## Monitoring and Troubleshooting

Lab Time: 30 minutes

Lab Folder: C:\Student\Modules\HealthMonitoring\Lab

Lab Overview: In this lab you will get experience using tools and utilities that allow you to inspect ULS log entries and monitor the health of a SharePoint farm.

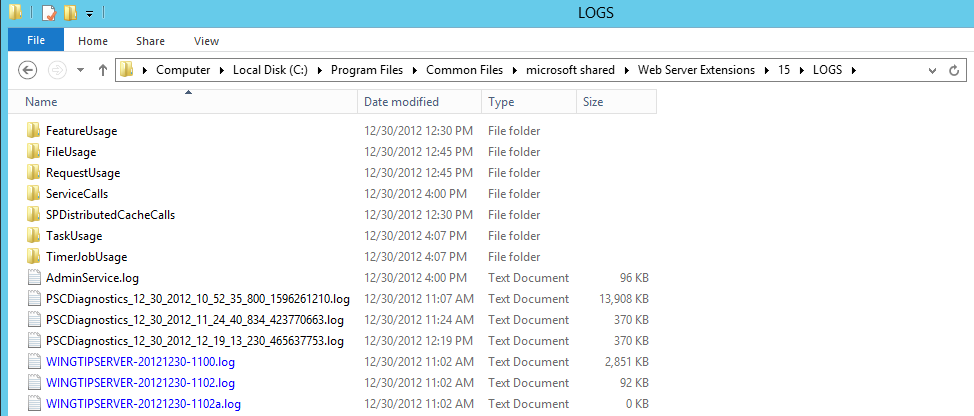
### Exercise 1: Reading ULS Logs

In this exercise you inspect ULS log entries. You will accomplish this two ways. First by inspecting raw ULS files directly using Notepad and then by examining ULS log entries using the **ULS Log Viewer** utility.

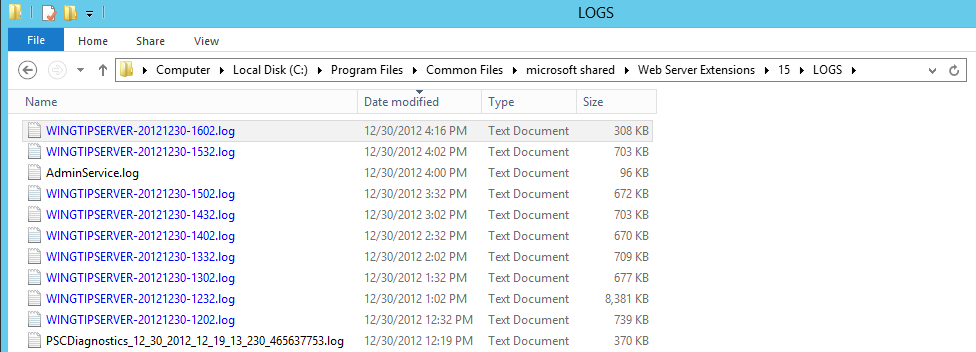
1. Locate and inspect the ULS log files being generated on the **WingtipServer** VM.
   1. Using the Windows Explorer, navigate to the following path.

C:\Program Files\Common Files\microsoft shared\Web Server Extensions\15\LOGS

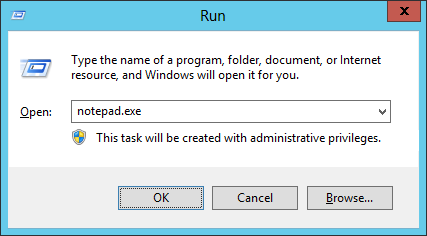
* 1. Inspect the contents of the folder. You will notice this folder contains log files and several child folders containing log files.



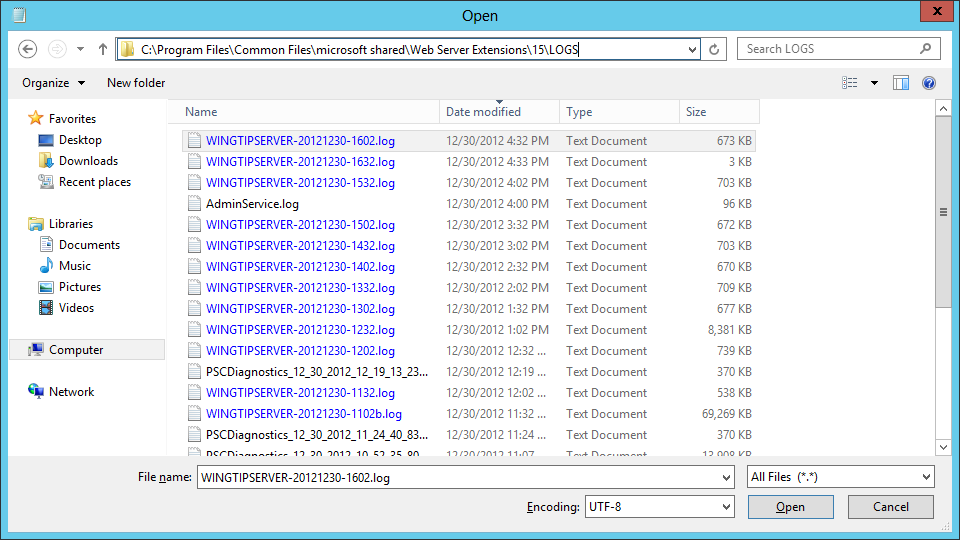
1. Click on the **Date modified** column header to sort the log files so the ones modified most recently are shown at the top. You should be able to observe that a new log file is created every 30 minutes. You should also be able to see that the most-recently modified log file has been modified in the last minute or so. That's because the local SharePoint web server is constantly adding ULS log entries with the default configuration.



