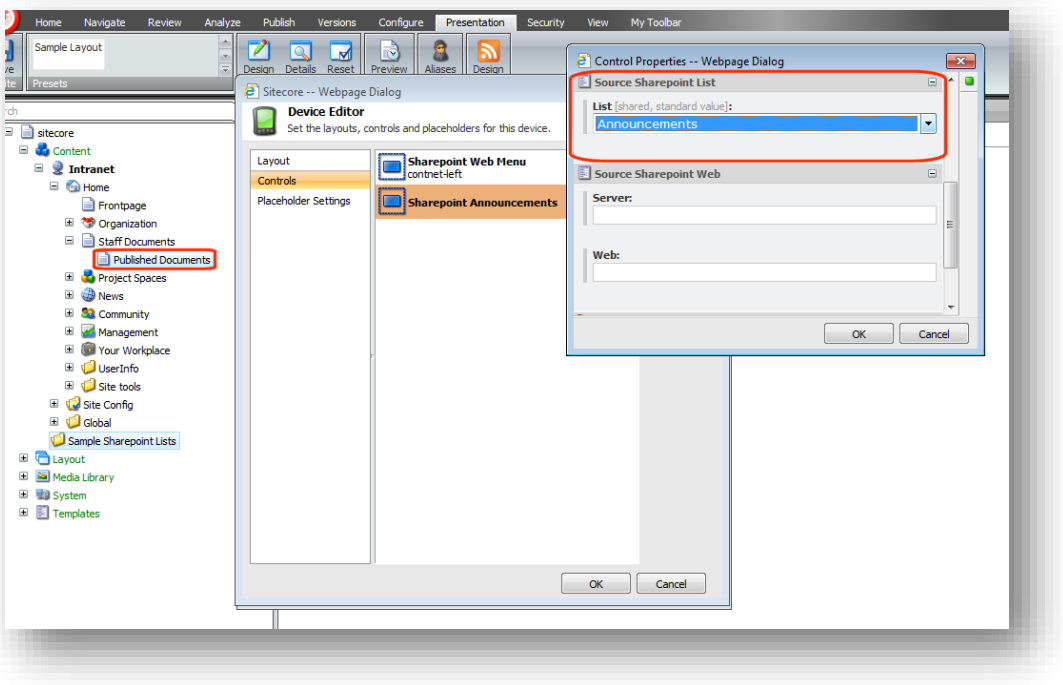
After Peter has added the Announcements List control, he must configure the control properties.

To configure the Announcements List control properties:

1. In the **Layout Details** dialog box, select the *Sharepoint Announcements* list control and then click **Edit**.
2. In the **Control Properties** window, in the **Sharepoint** group, you can select the following properties:

Property	Setting
List	*<name of sharepoint list>*
Server	*<http://sharepoint server>*
Web	*<name of sharepoint web>*

3. In the **List** field, select *Announcements.*



If the list you want to add does not appear in the drop-down, add it to the available list names.

For more information about how to define list items, see the section *Creating a List Value*.

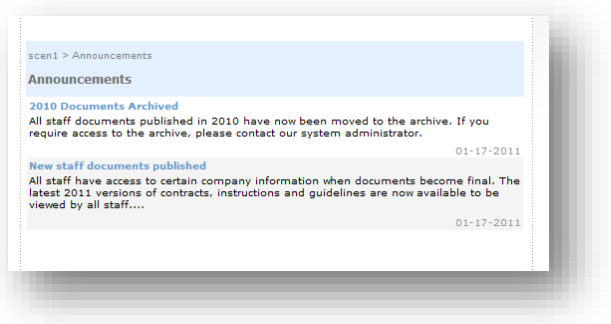
When you use the *Announcements List* control with the *Sharepoint Web* template, enter the *Website* and *Server* details on the item and not in properties.

4. Specify a placeholder setting to define where you want the control to appear. In the **Placeholder** field, enter *main*.
5. Click **OK**.

6. On the ribbon, click **Save** to save the changes you made to this item.

Previewing the Announcements List Control

When Peter has completed all the steps in this task, he can preview the announcements sample control in Sitecore preview mode or in a web browser.



It is now possible to see two announcements from SharePoint on the company Intranet page.

6.1.2 Implementing a SharePoint List Control

Jane asks Peter to make the final versions of documents available on the company Intranet. To do this Peter decides to create a new SharePoint view for published documents and a new site section on the company Intranet called *Published Documents* where staff can access the final versions.

First, he must create a new view in each SharePoint document library for final versions of documents. He will call the new view *Published*. He can then make the documents available to staff on the company Intranet using the SharePoint List control.

