## Getting Started

**Lab Time**: 45 minutes

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

**Lab Overview**: In this lab you will begin your work with SharePoint 2010 and become familiar with the Virtual Machine (VM) for this week’s SharePoint 2010 Developer Workshop. You will get experience working with the SharePoint Central Administration site as well as working with a standard team site. This will allow you to experience the new user interface concepts introduced in SharePoint 2010 such as the server-side ribbon and in-place item editing. At the end of this lab you will also get some hands-on experience working with Windows PowerShell and the new SharePoint 2010 Windows PowerShell Snap-in.

### Exercise 1: Creating a Site Collection in SharePoint Central Administration

In this exercise you will create a new SharePoint site collection using the SharePoint Central Administration site.

1. Navigate to the Central Administration site (refer to the lab overview page for instructions on loading Central Administration).
2. Explore the various pages of the Central Administration site by clicking the links in the Quick Launch such as Application Management, System Settings and Monitoring. This should give you an idea of how the Central Administration site breaks up management tasks into different functional areas.
3. Click on the Application Management link in the Quick Launch to navigate to the Application Management page. Inside the Site Collections section of this page, you should see several links for creating and managing site collections. Click on the link titled View all Site Collections which takes you to a page which shows you the site collections in a given Web Application. By clicking on the individual site collections on the left, you should be able to view information about each site collection on the right of the page. You should observe that the VM has a few existing site collections in this Web Application.
4. Now it is time to create a new site collection. Click on the Application Management link in the Quick Launch bar and then click on the link in the Site Collections section titled Create Site Collections. Once you have done this you should be at a page that allows you to enter information for creating a new site collection. Enter the following information so that your screen looks like the screenshot shown below and then click OK to begin the provisioning process to create a new site collection.

**Web Application**: intranet.wingtip.com

**Title**: Wingtip Sales Site

**Description**: My first SharePoint 2010 site collection

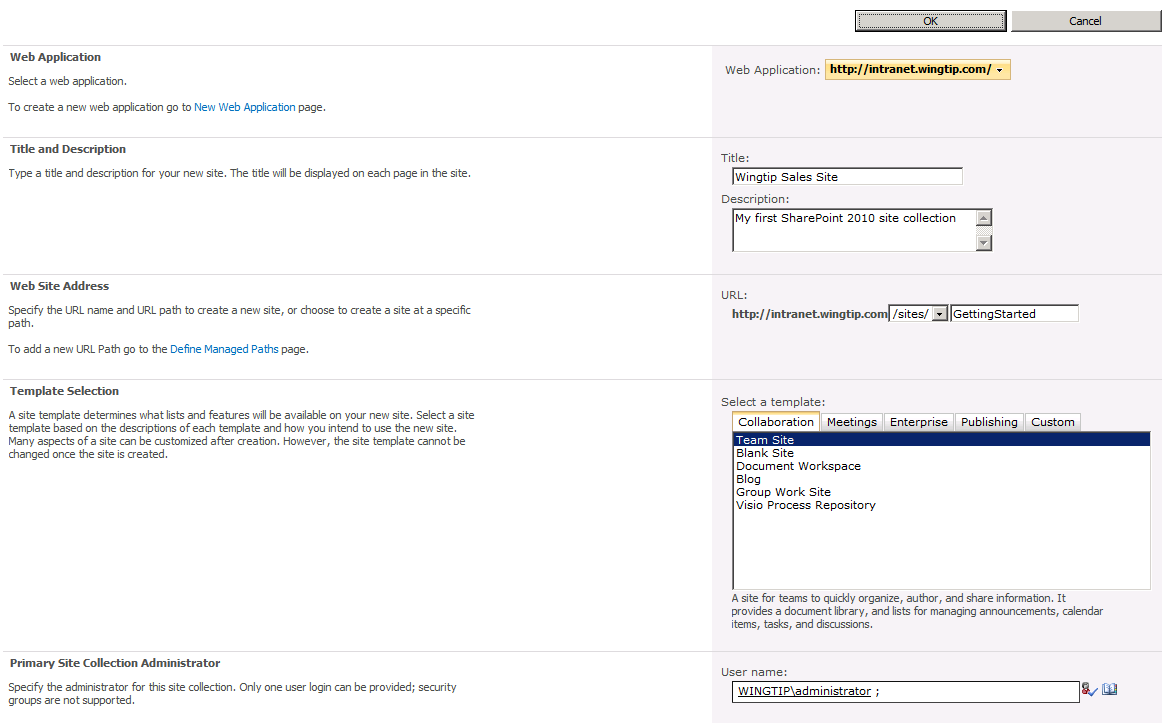
**Url**: http://intranet.wingtip.com/sites/GettingStarted

