## Extending SharePoint Server Search

**Lab Time**: 45 minutes

**Lab Folder**: C:\Student\Labs\Search

**Lab Overview**: SharePoint 2010 offers several new ways to customize and extend enterprise search capabilities. In this lab, you will create a search application for finding and managing tasks.

Lab Setup Requirements

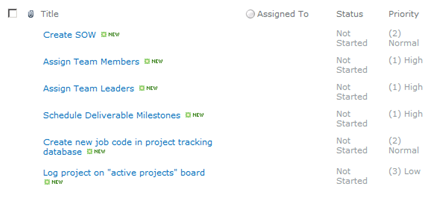
* Before you begin this lab, you must run the batch file named **SetupLab.bat**. This batch file creates a new blank site collection at the location **http://intranet.wingtip.com/sites/SearchLab**. This is the site you will use to test and debug the code you are going to write with the SharePoint Developer Tools.

### Exercise 1: Customizing Search Center

In this exercise, you will customize the Search Center to create the foundation for a search-based application. You will create a task list as a data source and then add a scope for the list. Finally, you will create a Search Center with custom settings to search the new scope.

**IMPORTANT**: Before you begin these steps, make sure that you have the Standard, Enterprise, and Publishing features activated at both the Site Collection and Site levels.

1. If you haven’t already done so, run the batch file named **SetupLab.bat**, found in the folder for this lab to create the new site collection that will be used to test and debug the code you will be writing in this lab. This batch file creates a new site collection at an URL of **http://intranet.wingtip.com/sites/SearchLab**. Launch Internet Explorer and browse to this site
2. Verify that you have the **Standard, Enterprise, and Publishing** features activated at both the **Site Collection** and **Site levels:**
   1. Select **Site Actions » Site Settings » Site Collection Features** (in the Site Collection Administration section).
      1. Verify that the three Features are active or activate them if necessary.
      2. Use the breadcrumbs to navigate back to **Site Settings**.
   2. Select **Site Actions »** **Manage Site Features**.
      1. Verify that the three features are active or activate them if necessary.
3. Now you need to create a new task list to hold some sample data we’ll index:
4. Select **Site Actions » View All Site Content**.
5. Click **Create** and create a new **Tasks** list for the site named **Project Tasks**.
6. Using the picture below, add these tasks to the list. Make sure to use the High/Normal/Low priority settings on these tasks as you will configure custom relevance rankings in a later exercise.



1. Now you need to create a **Search Center** sub site, but before doing that, you need to verify the necessary features are active.
2. Select **Site Actions » Site Settings**.
3. Pick **Site Collection Administration » Site collection features**.
4. Verify the following features are active. If they aren’t, activate them:
5. Office SharePoint Server Enterprise Site Collection features
6. Office SharePoint Server Publishing Infrastructure
7. Office SharePoint Server Standard Site Collection features
   * 1. Verify the similar corresponding three (3) features are active at the site level (**Site Actions » Site Settings » Manage site features** under the **Site Actions** section).
8. Next, we need to create a new **Search Center** sub site in the **SearchLab** site.
9. Select **Site Actions » New Site**. Use the following information to create the subsite:

**Title**: Search Center

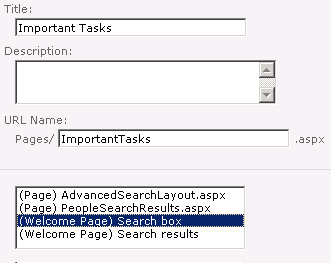
**URL**: searchcenter

**Template**: Search » Enterprise Search Center

1. With the **Search Center** site created the next step is to modify the publishing pages within the Search Center. First create a new page that will search for important tasks:
2. Browse to **http://intranet.wingtip.com/sites/SearchLab/searchcenter**. On the new search site, click **Site Actions » View All Site Content**.
3. Select the **Pages** library.
4. In the Pages library, click **Documents » New Document » Page**.
5. Select to create a new **(Welcome Page) Search Box** and use the following to create the page:

**Title**: Important Tasks

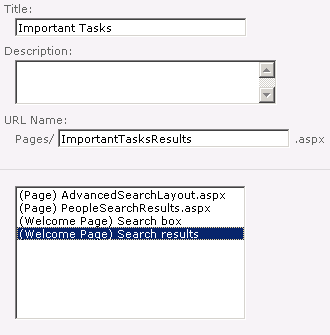
**URL Name**: ImportantTasks.aspx



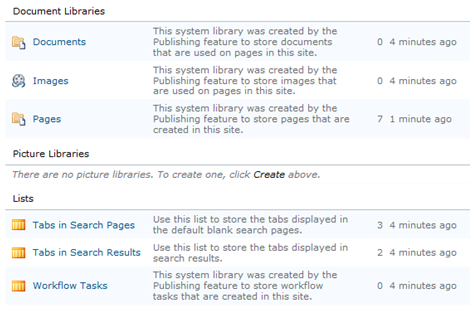
1. Next create the search results page for important tasks:
2. In the **Pages** library, click **Documents » New Document » Page**.
   1. Select to create a new **(Welcome Page) Search Results** and use the following to create the page:

**Title**: Important Tasks

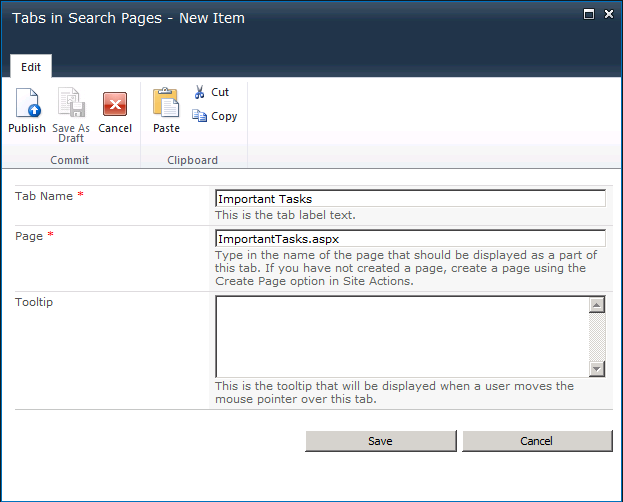
**URL Name**: ImportantTasksResults.aspx



1. Now you need to add a new tab to the **http://intranet.wingtip.com/sites/SearchLab/searchcenter** homepage.
2. On the new search site, click **Site Actions » View All Site Content**.
3. Click the **Tabs in Search Pages** library.



* 1. Select **Items » New Item** from the ribbon and use the following to create the new tab:
     1. **Tab Name**: Important Tasks
     2. **Page**: ImportantTasks.aspx



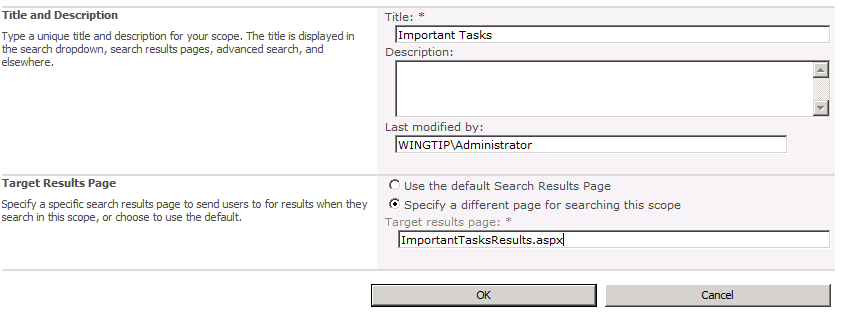
1. Click the **Save** button.
2. With the new tab created, you now need to configure that tab to send users who execute searches on it to the custom results page.
3. Browse to **http://intranet.wingtip.com/sites/SearchLab/searchcenter**.
4. Click the **Important Tasks** tab to jump to the ImportantTasks.aspx page.
5. Using the ribbon select **Page » Edit**.
6. In the Web Part that contains the search box, click the Web Part Menu drop down arrow in the right upper corner of the web part and pick **Modify this Web Part**.
7. In the **Search Box** task pane, expand the **Miscellaneous** section. Change the **Target search results page URL** from results.aspx to ImportantTasksResults.aspx and click **OK** at the bottom of the task pane.