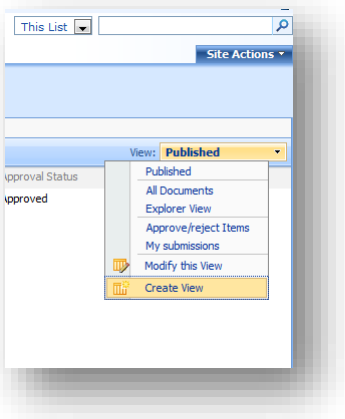
The SharePoint List control is the most complex but the most flexible control because you can use it to display any type of list.

Creating a SharePoint View

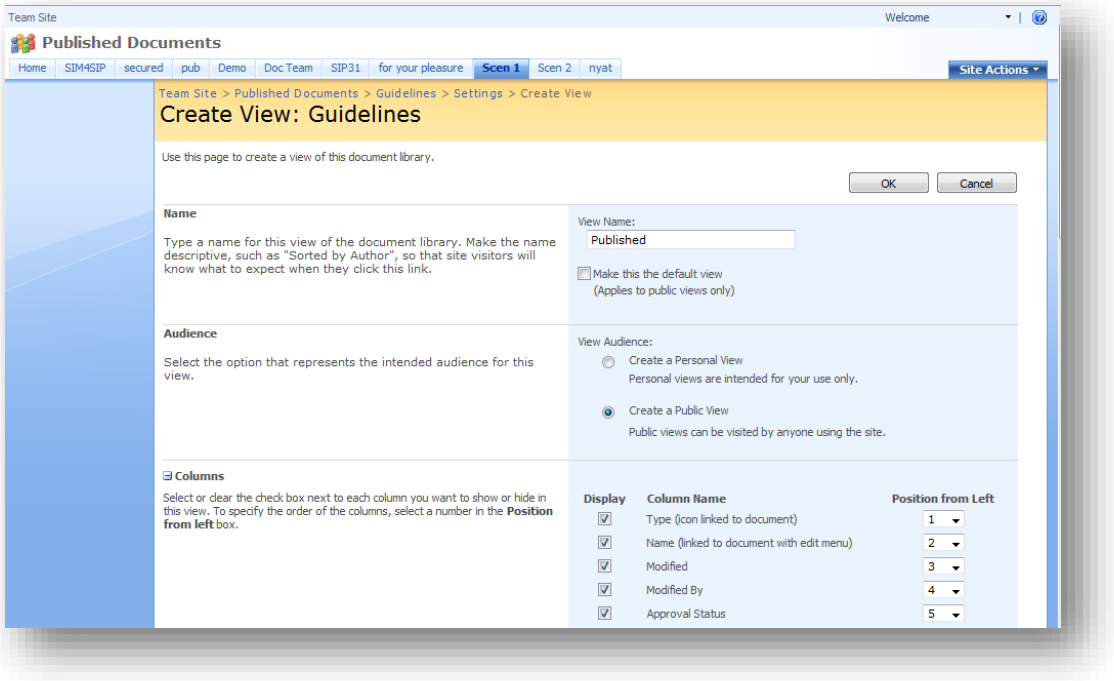
Peter first creates a *Published* view to display final versions of Word documents and decides to use SharePoint workflow to move documents from a draft to final state.

To create a new SharePoint view:

1. In your SharePoint web, select the *Guidelines* document library.
2. In the **View** menu, select **Create View**.



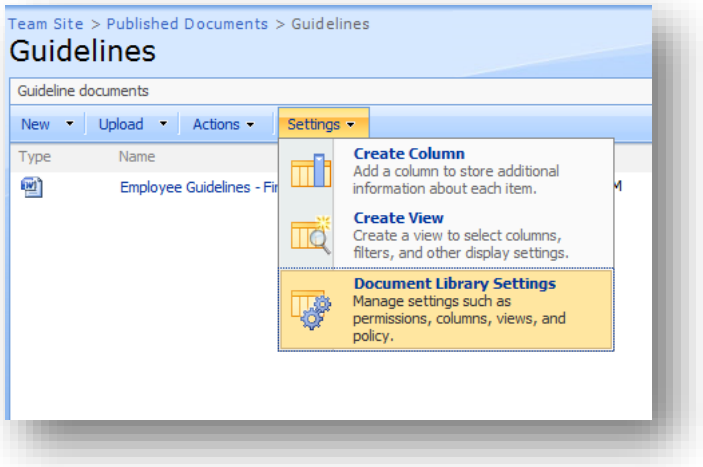
3. In **Create View**, select **Standard View** and enter the name *Published*.