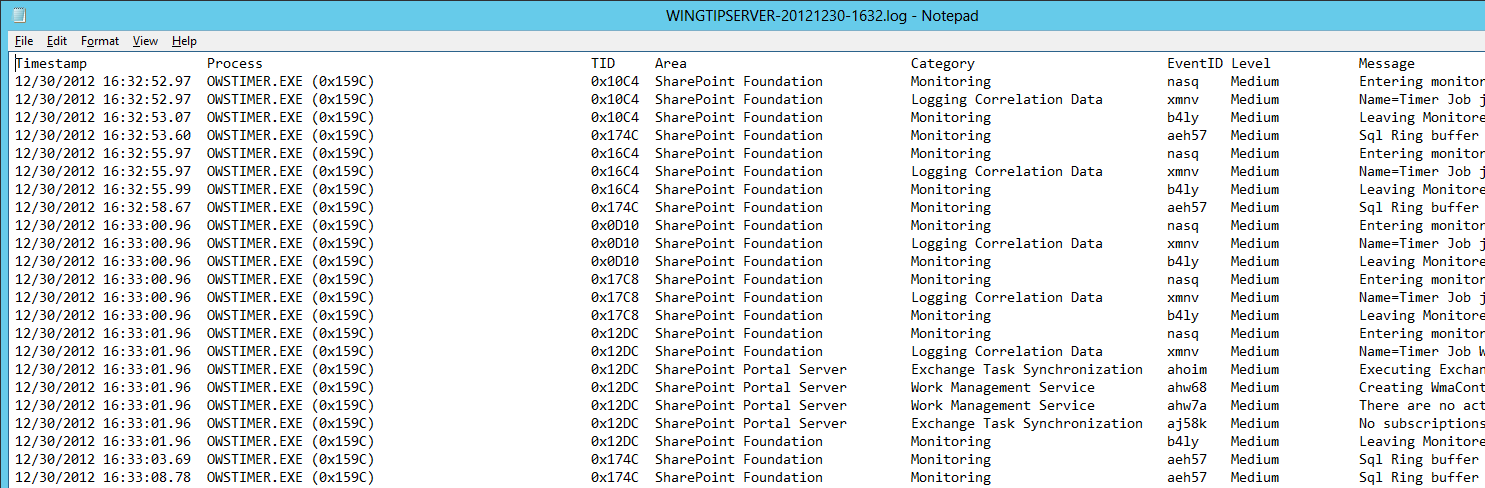
1. Open the most-recently modified log file using the **notepad.exe** utility.
   1. Press the **Windows** key + **R** keyboard combination to display the Windows Run command dialog. Type in **notePad.exe** and then click the **OK** button to start the Windows Notepad utility.



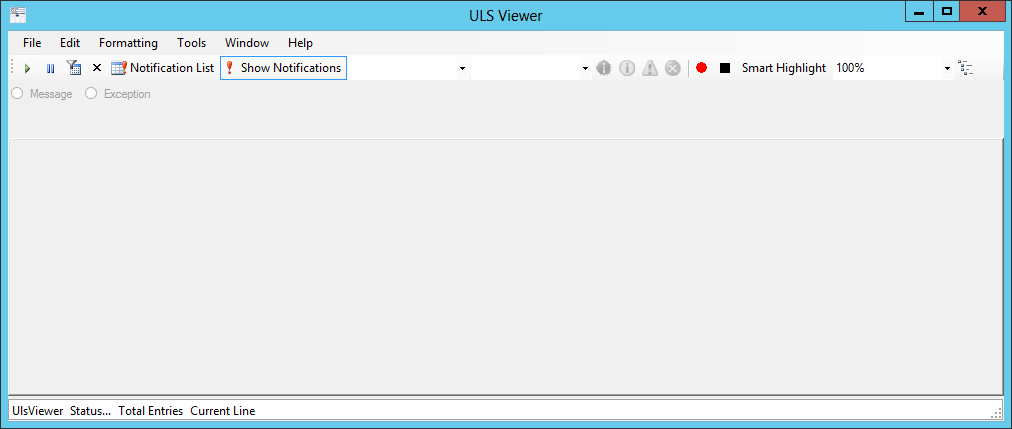
* 1. Once **Notepad** has started, use the **File >> Open** command to open the most-recently modified log file in the root folder for log files at **C:\Program Files\Common Files\microsoft shared\Web Server Extensions\15\LOGS**.