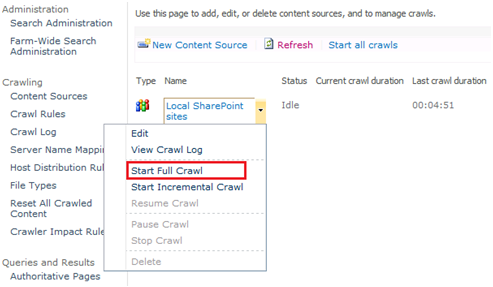
1. Now check the page in to save your changes. You can find the **Check in** button on the **Page** tab.

Now that the new Search Center is created, you will create a search scope to use with it. The search scope will include only the task list you created earlier.

1. Open the **SharePoint 2010 Central Administration** site.
2. Click **Application Management » Manage Service Applications**.
3. On the **Manage Service Applications** page, click the **Search Service Application** link. This brings you to the **Search Administration** page.
4. Click the **Scopes** link in the **Quick Launch** bar under the **Queries and Results** section.
5. On the **View Scopes** page, click **New Scope**.
6. On the **Create Scope** page, enter **Important Tasks** in the **Title** field.
7. Enter ImportantTasksResults.aspx for the **Target Results Page** and click the **OK** button.

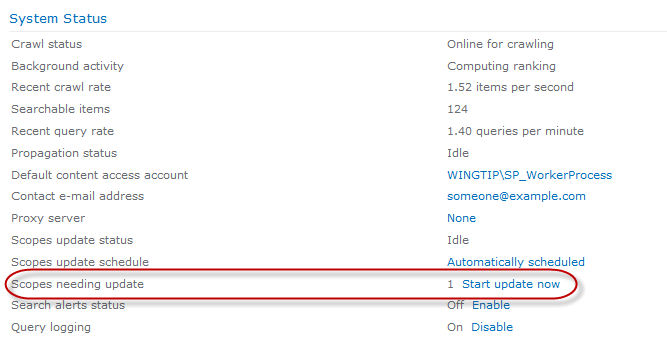


1. When the new scope is created, click the **Add Rules** link in the **Upgrade Status** column to add a rule to the scope. We will filter the results to include only those from our list we created previously.
2. Leave the **Web Address** option selected and enter the complete URL to the task list (e.g. **http://intranet.wingtip.com/sites/SearchLab/Lists/Project%20Tasks**) in the **Folder** field in the **Web Address** section.
3. Click the **OK** button.
4. With the scope created you now need to start a full crawl.
5. Click the Content Sources link in the Quick Launch under the Crawling section.
6. Select the **Local SharePoint sites** and then **Start Full Crawl** to trigger the search indexer to index all the content in our new task list.



* 1. Wait for the full crawl to finish (about 5 minutes).

1. Finally, you need to update the scope. This will happen automatically but not for quite a while. Let’s force it to update the scope now:
2. Select **Search Administration** from the top of the **Quick Launch**.
3. In the System Status section, you’ll see that the 2nd to last item reports one scope is pending update. Click the Start update now link to initiate it.

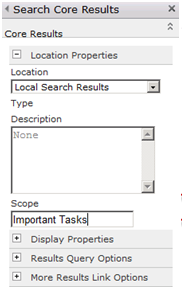


Once the new scope is ready, you must return to the **Search Center** and edit the web parts so that they will use the new scope.