**Template**: Team site (this template is in the Collaboration tab)

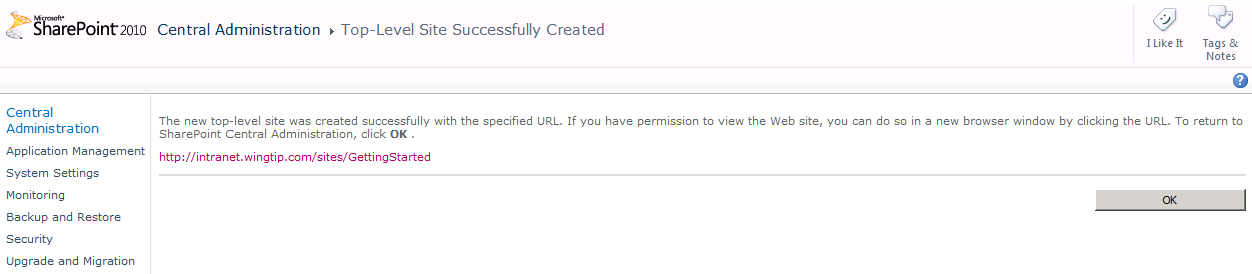
**Primary Site Collection Administrator User Name**: WINGTIP\Administrator

**Secondary Site Collection Administrator User Name**: (leave blank)

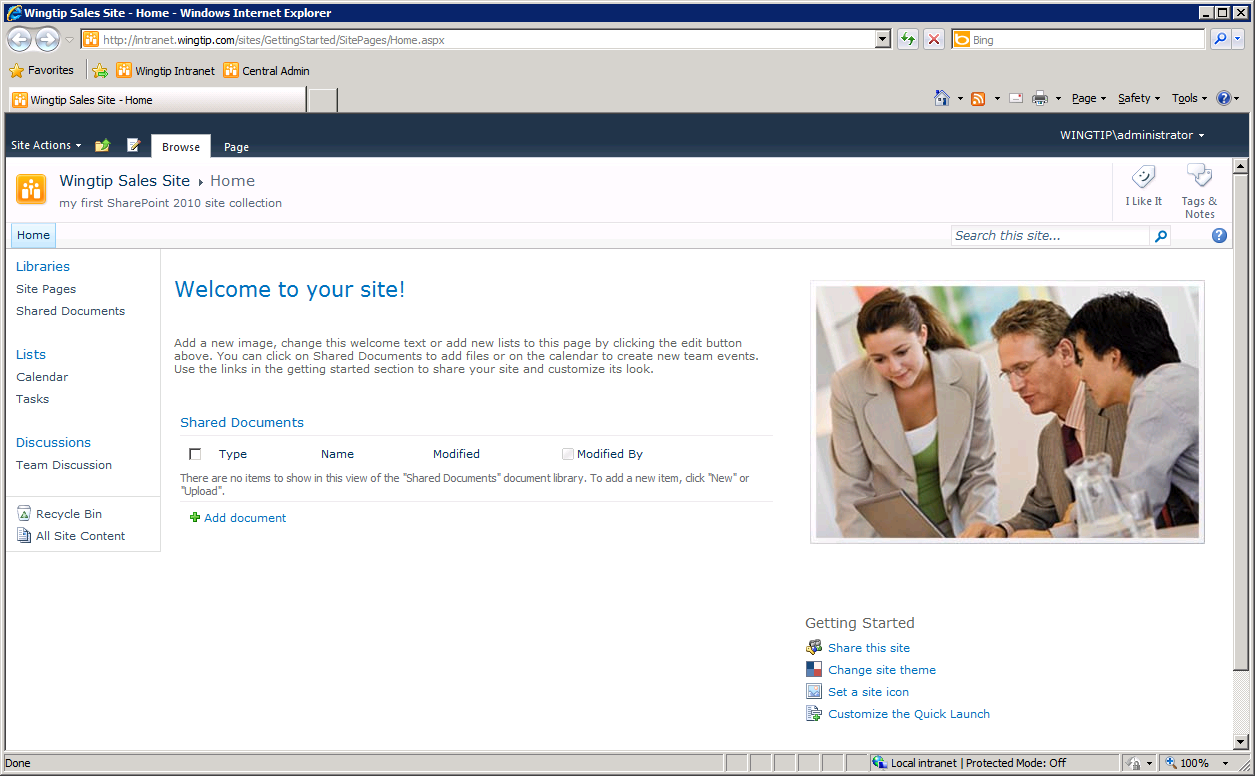
**Quota Template**: No Quota



1. After the site collection provision process is complete, you should a page as the one shown below. Click on the provided link to navigate to new site.



1. When you see the newly-provisioned site collection with a Team site as its top-level site, move on to the next exercise.

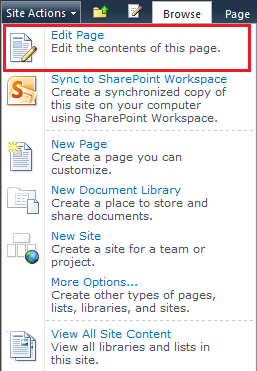


In this exercise you learned how to create a SharePoint site collection using the SharePoint Central Administration site.

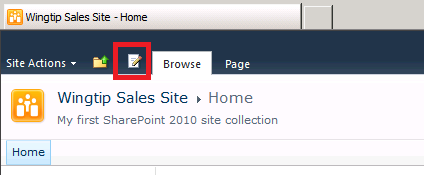
### Exercise 2: Working with the SharePoint 2010 Ribbon and In-place Editing

In this exercise, you will complete work inside the top-level site of the site collection you created in the previous exercise at http://intranet.wingtip.com/sites/GettingStarted. You will begin by adding and viewing items inside of some of the lists that are automatically created as part of a new Team site so you can experience the new paradigm for in-place editing. After that, you will work with a Web Part Page so you can experience how the SharePoint 2010 user interface has changed the way users manage Web Parts.