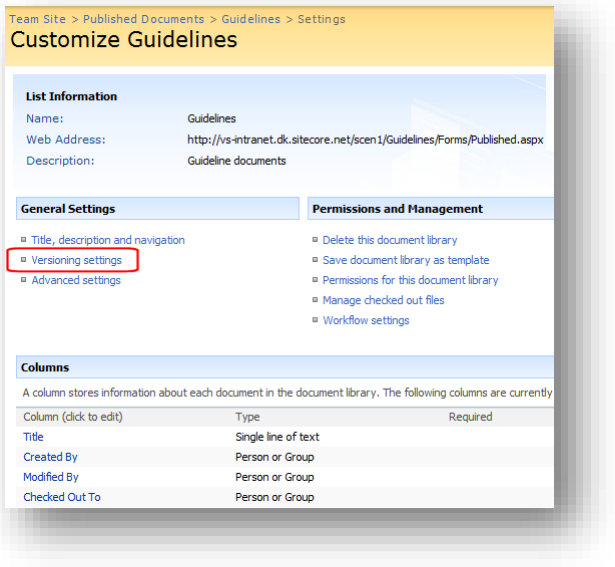
4. Click **OK**.

To activate versioning and workflow:

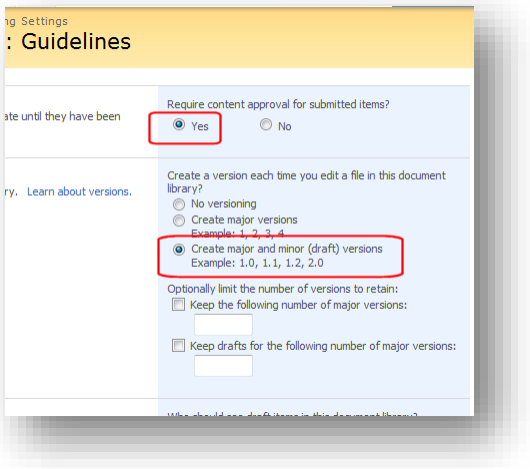
1. In the *Guidelines* document library, select **Settings**.
2. Then select **Document Library Settings**



3. Under **General Settings**, select **Versioning Settings**.



4. In **Versioning Settings**, under **Content Approval**, select *Require content approval for submitted items*.
5. Under **Document Version History**, select *Create major and minor (draft) versions*.

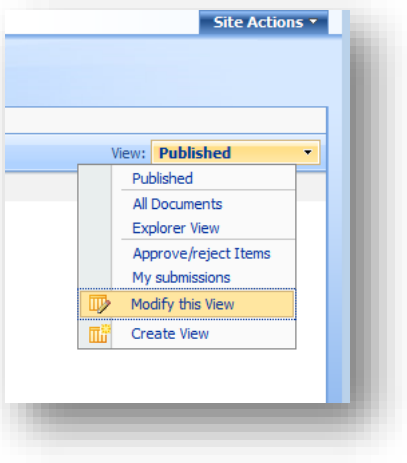


6. Click **OK**.

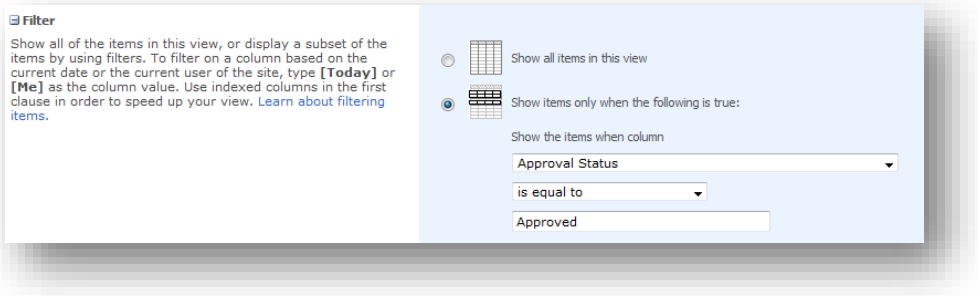
Peter creates a filter on the *Published* view so that only documents approved as *final* in the workflow appear in the *Published* view.

1. In the *Guidelines* document library, select the **Published** view.

2. Select **Modify this View**.

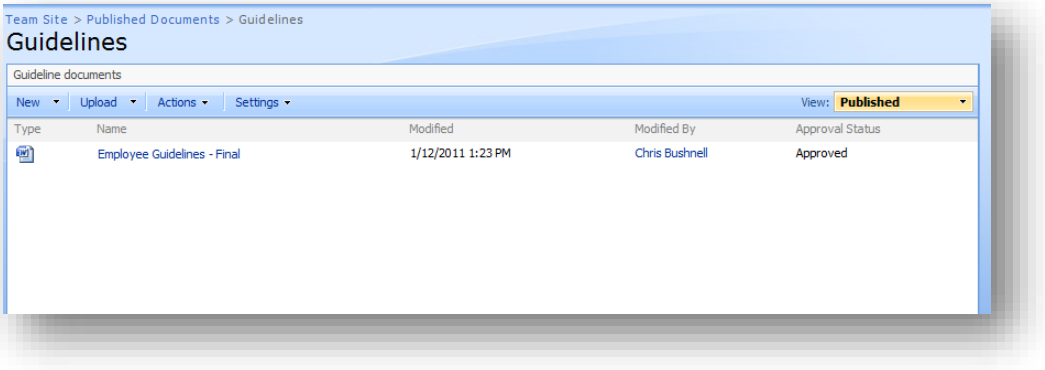


3. In the **Filter** panel, create the following filter rule: *When Approval Status is equal to Approved*.



4. Click **OK**.

Now when you select the *Published* view in SharePoint, only final versions of the Word documents are visible.