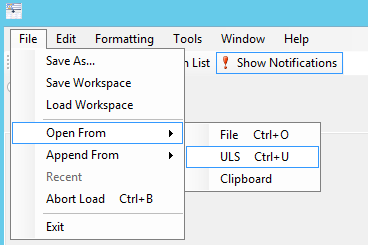
* 1. Inspect the contents of the log file. As you can see, it is pretty hard to read the information in a raw ULS log file.



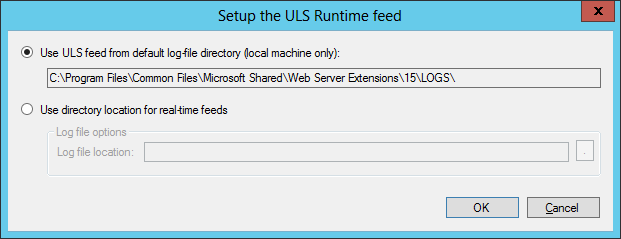
1. Inspect ULS log entries using the **ULS Viewer** utility.
   1. Using the Windows Explorer, navigate to the folder at **C:\Student\ExtraStudentFiles\Resources**.
   2. You should be able to locate the executable file for the **ULS Viewer** utility named **UlsViewer.exe**.
   3. Double-click on **UlsViewer.exe** to launch the **ULS Viewer** utility.



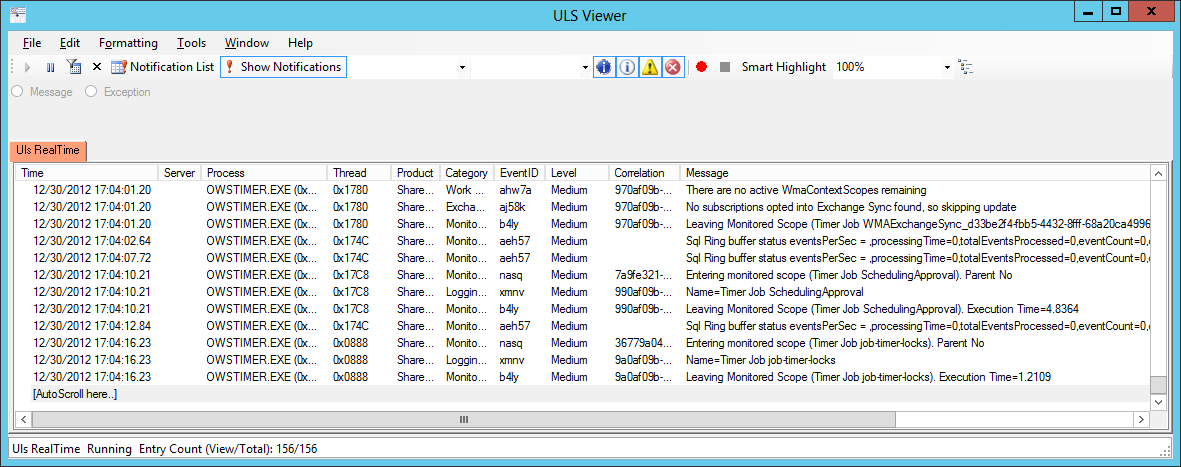
* 1. Open the live ULS log feed on the local web server by selecting the **File >> Open From >> ULS** menu command.



* 1. When you select the **Open From >> ULS** command, the ULS Viewer utility prompts you with the **Setup the ULS Runtime feed** dialog as shown in the following screenshot. Accept the default settings and click **OK**.