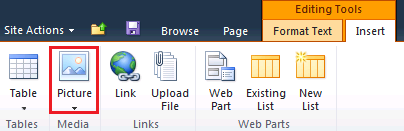
1. At this point you should be at the home page of the site created in the previous exercise at the URL http://intranet.wingtip.com/sites/GettingStarted/SitePages/Home.aspx. Note that the home page is not default.aspx but rather a wiki page named Home.aspx located inside a wiki page library named SitePages.
2. Observe that the page Home.aspx already contains some generic content including a large text block at the top of the page which reads "Welcome". Go into edit mode for the wiki page using the ribbon: Site Actions » Edit Page.



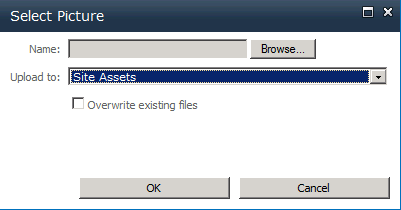
Alternatively, you could use the small edit button just before the **Browse** tab.



1. Once in edit mode, change the entire section of text that begins with "Welcome…" to "Wingtip Sales Site". Use the text formatting buttons in the ribbon to give your new text a font with a size and a color to your liking.
2. While still in edit mode, delete the graphic of the generic stock photo that has been placed on the right-hand side of the page. Now it’s time to add a different photo. Place you cursor on the page at the position where the old photo used to be. Using the ribbon, select the **Edit Tools »** Insert.



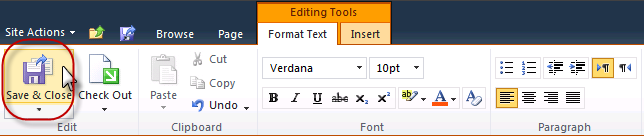
1. Click on the Picture button to add a new picture to the page which should bring up the Select Picture dialog shown below.