Follow the same steps for each SharePoint document library that you want to publish.

Adding Drop-Down List Items to Control Properties

Before adding a control to display his list, Peter notices that in Control Properties, only *Shared Documents* appears by default in the drop-down list. He wants to be able to select document libraries with

other names, such as *Guidelines, Instructions* and *Contracts*. To do this he knows that he must first create new drop-down list items for each document library.

For instructions on how to create a new list value items, see Creating a List Value.

Repeat the same steps for each document library that you want to add.

Adding a Sharepoint List Control

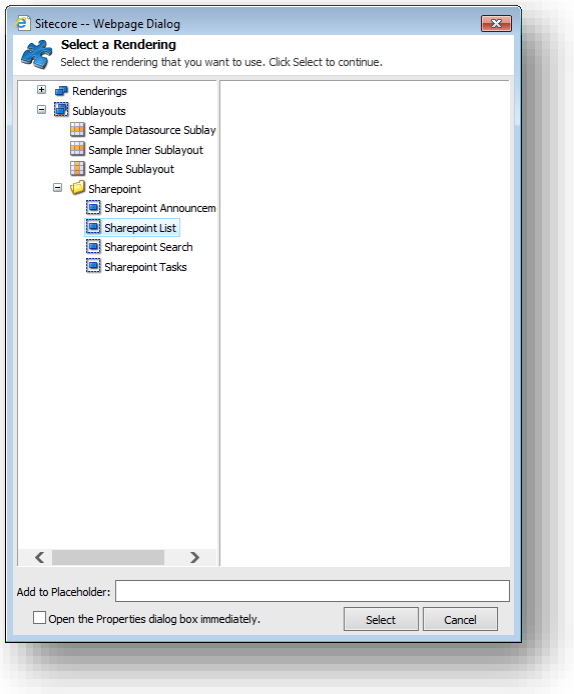
Peter is ready to add sample controls to the *Published Documents* section to display the following SharePoint document libraries:

- Guidelines
- Instructions
- Contracts

To do this, he decides to use the *Sharepoint List* sample control.

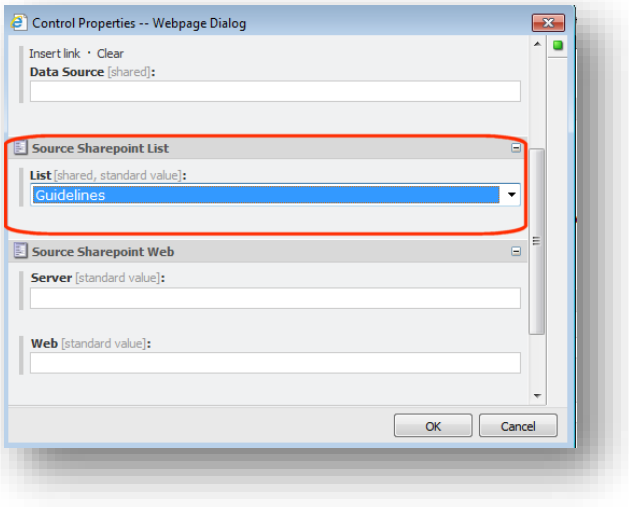
To add a *Sharepoint List* control:

1. In the content tree, select the *Published Documents* site section you created earlier.
2. On the ribbon, click the **Presentation** tab, then in the **Layout** group click **Details**.
3. In the **Layout Details** dialog box, under **Default** layout details click **Edit**.
4. In the **Device Editor** dialog box, select **Controls** and click **Add**.
5. In the **Select a Rendering** window, select the *SharePoint List* sublayout and click **OK**.



6. In the **Device Editor** dialog box, select the *SharePoint List* control and click **Edit**.

7. In the **Properties** window, select the *Guidelines* document library (one of the list items you added in the previous step).



8. Specify a placeholder setting to define where you want the control to appear. Enter *main* in the **Placeholder** field.
9. Click **OK** and then click **OK** again to close all open presentation dialogs.

Follow the same steps for the *Instructions* and *Contracts* document libraries. Add a new *Sharepoint List* control for each document library you want to add.