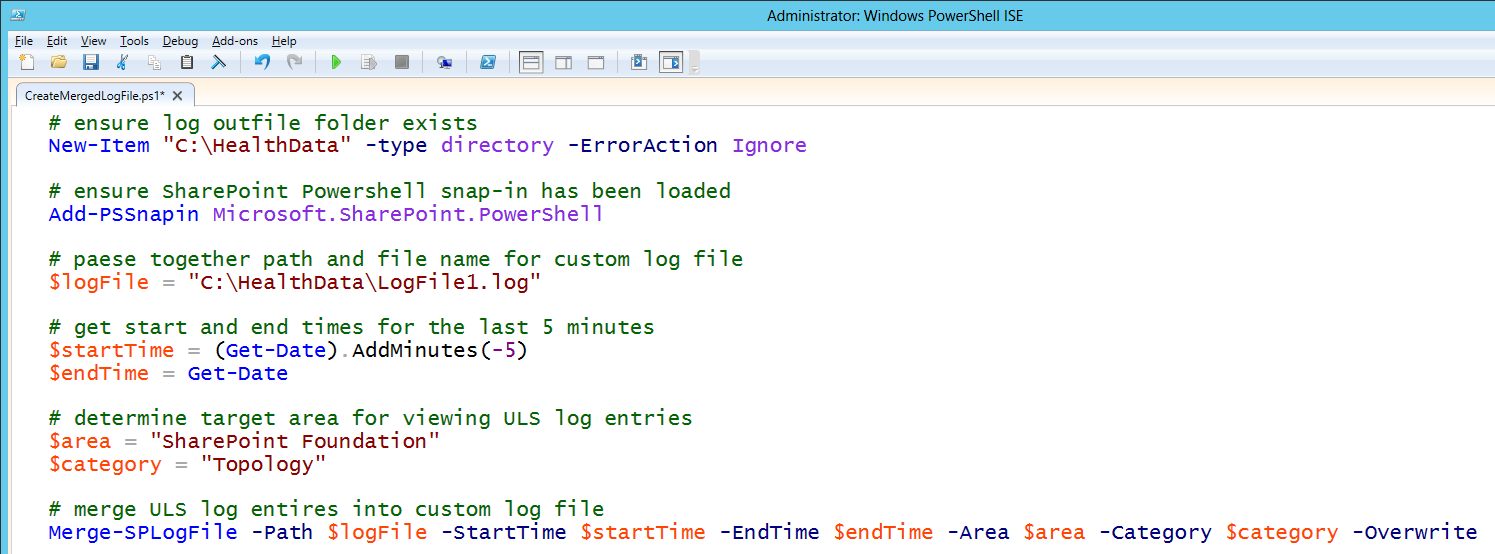


* 1. You should now be able to view log entries as they are added in real time to the ULS log files on the local web server. You should observe that new log entries are added every few seconds.



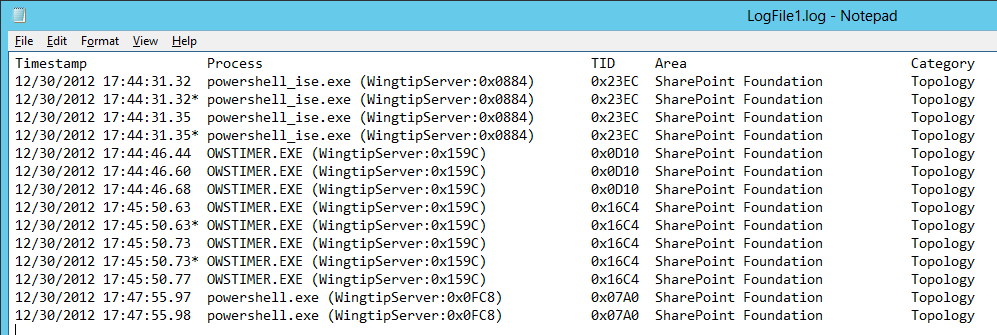
* 1. Leave the ULS Viewer utility open because you will continue using it in the next step.

1. Use the **Merge-SPLogFile** cmdlet to create a merged log file in a Windows PowerShell script.
   1. Using the Windows Explorer, navigate to the folder for this lab at **C:\Student\Modules\HealthMonitoring\Lab**.
   2. Locate the Windows PowerShell script named **CreateMergedLogFile.ps1** and open it in the **Windows PowerShell ISE**.

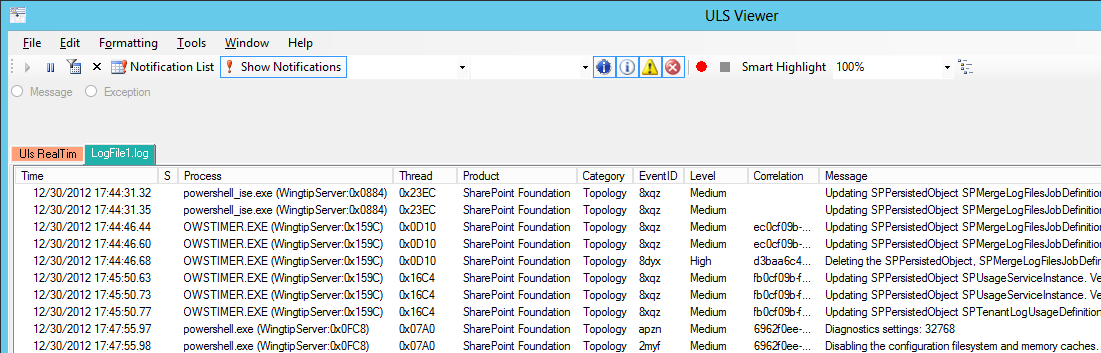


* 1. You are ***not*** required to make any changes to **CreateMergedLogFile.ps1**. However, you shouldtake a moment to examine the script to see how it creates a merged log file which only includes ULS log entries in the area of **SharePoint Foundation** and inside the category of **Topology**.
  2. Execute **CreateMergedLogFile.ps1**. You can do this inside the PowerShell ISE by pressing the **{F5}** key or by right-clicking the script in the Windows Explorer and selecting the **Run with PowerShell** command.
  3. Once the script named **CreateMergedLogFile.ps1** has run, use the Windows Explorer to locate the new log file that was created inside the folder at **C:\HealthData** with the name **LogFile1.log**.

1. Open **LogFile1.log** using Notepad and verify that all the ULS entries inside are in the area of **SharePoint Foundation** and inside the category of **Topology**.



1. Inspect ULS log entries inside **LogFile1.log** using the ULS Viewer utility.
   1. Return to the **ULS Viewer** utility.
   2. Open **LogFile1.log** by selecting the **File >> Open From >> File** menu command and selecting the log file from the **C:\HealthData** folder.
   3. You should be able to see a small number of entries in **LogFile1.log**. Note that the ULS Viewer displays a **Product** column instead of an **Area** column to show the area of each ULS log entry.