1. Note that the Add Picture dialog will allow to select a graphic image which will be automatically uploaded and stored in a special document library named Site Assets. Click the Browse button of the **Add Picture** dialog and navigate to and select the image at the following location:

[[Lab Files]]\StarterFiles\JayHenningsen.jpg

1. Click the Save button in the next dialog.
2. Place the cursor after the picture and press **[ENTER]**.
3. Add the following text right below the picture: **Jay Henningsen – Director of Sales**
4. Now that you have made some cosmetic changes to the home page, save your changes using the ribbon by selecting **Editing Tools » Format Text » Save & Close**. Alternatively, you could use the small Save button next to the Browse tab.



1. Your changes should look as follows:

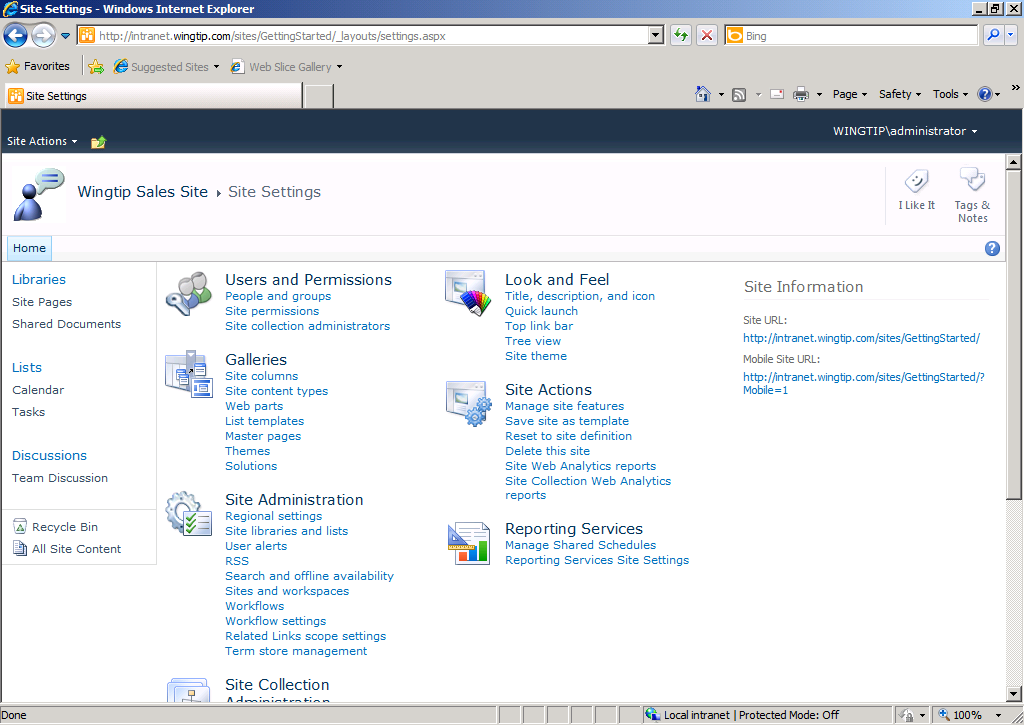


1. Now it's time to explore some other parts of the new site. Drop down the Site Actions menu and see what menu items are available. Click on the menu item titled Site Settings to navigate to Site Settings page. Inspect all the various links to the various administration pages that have been designed for site collections owners and site administrators.
2. Click on the link on the Site Settings page in the Look and Feel section titled Title, description and appearance. This hyperlink takes you to a page where you can see the title and description you entered for the site in the previous exercise. Enter a new value for the Logo URL to assign the site a different site icon. You can use one of the following URLs which point to images that are part of the standard installation for SharePoint 2010.