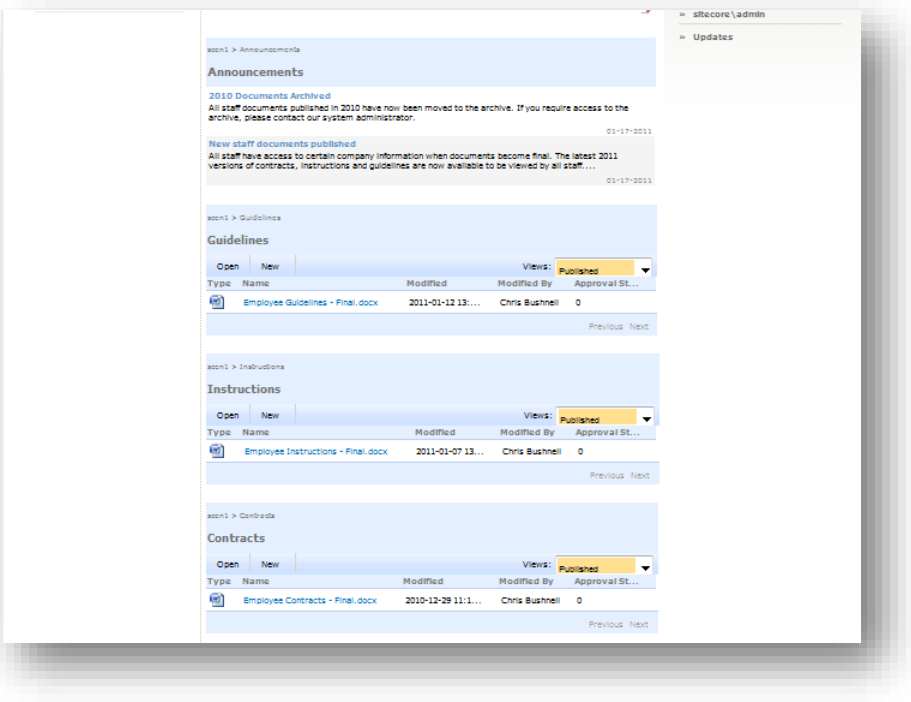
Previewing the Sharepoint List Control and Selecting Views

Peter is now ready to preview the new *Published Documents* site section. This section uses three *Sharepoint List* controls and one *Announcements List* control

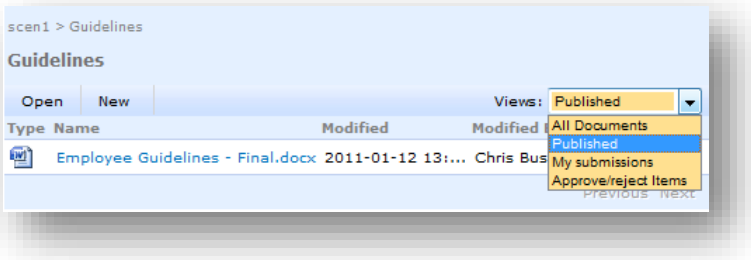
To preview the Sharepoint List control and select a view:

1. Open your Sitecore website in a new browser window.
2. Navigate to the *Staff Documents* site section.

The control displays all announcements at the top of the page and the document libraries *Guidelines*, *Instructions* and *Contracts* embedded below.



3. To change document library views, click the **Views** drop-down control and select a different SharePoint view.



Note
In this scenario, Published is the default view in SharePoint so it is also the default view in Sitecore. Change the default view in SharePoint.

6.2 Item Level Integration on a Sitecore Extranet

The following scenario demonstrates how to implement the SharePoint Integration Framework using the SharePoint Integration wizard and a Sitecore extranet website.

About these Walkthroughs

The walkthroughs in this section demonstrate how to:

- Map SharePoint announcements to the Sitecore content tree.
- Map SharePoint document libraries to the Sitecore Media Library.

Story

A quality assurance company serves among other clients the European financial services industry and automotive industry. Standards and rules are changing at a high rate and changes must appear in the documentation immediately. They want to make changes available to customers right away.

The company authors its own documents internally and stores them in several SharePoint document libraries. They also have a company extranet powered by Sitecore. When the documents are final they want to publish them on the customer portal extranet web site.

Personas

Miriam – SharePoint Editor/Contributor

Miriam creates and updates Word documents and stores them in several SharePoint document libraries. She has created document libraries for product testing, improvement plans, and review processes. She wants to make some of these documents available to customers on the company extranet.

Johan – Sitecore developer

Johan has many years of experience working with Sitecore but is not so familiar with SharePoint. The company extranet uses Sitecore, so he wants to utilize the power of Sitecore to manage integration items natively.

Aims

Use the SharePoint Integration wizard to import documents and lists from SharePoint to Sitecore. Once the required lists are in the Sitecore content tree use functionality such as publishing, versioning and workflow to manage the items. Publish content to the customer portal using appropriate presentation options.