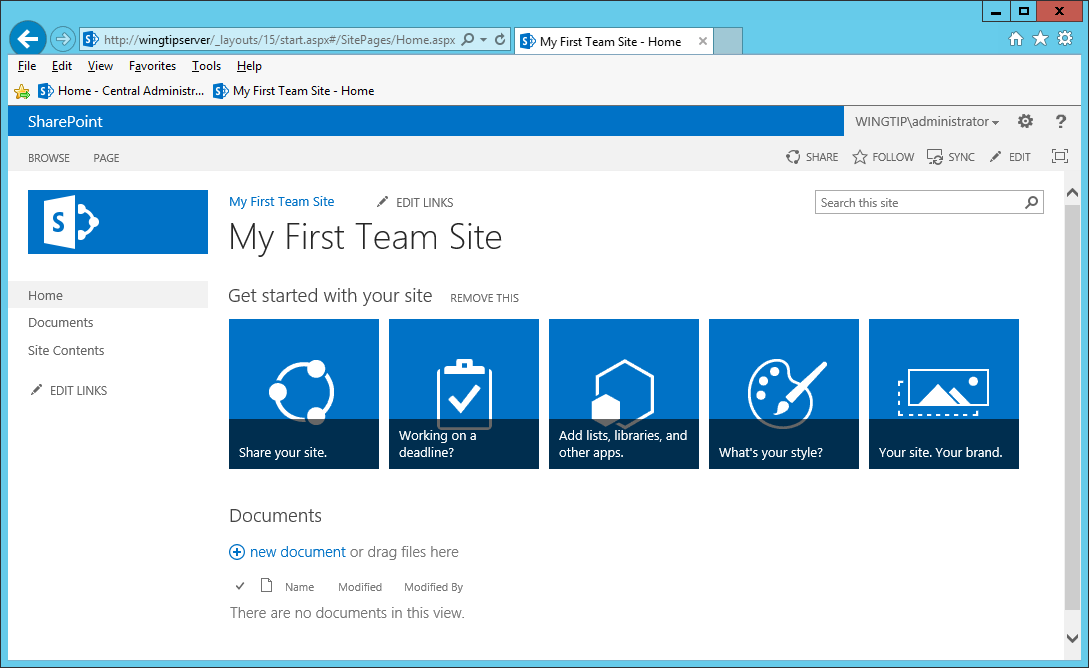


In this exercise you learned two different ways to inspect ULS log entries to monitor the health of the local farm and to troubleshoot errors. You can see that the ULS Viewer utility makes it much easier to inspect ULS log entries.

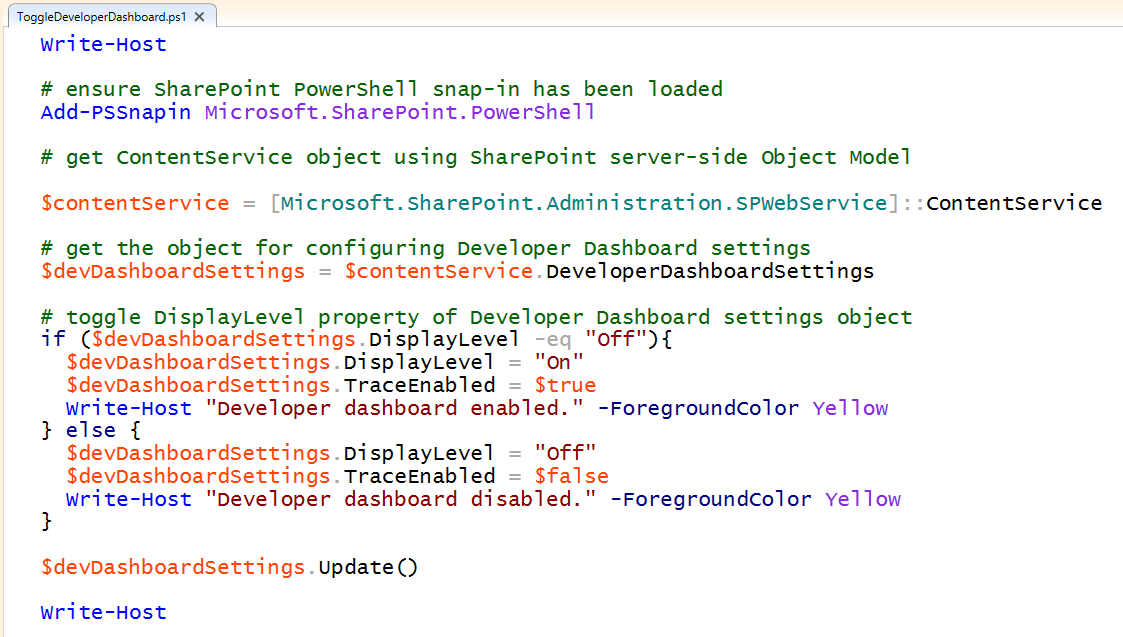
### Exercise 2: Using the SharePoint Developer Dashboard

In this exercise you will enable the Developer Dashboard in order to see information related to the processing of page requests.