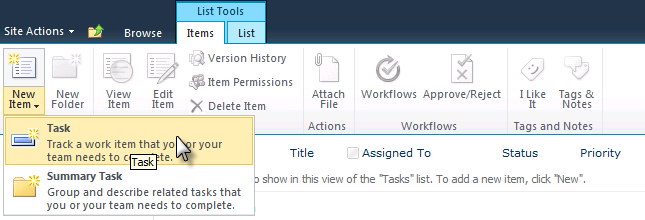
/\_layouts/images/search.png

/\_layouts/images/unknownperson.png

/\_layouts/images/gears\_an.gif



1. Now it is time to add a new item to a list so you can experience working with the new server-side ribbon and in-place editing. Click on the Tasks link in the Lists section of the Quick Launch bar to navigate to the Tasks list which is created as part of every new Team site. You should observe that there are no existing task items in this list. You should also observe the Task list has an associated server-side ribbon with a set of tabs including Browse, Items and List. Click on each of these tabs to see how the ribbon changes to display a contextual set of controls for a given set of tasks.
2. Click on the Items tab of the Task list and then drop down the New Item menu and select Task. This action will display an input form for you to enter the data for a new task item and to click OK to save the new task item back to the content database. The next step will provide you with the data for the new task. At this point it is important for you to observe that the new SharePoint 2010 user interface experience is allowing you to add and edit task items without making the user endure HTTP post backs as was required in the previous version of SharePoint.



1. Create a new task using the following data.

**Title**: Learn how to develop on SharePoint 2010

**Predecessors**: none

**Priority**: (1) High

**Status:** In Progress

**Complete**: 2%

**Assigned To**: Wingtip\Administrator

**Description**: Find out how to develop useful solutions

**Start Date**: 11/1/2009

**Due Date**: 11/4/2009

1. Create a second new task using the following data.

**Title**: Develop Solutions on SharePoint 2010

**Predecessors**: Learn how to develop on SharePoint 2010

**Priority**: (1) High

**Status**: Not Started

**Complete**: 0%

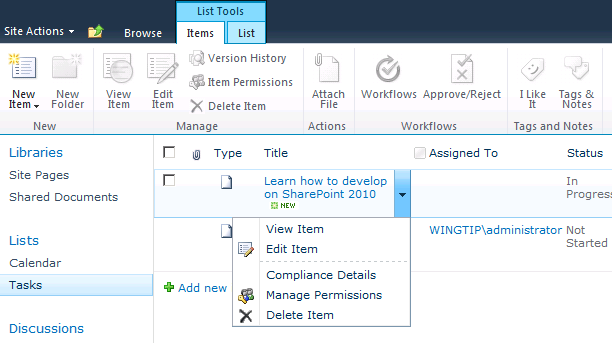
**Assigned To**: Wingtip\Administrator

**Description**: Begin developing solutions on SharePoint 2010

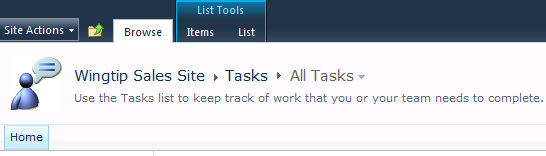
**Start Date**: 11/5/2009

**Due Date**: leave blank

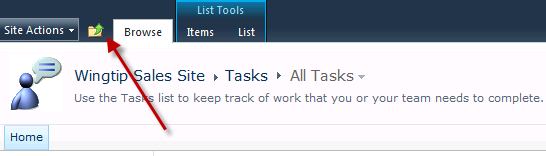
1. Once you have entered the two tasks from the previous steps, experiment with the list view of items in the task list by hovering over individual items. Note that each item highlights as you hover over it and the UI gives you the ability to select it. Use the Edit Control Block (ECB) menu (the standard SharePoint item-hover menu) to go into edit mode for one of the tasks you have created as shown below. Make a simple edit to the task and save it to observe the in-place editing behavior.



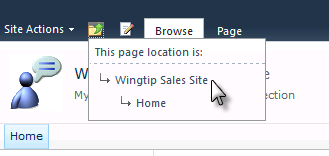
1. Now it is time to add a Web Part to a Web Part Page using the new UI experience for managing Web Parts. Navigate to the home page of the site by first clicking the **Browse** tab to show the top navigation (*the horizontal navigation that starts with “Home”*) bar and then by clicking on the **Home** button on the top navigation bar.