This task makes SharePoint documents available as Sitecore items as soon as they are ready and means that any changes made to a document appears on the website straight away.

Prerequisites

You need the following to complete this walkthrough:

- A SharePoint server
- Webs – One or more SharePoint websites
- One or more announcements lists
- One or more document libraries
- Sample draft and final Word documents

6.2.1 Integrating SharePoint Announcements with the Sitecore Content Tree

Miriam asks Johan if he can find a way for her to publish announcements related to the release of documents straight to the customer portal extranet site. Johan knows that if he uses the SharePoint Integration wizard he can work with SharePoint list content directly in Sitecore. This will give him more control over the presentation of the announcements.

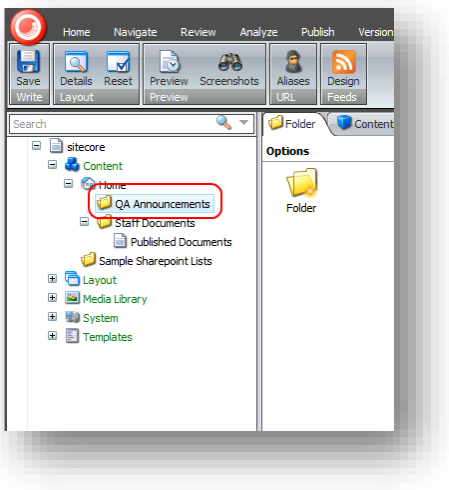
Johan decides to use the SharePoint Integration wizard to map SharePoint announcements to Sitecore. He will then publish the announcements to the customer portal extranet.

Creating a SharePoint Integration Definition Item in Sitecore

To map announcements to Sitecore, Johan knows that he must first create a SharePoint Integration definition item in Sitecore using the SharePoint Integration wizard.

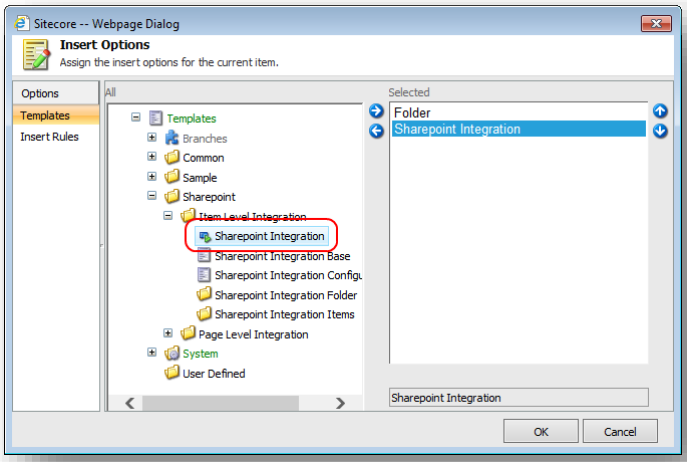
To create a SharePoint Integration definition item:

1. In the content tree, create a new folder to store your integration items.



Enter a name for the folder, for example *QA Announcements*. This folder will contain the SharePoint definition items. Integration items appear as sub items of the SharePoint definition item.

2. To use the SharePoint Integration wizard to create a SharePoint Integration definition item, add the wizard to your **Insert Options**.

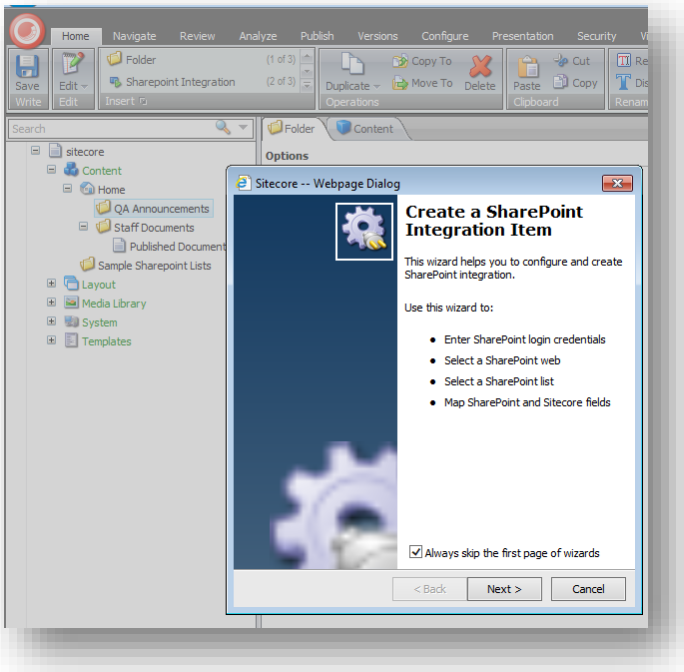


Using the Wizard to Map SharePoint Announcements

Johan uses the SharePoint Integration wizard to configure integration items. This makes it easy for him to map fields and configure other settings without the need to edit XML.