1. Using the browser, navigate to the Team site at **http://WingtipServer**. At this point, there is nothing displayed on the pages of this SharePoint site that allows you to launch the Developer Dashboard utility.

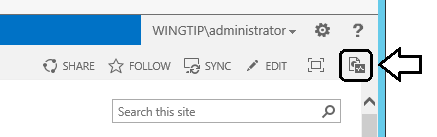


1. Use a PowerShell script to enable the Developer Dashboard in the local farm.
   1. Using the Windows Explorer, navigate to the folder for this lab at **C:\Student\Modules\HealthMonitoring\Lab**.
   2. Locate the PowerShell script named **ToggleDeveloperDashboard.ps1** and open it in the **Windows PowerShell ISE**.

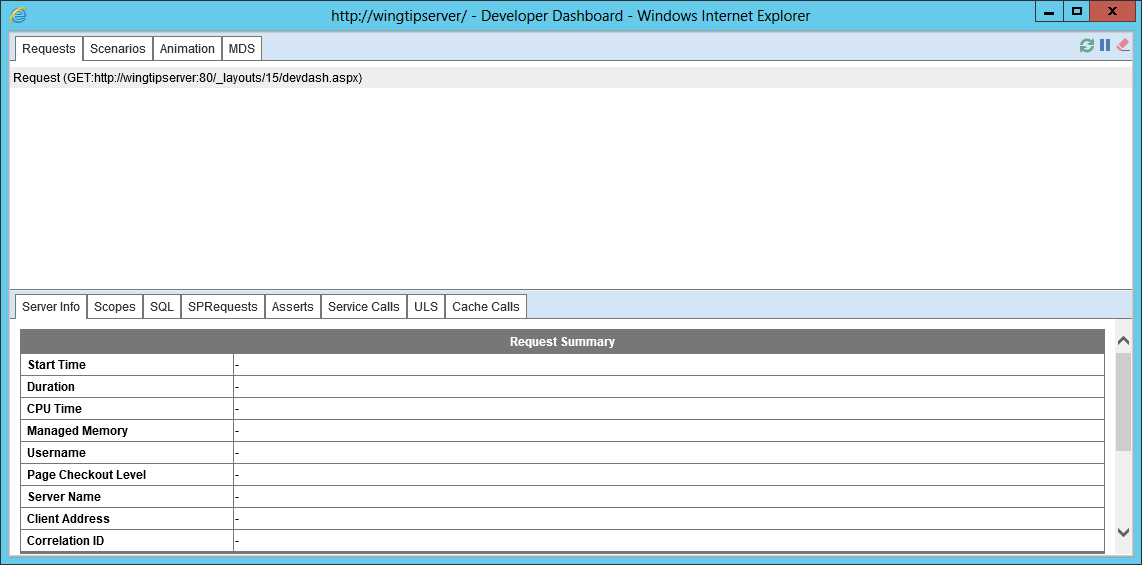


* 1. You are ***not*** required to make any changes to **ToggleDeveloperDashboard.ps1**. However, you shouldtake a moment to examine the script to see how toggles the **DisplayLevel** property and the **TraceEnabled** property of the developer dashboard settings object.
  2. Execute **ToggleDeveloperDashboard.ps1** to set value of the **DisplayLevel** property to **On**.

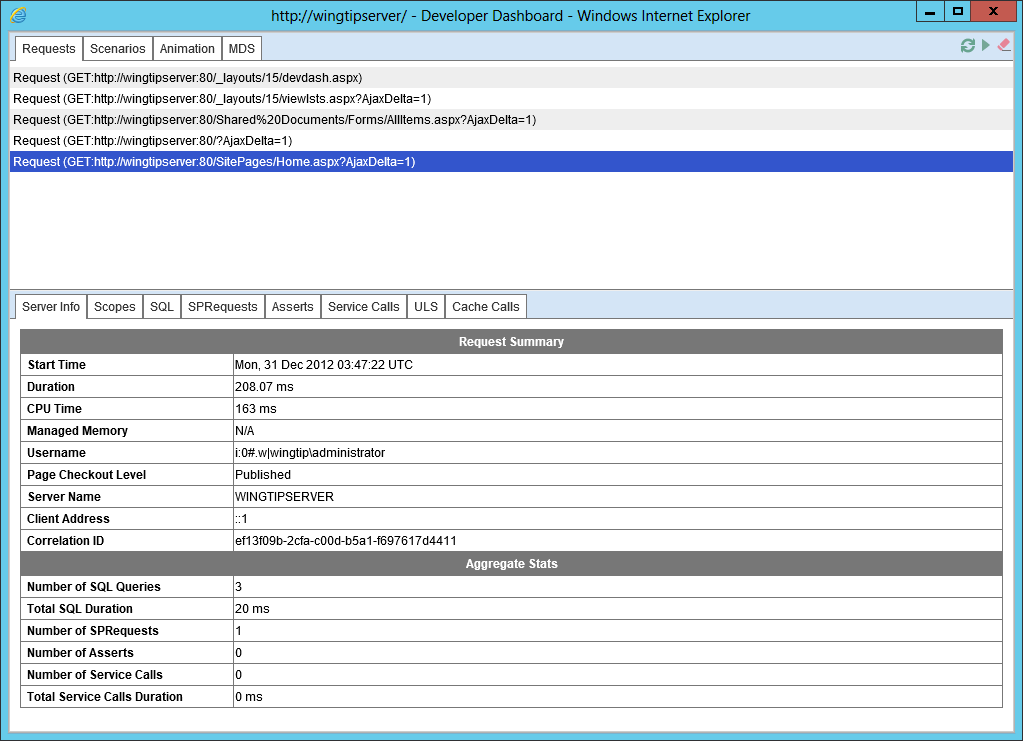
1. Once have run **ToggleDeveloperDashboard.ps1**, return to the browser and refresh the home page for the Team site at **http://WingtipServer**.
2. Look at the top-right corner of the home page. Locate the icon for the Developer Dashboard to the right of the icons for **SHARE**, **FOLLOW** and **SYNC**.