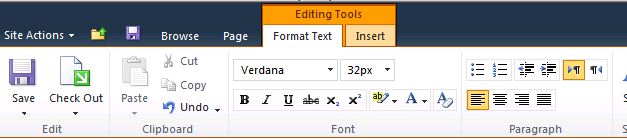
Alternatively, you could use the Navigate Up (*the folder icon next to the Browse tab*) button.



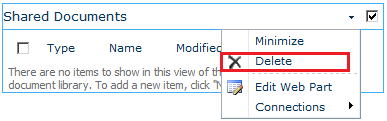
A treeview appears containing the site hierarchy. Select Wingtip Sales Site from the treeview.



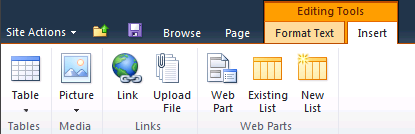
1. Note that the ribbon on the home page has two tabs titled Browse and Page. Click on the Page tab and then click the **Edit** button. Once you place the page in edit mode you should see a new contextual tab with the caption of Editing Tools. Now click on each of these tabs and see how the contextual controls on the ribbon change for each of these tabs.



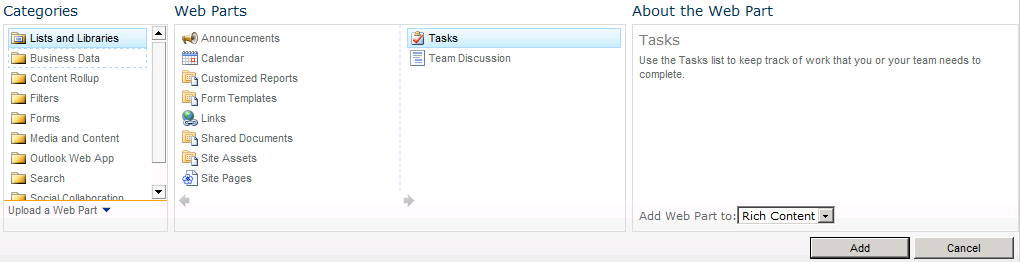
1. Click on the Insert tab that takes you into the mode for adding, modifying and deleting Web Parts from the page.
2. Delete the Shared Documents Web Part in the Left Web Part Zone by clicking on the Delete command inside the Web Part menu.



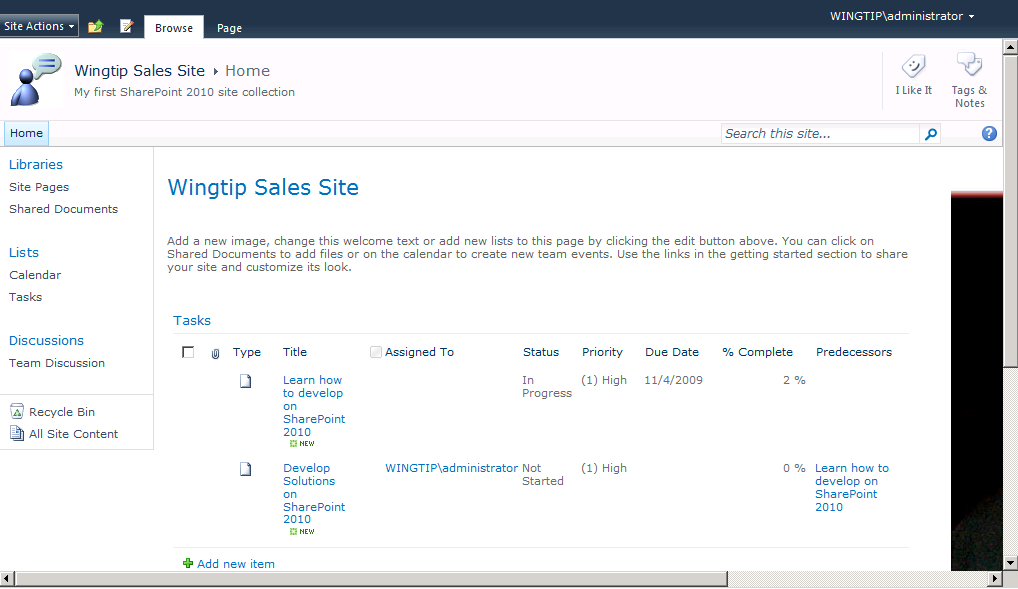
1. Now add a new Web Part to display the tasks inside the Tasks list. Place your cursor below the welcome message and using the ribbon select **Editing Tools » Insert » Web Part** button which is shown in the following screenshot:



1. At this point you should see the new SharePoint 2010 UI for adding new Web Parts to a page. Select Lists and Libraries in the left-hand section and then select Tasks in the right-hand section. Once you have selected the Tasks list, you can observe that the drop-down box on the far right bottom (shown below) has the **Rich Content** zone selected. Click the Add button to add the Web Part instance to the page.



1. Save your changes by selecting the Save button on the ribbon. Now that you have completed this exercise, you should have a page that looks like the one shown below.



In this exercise you got some hands-on experience in working with improvements to the SharePoint user interface such as the new ribbon and adding Web Parts to a page.

### Exercise 3: Working with Windows PowerShell Scripts

In this exercise you will get some exposure to working with Windows PowerShell and the Windows PowerShell add-in created for SharePoint 2010.

1. Launch the Windows PowerShell console from Windows Start menu. You’ll find the PowerShell shortcut at Start » All Programs » Accessories » Windows Powershell » Windows Powershell.

*Make sure to launch the 64 bit version with the shortcut label of Windows Powershell and not the 32-bit version with a shortcut label of Windows Powershell (x86).* Also take note that in this step you should be launching the PowerShell console from standard Windows PowerShell menu item and not SharePoint-specific menu item under *Microsoft SharePoint 2010 Products*.

1. Once you have a command prompt in the Windows PowerShell console windows, type the following command using the Set-Location cmdlet and single parameter with the path [[LAB FILES]] drive and press [Enter]. This is the equivalent of the well-known DOS CD command.

Set-Location [[LAB FILES]]\Powershell\

1. The current folder of the Windows PowerShell console should now be located in a folder that has several Windows PowerShell scripts (\*.ps1 files) that will be used in this lab. Type the following command which uses the Get-ChildItem cmdlet and passes no parameters and press [Enter] to see a listing of all the \*.ps1 files in this folder.

Get-ChildItem

1. Type in the following command to open Notepad and the script named Hello.ps1. Press [Enter] after typing the command to execute it.

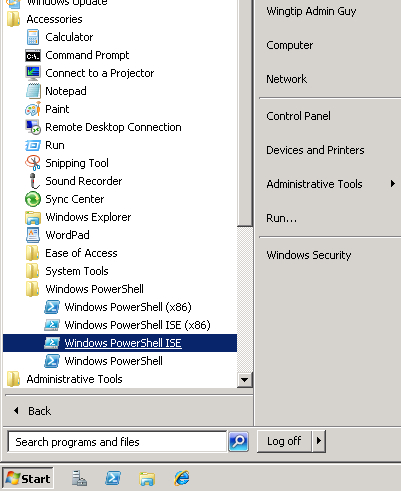
Notepad Hello.ps1

1. After examining the Windows PowerShell code inside Hello.ps1, attempt to run it by typing .\Hello.ps1 and pressing [Enter]. If the Windows PowerShell scripting support on your VM still has the default execution policy of restricted, the script will not run at all. If the execution policy has been changed to unrestricted, the console will prompt you whether to run the script or not. When writing and testing Windows PowerShell scripts, it is easiest to change the execution policy to Bypass so that scripts can freely run without any user prompts. Type the following command and press [Enter] to enable scripting support.

Set-ExecutionPolicy ByPass

1. Now, try to run the script named Hello.ps1 a second time by attempt to run it by typing .\Hello.ps1 and pressing [Enter]. You should now see the script is able to run and output a simple message to the Windows PowerShell console.
2. Now it's time to move to a better script editor. Open the Windows PowerShell ISE from the Windows Start menu. The shortcut can be found at **Start » All Programs » Accessories » Windows PowerShell »** Windows PowerShell ISE as shown below.