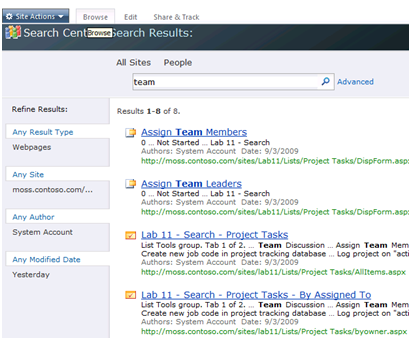
1. Return to the Search Center (**http://intranet.wingtip.com/sites/SearchLab/searchcenter**) and click the **Important Tasks** tab. This will take you to the ImportantTasks.aspx page.
2. Enter a term and perform a search. This will take you to the ImportantTasksResults.aspx page.
3. Select **Site Actions » Edit Page** to place the page in edit mode.
4. In the Web Part that contains the search box, click the arrow in the right upper corner of the Web Part and pick Edit Web Part.
5. In the Search Box task pane, expand the **Miscellaneous** section. Change the **Target search results page URL** from results.aspx to ImportantTasksResults.aspx and click **OK** at the bottom of the task pane.
6. On the **Search Core Results Web Part**, select Edit Web Part from the Edit menu (in the right upper corner of the web part).

**Note**: Once you select the edit option you may need to scroll the web page up to the top to see the Search Core Results edit options.

1. Under the **Location Properties** section, select **Local Search Results** in the **Location** property and enter **Important Tasks** in the **Scope** property.



1. Click the **OK** button.
2. **Check In** the page.
3. You should now be able to return results from the task list you created earlier.



In this exercise you created a new Search Center site and customized it to use a new search results page that leverages a custom scope you created.

### Exercise 2: Creating a Custom Ranking Model

**Custom Ranking Models** allow you to change the weights that are used by SharePoint search to rank results. In this exercise you will create a custom ranking model for tasks and use it with the search web parts.

1. Launch **Visual Studio 2010**.
2. In Visual Studio open the XML file [[LAB FILES]]\Ranking Models\Important\_Tasks\_priority.xml.
3. Here is the XML content that you will find inside:

<?xml version=‘1.0’ encoding=‘utf-8’?>

<rankingModel name=‘Important Tasks Priority’

id=‘c978ef2b-300a-444b-af9a-d51261294587’

description = ‘Ranking Model for Tasks’

xmlns=‘http://schemas.microsoft.com/office/2009/rankingModel’>

<queryDependentFeatures>

<queryDependentFeature name=‘Title’

pid=‘2’ weight=‘0’

lengthNormalization=‘10.0000000000’/>

<queryDependentFeature name=‘Status’

pid=‘359’ weight=‘0’

lengthNormalization=‘5.0000000000’/>

<queryDependentFeature name=‘DueDate’

pid=‘400’ weight=‘0’

lengthNormalization=‘2.5000000000’ />

<queryDependentFeature name=‘QueryLogClickedText’

pid=‘100’ weight=‘0’

lengthNormalization=‘20.0000000000’/>

</queryDependentFeatures>

<queryIndependentFeatures>

<queryIndependentFeature name=‘DistanceFromAuthority’

pid=‘96’ default=‘5’ weight=‘0.0000000000’>

<transformInvRational k=‘0.1359244473’/>

</queryIndependentFeature>

<queryIndependentFeature name=‘URLdepth’

pid=‘303’ default=‘3’ weight=‘0.0000000000’>

<transformRational k=‘1.2170868558’/>

</queryIndependentFeature>

<queryIndependentFeature name=‘DocumentPopularity’

pid=‘306’ default=‘0’ weight=‘0.0000000000’>

<transformRational k=‘1.2170868558’/>

</queryIndependentFeature>

<queryIndependentFeature name=‘DocumentUnpopularity’

pid=‘307’ default=‘0’ weight=‘0.0000000000’>

<transformRational k=‘0.7333557072’/>

</queryIndependentFeature>

<categoryFeature name=‘Priority’ pid=‘347’ default=‘0’>

<category name=‘Low’ value=‘3’ weight=‘25.0000000000’/>

<category name=‘Medium’ value=‘2’ weight=‘50.0000000000’/>

<category name=‘High’ value=‘1’ weight=‘100.0000000000’/>

</categoryFeature>

<languageFeature name=‘Language’ pid=‘5’ default=‘1’

weight=‘1.0000000000’/>

</queryIndependentFeatures>

</rankingModel>

1. To work with custom relevance rankings, you use Windows PowerShell. Start Windows PowerShell by selecting **Start » All Programs » Microsoft SharePoint 2010 Products » SharePoint 2010 Management Shell**.
2. Execute the following Windows PowerShell cmdlet to list all of the Managed Properties in SharePoint Search:

Get-SPEnterpriseSearchServiceApplication | Get-SPEnterpriseSearchMetadataManagedProperty

1. Using the information about the **Managed Properties**, carefully replace the property IDs (pid) in your XML file with the values given by the cmdlet.