To use the wizard to map SharePoint announcements:

1. In the content tree, select the *QA Announcements* content item.
2. On the **Home** tab, click **SharePoint Integration** to open the SharePoint Integration wizard.



3. In the **Connect to a SharePoint Site** page, provide connection details and enter a SharePoint URL.
   - Enter the URL to a SharePoint server.
   - Enter your SharePoint server credentials in the `sharepoint.config` file or in the *Credentials for Sharepoint server* fields in the wizard. If you have added default credentials to the `sharepoint.config` file the wizard will find these automatically.
4. In the **Select a SharePoint List** page of the wizard select the *Announcements* list.
5. In the **Select a SharePoint View** page, select a view. For example, *All Items*.
6. In the **Integration Mapping Template**, either use an existing mapping template or create a new template to save your own mappings.

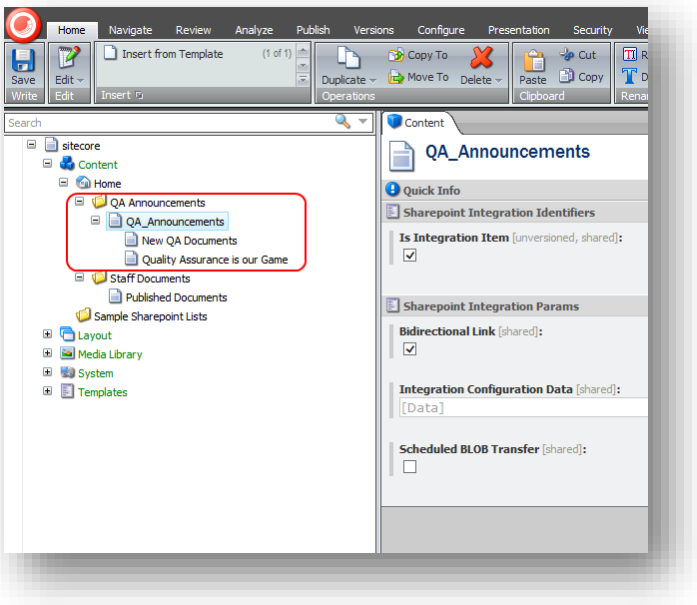
   Map SharePoint fields with similar or equivalent Sitecore fields. An example XML field mapping that maps the SharePoint body field with the Sitecore body field.

```
<Source>ows_Body</Source>
      <Target>Body</Target>
```

7. In the **Map Integration Fields** page check that you have the correct SharePoint fields mapped to the correct Sitecore fields. You can also add and remove mappings.
8. Configure **Advanced Settings**, for example set an expiration interval and select scheduled BLOB transfer, if appropriate.

   Set the expiration interval to 100 seconds.
9. In the **Confirmation** page, check your settings and click **Create**.

When you have completed all the pages in the wizard, the SharePoint list items that you integrated appear straight away as content items in the content tree under the node you specified. In this example, they appear under the *scen2_Announcements* SharePoint integration definition item.



Configuring Presentation

Item level integration allows you to integrate SharePoint list items with Sitecore without the need to use special SharePoint controls to display content on your website. Each list item that you integrate appears as a separate content item in Sitecore.

Some presentation options include:

- Use standard Sitecore renderings and sublayouts - For example, you can display announcement text using the *Sample* rendering control.
- Create custom renderings or sublayouts - You can create your own custom control to display documents that come from a SharePoint document library.

Configuring Other Sitecore Options

Configure Sitecore workflow, publishing or versioning on your integration items. For example, you could integrate all document library content with Sitecore and then use Sitecore workflow rather than SharePoint workflow to handle the publication of documents to the extranet.

Previewing Integration Items on an Extranet

To preview integration Items on an extranet:

- In the Content Editor, on the ribbon, click **Publish** and then click **Preview** to view the content that you have integrated with Sitecore.

OR

- Open a new browser window and navigate to your extranet site.

6.2.2 Integrating SharePoint Document Libraries with the Sitecore Media Library

Introduction

Miriam asks Johan if he can find a way for her to publish final versions of QA Word documents to the customer portal extranet site. Johan wants to use item level integration again but this time decides to integrate the SharePoint document libraries with a Media Library folder. He thinks that the functionality available in the Media Library is more suitable for SharePoint document libraries than the Sitecore content tree.

Johan uses the SharePoint Integration wizard to map a SharePoint document library to Sitecore. He then uses Sitecore to publish the Word documents on the customer portal extranet.

SharePoint Prerequisites

In SharePoint create the following quality assurance document libraries:

- Standards
- Testing
- Improvement
- Review

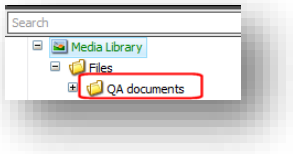
Upload sample Word documents to each document library.