ISE = Integrated Scripting Environment



1. Open Hello.ps1 once again in the Windows PowerShell ISE. Drop down the **Debug** menu and you should see it gives you the ability to run and debug the code inside a Windows PowerShell script. Execute Hello.ps1 by pressing [F5] and examine the output.
2. Make a small change to the text message inside Hello.ps1 that is assigned to the variable named $HelloMessage and save your work.

$HelloMessage = "Sample message"

Write-Host "-----------------------------------"

Write-Host "Hello World of Powershell Scripting"

Write-Host "Host name: "$(Get-Item env:\computerName).value

Write-Host $HelloMessage

Write-Host "-----------------------------------"

Run Hello.ps1 one more time in the Windows PowerShell console. You should see your changes and also observe how easy it is to edit and run Windows PowerShell scripts.

1. Now, it's time to practice debugging and single stepping through a Windows PowerShell script. Set a breakpoint by right-clicking a line and selecting Toggle Breakpoint or by pressing [F9] in the first line in Hello.ps1. Now, press [F5] to begin execution. The execution should stop at the breakpoint you set. Now, click [F11] repeatedly to single step through the remaining lines of code inside the script.
2. Open the script named LoadSharePointSnapin.ps1. Examine the Windows PowerShell script code inside. There is call to the Add-PSSnapin cmdlet that loads the snap-in for SharePoint 2010 named Microsoft.SharePoint.Powershell.
3. Now, leave the Windows PowerShell ISE and return the Windows PowerShell console window. Run script by typing .\LoadSharePointSnapin.ps1 and pressing [Enter]. At this point, you can call the cmdlets provided by the Microsoft.SharePoint.PowerShell snap-in.
4. Run the Get-Command cmdlet with the following parameter and press return to see a listing of all the cmdlets included with the Microsoft.SharePoint.PowerShell snap-in.

Get-Command -PSSnapin Microsoft.SharePoint.PowerShell

1. In the last step, there are too many cmdlets in Microsoft.SharePoint.PowerShell to be able to see them all at once in the console windows. Try running the Get-Command cmdlet again but this time requesting only the cmdlets based on the cmdlet verb of Get.

Get-Command -PSSnapin Microsoft.SharePoint.PowerShell -Verb Get

1. Run the Get-Command again with the Verb parameter and pipe the output to a new text file named SP2010Cmdlets.txt.

Get-Command -PSSnapin Microsoft.SharePoint.PowerShell –Verb Get | out-file –filepath “c:\SP2010Cmdlets.txt”

1. Open SP2010Cmdlets.txt with Notepad using the following command to inspect the cmdlets provided by the Microsoft.SharePoint.PowerShell snap-in.

Notepad c:\SP2010Cmdlets.txt

1. Now it is time for you to create a new SharePoint 2010 site collection using the New-SPSite cmdlet. Begin by typing the following command and pressing [Enter] to get help information on the New-SPSite cmdlet which will show you a list of parameters and indicate which ones are mandatory and which ones are optional. Note that optional parameters are enclosed in square brackets.

Get-Help New-SPSite

1. Experiment with the New-SPSite cmdlet to create a new site collection using the parameter values shown below. When you have completed this step you should be able to navigate to the top-level site at http://intranet.wingtip.com/sites/TestSite1.

**Url**: http://intranet.wingtip.com/sites/TestSite1

**OwnerAlias**: Wingtip\Administrator

**Template**: STS#1

**Name**: TestSite1

1. Open the Windows PowerShell script named CreateWingtipSite.ps1 in the **Windows PowerShell ISE**. Review the Windows PowerShell script code inside. Now return to the Windows PowerShell console and run the following command from the Windows PowerShell console to call this script and create a new site.

.\CreateWingtipSite.ps1 TestSite2

1. When you have successfully run the script in the previous step, you should see an output message like the one shown below.

New Wingtip site successfully created

-------------------------------------

Title: Wingtip Site: TestSite2

URL: http://intranet.wingtip.com/sites/TestSite2

ID: *[GUID value of the new SPSite created]*

1. Navigate to the new site at the URL of http://intranet.wingtip.com/sites/TestSite2 and make sure the top-level site behaves as you would expect.

In this exercise you got some hands-on experience in working with some of the provided SharePoint 2010 Windows PowerShell cmdlets as well as working with custom Windows PowerShell scripts.