Most of the PID’s are correct… just verify none of the names and PIDs are mismatched. Some in the XML file will not be listed in the Managed Properties list reported by the PowerShell command. Ignore these.

1. Using Windows PowerShell, you can add a new ranking model by simply copying the XML. Use the following cmdlet to add your custom ranking model to SharePoint -OR-

Get-SPEnterpriseSearchServiceApplication | New-SPEnterpriseSearchRankingModel –RankingModelXML ‘{YOUR XML PASTED AS A STRING}’

To make things easier, you can run a preconfigured **PowerShell** script that will install a new ranking model for you. Execute the following batch file that will call the associated **PowerShell** script: [[LAB FILES]]\StarterFiles\Ranking Models\Add\_Ranking\_Models.bat.

1. After you have added the custom ranking model, run the following cmdlet to list the available ranking models and verify that yours is present.

Get-SPEnterpriseSearchServiceApplication|Get-SPEnterpriseSearchRankingModel

Note: If you need to delete the model, use the following cmdlet.

Remove-SPEnterpriseSearchRankingModel -Identity ‘c978ef2b-300a-444b-af9a-d51261294587’

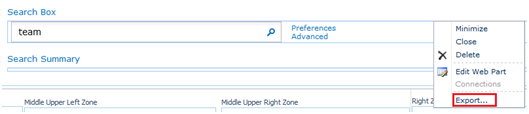
-SearchApplication Get-SPEnterpriseSearchServiceApplication

In this exercise you created a custom ranking model and installed it into SharePoint search.

### Exercise 3: Using a Custom Ranking Model

**Custom Ranking Models** are used in conjunction with the **Core Results Web Part**. The **Core Results** web part has a **DefaultRankingModelID** property that takes the **ID** of the ranking model to use. Unfortunately, this property is not exposed in the property pane. Therefore, you have to add the property into the **\*.webpart** file.

1. Navigate to the ImportantTasksResults.aspx page either directly or by running a query.
2. Select **Site Actions » Edit Page** to place the page in edit mode.
3. On the **Search Box** Web Part, select **Export** from the **Edit** menu.

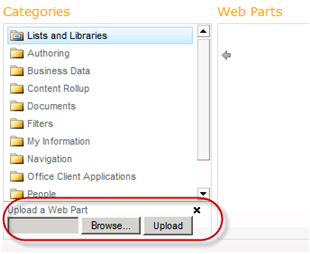


1. Save the file to your desktop.
2. Open the exported file in **Visual Studio 2010** for editing.  
   (Note: you will need to open **Visual Studio 2010** first, then in the **File » Open » File…** and select the Search\_Box.dwp file found on your desktop)
3. Within the file, add a new property as follows to have the **Core Results Web Part** utilize the new ranking model (the GUID is the ID of the ranking model to use):

<DefaultRankingModelID xmlns="urn:schemas-microsoft-com:SearchBoxEx">c978ef2b-300a-444b-af9a-d51261294587</DefaultRankingModelID>

**Note:** Add this just before the **</WebPart>** tag at the end of the file.

1. Save the file and return to the **Search Center** site at **http://intranet.wingtip.com/sites/SearchLab/searchcenter**.
2. With the **ImportantTasksResults.aspx** page still in edit mode, place a check in the Web Part’s top right corner checkbox to select the Web Part. Then select **Insert » Web Part** from the ribbon. If the **Insert** tab is not visible, select the **Top Zone**. This should make the **Insert** tab visible.
3. Select **Upload a Web Part** and upload the modified file:



1. Drop the modified file on the page.

**Note:** Make sure you are using the newly uploaded copy found in **Imported Web Parts** and **not** the original found in the **Search** folder.