Using the Wizard to Integrate a SharePoint Document Library

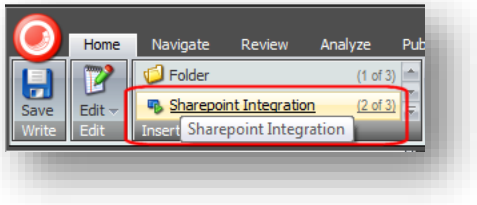
Johan uses the SharePoint Integration wizard to map the Word documents for each of Miriam's document libraries to the Sitecore Media Library.

To use the SharePoint Integration wizard to integrate a SharePoint Document Library:

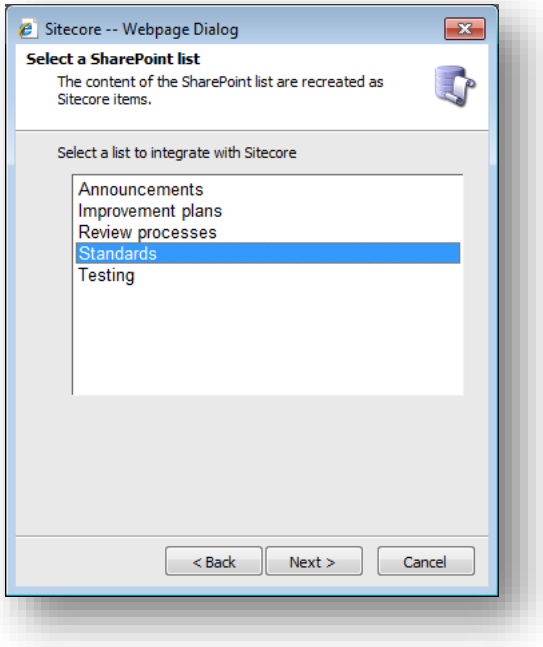
1. Select a suitable node in the Media Library content tree and create a new folder for your integration items. Name the folder *QA documents*.



2. In the Media Library, select the *QA Documents* media content item.
3. Open the SharePoint Integration wizard.



4. In the **Connect to a SharePoint Site** page, provide user credentials and enter a SharePoint URL.
5. In the **Select a SharePoint List** page, select the document library that you want to integrate, for example, *Standards*.

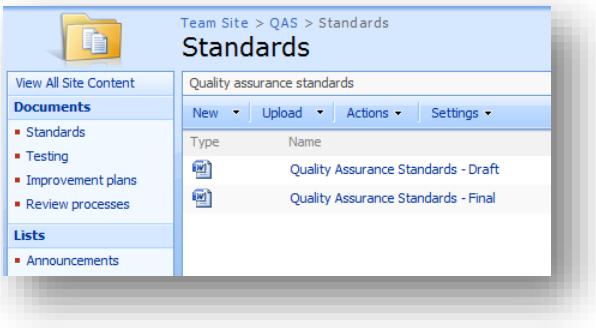


6. In the **Select a SharePoint View** page, select a view. For example, *All Documents*.

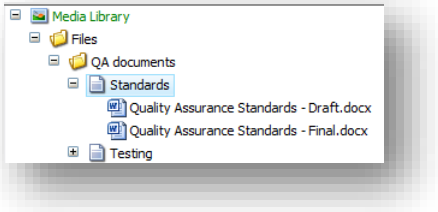
7. In the **Integration Mapping Template**, either use an existing mapping template or create your own mappings.
8. In the **Map Integration Fields** page, check the default mappings. Accept the default mappings or you can add or remove more fields.
9. Configure **Advanced Settings**, for example set an expiration interval and select scheduled BLOB transfer, if appropriate.
10. Click **Create**.

    The list items that you mapped now appear in the Media Library content tree under the SharePoint integration definition item you created.

SharePoint Document Library
(Source)



Sitecore Media Library
(Destination)



Configuring Presentation

Having more experience with Sitecore than SharePoint, Johan knows that once he has integrated SharePoint lists with Sitecore he has more control over presentation and other options. He can use standard Sitecore controls or use the API to create his own custom controls.

Configuring Other Sitecore Options

Configure Sitecore workflow, publishing or versioning on your integration items. Johan decides to use Sitecore publishing and workflow to manage the QA documents and publish them to the staff extranet.

Previewing Integration Items on an Extranet

When Johan has finished using the SharePoint Integration Framework to integrate the QA document libraries, he wants to preview them on the staff extranet site.

To preview integration items on an extranet:

1. Open a web browser and enter the URL of your extranet site.
2. Navigate to the section of your site that displays your SharePoint content.

For more information about the SharePoint Integration wizard and how to edit the XML file, see the section *Item Level Integration*.