1. Click on the Developer Dashboard icon to launch the Developer Dashboard utility in a separate browser Window.



1. Return to the browser Window which is displaying the home page of the Team site at **http://WingtipServer**. Click on the following links in the Quick Launch navigation menu to generate a few page requests.
   1. **Site Contents**
   2. **Documents**
   3. **Home**
2. Return to the browser window for the Developer Dashboard. After a few seconds, you should be able to see several more page requests in the list shown in the **Requests** tab. Select the last request which targets the sites home page (**Home.aspx**) and examine the information shown below in the **Server Info** tab.



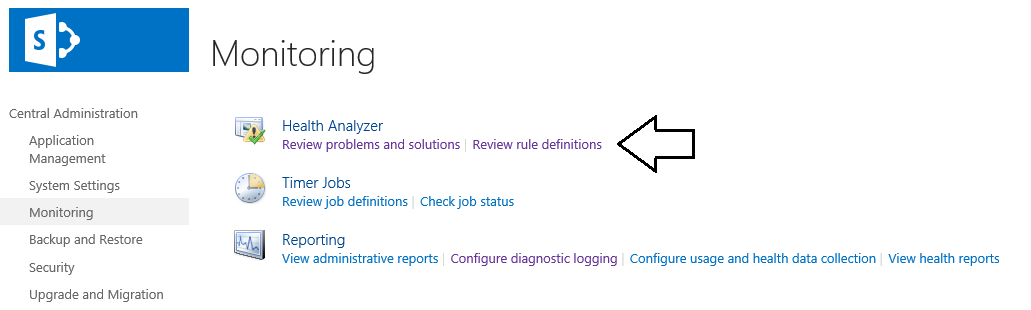
1. Examine the information in the other tabs shown at the bottom of the page.
   1. Scopes
   2. SQL
   3. SPRequests
   4. Assets
   5. Service Calls
   6. ULS (the list of ULS log entries for each page request will likely be empty at this point)
   7. Cache Calls
2. Close the browser window with the Developer Dashboard.
3. Run the **ToggleDeveloperDashboard.ps1** PowerShell script once more to disable the Developer Dashboard. This will have the effect of making the Developer Dashboard icon disappear from pages in a SharePoint site. You can run this script again to add the Developer Dashboard icon back to pages in SharePoint sites if you want to use the Developer Dashboard.

In this lab you learned how to enable and launch the Developer Dashboard to see information related to the processing of pages within a SharePoint site.

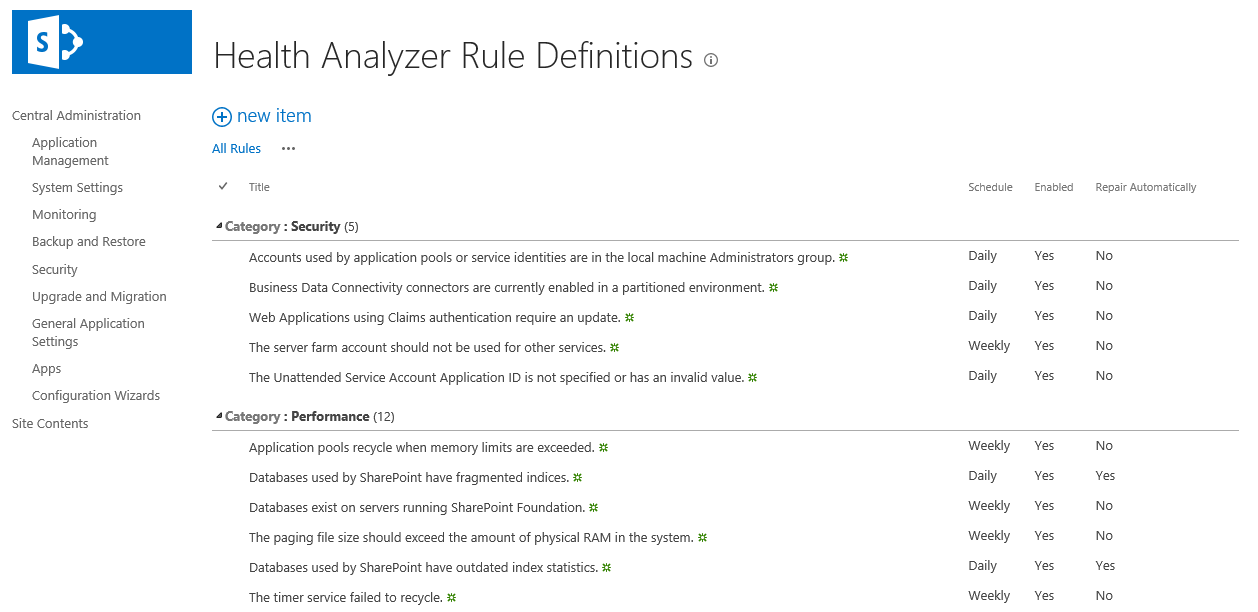
### Exercise 3: Examining the Health Rules used by the Health Analyzer