1. Remove the original **Search Box** web part from the page (i.e. use the **Search Box [1]** drop down arrow in the web part’s top right corner to select **Delete**).
2. In the newly added Web Part that contains the search box, click the **Web Part** **Menu** drop down arrow in the right upper corner of the web part and pick **Edit Web Part**.
3. In the **Search Box** task pane, expand the **Miscellaneous** section. Change the **Target search results page** URL from ImportantSearchResults.aspx to ImportantTasksResults.aspx and click **OK** at the bottom of the task pane.
4. Now check the page in to save your changes. You can find the **Check in** button on the **Page** tab.
5. Verify that your new ranking model is working by entering **Team** as a search term into the Search box.

In this exercise you implemented the custom ranking model you created in the previous exercise.

### Exercise 4: Extending Search Web Parts

In SharePoint 2010, all of the search web parts may be extended through inheritance. This allows you to interact with the query pipeline both before and after the query is run. In this exercise, you’ll create a web part that makes use of your custom ranking model.

1. Start **Visual Studio 2010** and select **File » New » Project** from the main menu.
2. In the **New Project** dialog, select **Visual** **C# » SharePoint » 2010 » Empty SharePoint Project**. Give it the name **SearchCustomizations**.
3. On the **SharePoint Customization Wizard** dialog box choose **Deploy as a farm solution** to the **http://intranet.wingtip.com/sites/SearchLab** site and click **Finish.**
4. When the new project is created, right click the project and select **Add » New Item**. When prompted, pick the template **Web Part** and give it the name **CoreResultsWithRankingModelID**.
5. Select **Project » Add Reference…** from the Visual Studio main menu.
6. In the **Add Reference** dialog, on the **.NET** tab select the first **Microsoft ® Search component.** Alternatively, if you have trouble finding this, browse to c:\Program Files\Common Files\Microsoft Shared\web server extensions\14\ISAPI and select Microsoft.Office.Server.Search.dll. Then click the **OK** button.
7. Open the Web Part in code view and add the following statements to the top of the code window.

using System;

using System.ComponentModel;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using Microsoft.SharePoint;

using Microsoft.SharePoint.WebControls;

using Microsoft.Office.Server.Search.WebControls;

using Microsoft.Office.Server.Search.Query;

using System.Xml.XPath;

1. Now in the same code view add the following code

[ToolboxItemAttribute(false)]

public class CoreResultsWithRankingModelID : CoreResultsWebPart {

QueryManager qm;

protected override XPathNavigator GetXPathNavigator(   
 string viewPath) {

try {

qm = SharedQueryManager.GetInstance(this.Page).QueryManager;

qm.UserQuery = " scope:Important Tasks";

foreach (LocationList ll in qm) {

foreach (Location l in ll) {

try {

l.RankingModelID = "c978ef2b-300a-444b-af9a-d51261294587";

}  
 catch { }

}

}

}

catch { }

return base.GetXPathNavigator(viewPath);

}

}

1. Build the project and verify that the code compiles. If everything builds, **Deploy** the project.
2. In the **Search Center** at **http://intranet.wingtip.com/sites/SearchLab/searchcenter**, select the **Important Tasks** tab and enter some text and click the **Search** button (i.e. magnifying glass button) to navigate to the ImportantTasksResults.aspxpage.
3. Put this page in edit mode by selecting **Site Actions » Edit Page**.
4. Delete the **Search Core Results** web part from the **Bottom Zone**.
5. Insert the new Web Part by clicking on the **Bottom Zone** on the page and the **Insert** group on the ribbon should appear. Now on the **Insert** ribbon group click the **Web Part** button.
6. Your new Web Part will be in the **Custom** category. Locate the part and then click the **Add** button.
7. In the **CoreResultsWithRankingModelID** web part select the **Web Part Menu** drop down arrow (top right corner of Web Part) and select **Edit Web Part.**
8. On the Edit properties window, expand **Location Properties**
9. Try running a query and verifying that the Web Part is returning results.

In this exercise you created a custom Web Part that inherited and extended the functionality of the Search Core Results Web Part.