In this exercise, you will examine the default set of health rules that are periodically run by the Health Analyzer. You will also work through the steps of disabling a rule which can be done in a scenario when one of the default health rules is not relevant for the current deployment of the local SharePoint farm.

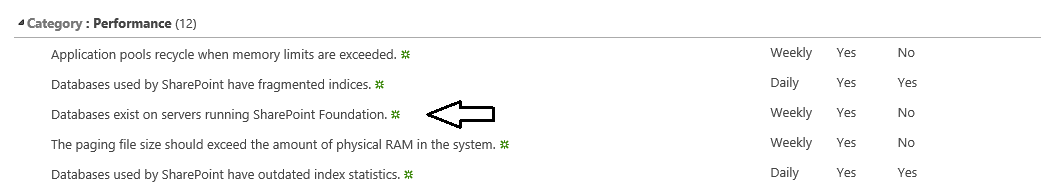
1. Using the browser, navigate to **Central Administration**.
2. In the Quick Launch navigation menu, click the **Monitoring** link to navigate to the **Monitoring** page.
3. On the **Monitoring** page, click the **Review rule definitions** link to navigate to the **Health Analyzer Rule Definitions** page.



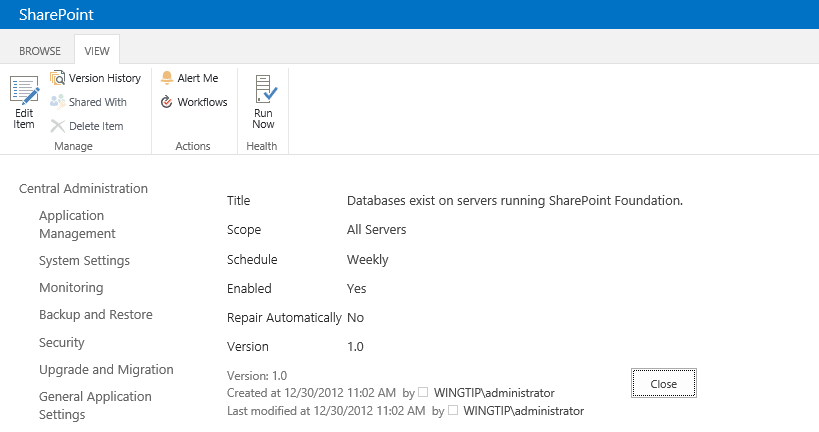
1. Take a moment to review the default set of rules on the **Health Analyzer Rule Definitions** page. Note that the **Health Analyzer Rule Definitions** page using a paging scheme where it only shows 30 rules at a time. Use the navigation control at the bottom of the page to move through all the rules.



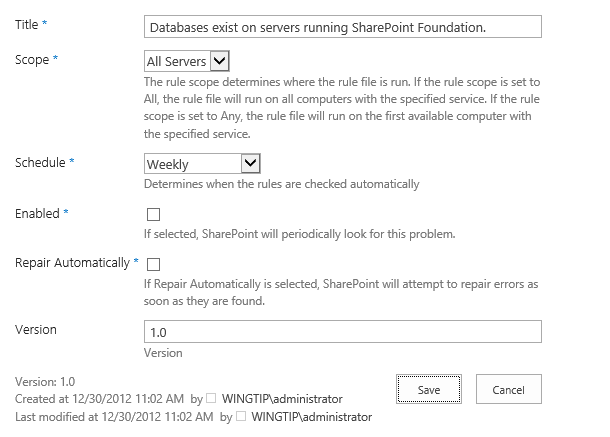
1. Disable the **Databases exist on servers running SharePoint Foundation** rule.
   1. Locate the rule with the title of **Databases exist on servers running SharePoint Foundation** in the **Performance** section. Click on this rule to navigate to the page that allows you to view rule properties.



* 1. On the view item page for the of **Databases exist on servers running SharePoint Foundation** rule, click the **Edit Item** button in the ribbon to navigate to the **Edit Item** page for this rule.

.

* 1. On the edit item page, uncheck the **Enabled** checkbox and click the **Save** button to save your changes.



* 1. At this point, you have disabled this rule and it will not be run within the local SharePoint farm.

1. Close all the browser windows and any other utilities used in this set of lab exercises.

In this exercise, you used Central Administration to review the existing set of health rules in the local SharePoint farm. You also disabled a health rule that is not relevant to a testing environment where you installed SharePoint Server 2013 on the same server computer that is running SQL Server.