## Monitoring a SharePoint 2010 Farm

Lab Time: 45 minutes

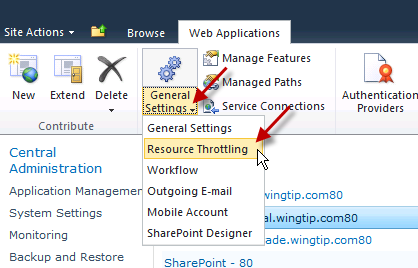
Lab Folder: C:\Student\Labs\15.Monitoring

Lab Overview: In the following lab, you will explore some of the new performance management features new to SharePoint 2010. These features include Resource Throttling (including the HTTP Request Throttling and Large List Throttling), as well as the new Developer Dashboard.

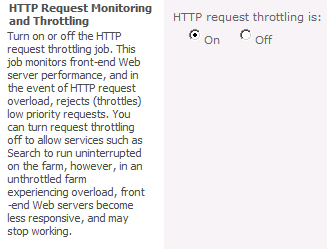
### Exercise 1: Playing with HTTP Request Throttling

HTTP Request Throttling is a new feature introduced in SharePoint 2010 that monitors the CPU usage of the SharePoint server, and can throttle back any low priority HTTP requests during times of very high CPU usage. Any new HTTP requests made to the server will be denied, but the server will still keep users’ work available for when the CPU usage returns to a normal level.

1. Open the Resource Throttling window for the Wingtip Portal Web Application.
   1. Open the **Central Administration** home page.
   2. Under Application Management, click **Manage web applications**.
   3. Click **Wingtip Intranet**.
   4. In the ribbon, click the dropdown underneath the **General Settings** button and select **Resource Throttling**.



* 1. Scroll through the list and look at the various throttling options available. Find the **HTTP Request Monitoring and Throttling** section and ensure that HTTP request throttling is set to **On** (which is its default setting).



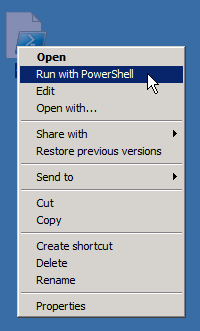
* 1. Click **OK** to close the window.

These throttling settings are set on a per-Web Application basis, giving administrators additional flexibility in these settings. For example, you may want to use different resource throttling settings on a public-facing Internet site than on an internal company Intranet.

1. Make a Windows PowerShell script using Notepad to crank up the CPU usage.
2. Click Start, type notepad and press [Enter].
3. In Notepad, type the following:

while ($true){  
 $x = 98765  
 $y = 56789  
 $z = $x \* $y  
}

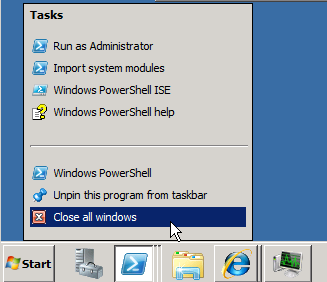
1. This script is basically saying that while the following equation is true, keep doing it. The variable X is equal to 98765, and the variable Y is equal to 56789, and the variable Z is equal to X \* Y. As long as all this is true (and there’s no reason it wouldn’t be) the script will keep running. Because this script is continually multiplying large numbers together, it will drive the CPU usage up very quickly, allowing us to test the HTTP Request Throttling feature.
2. Click **File » Save As**.
3. Click **Desktop** in the left pane to save the file to the Desktop.
4. In the File name: field, type CPU.ps1.
5. In the **Save as type:** dropdown, select **All Files**.
6. Click **Save**.
7. **Close** Notepad.
8. Test the HTTP Request Throttling feature with the PowerShell script.
9. Click the **Show Desktop** icon in the bottom right of the VM window, next to the clock.
10. Click **Start**, type **taskmgr**, and press [Enter] to open Task Manager.
11. In **Task Manager**, switch to the **Performance** tab.
12. **Right-click** on the **CPU** file on the Desktop and select **Run with PowerShell**. Depending on your VM settings, you may need to start multiple instances of this script, so repeat this step as many times as it takes to spike the CPU utilization.



1. When **Task Manager** reports **CPU Usage** at 50-100%, wait about 20 seconds for throttling to start.
2. Switch back to Internet explorer and navigate to http://intranet.wingtip.com.
3. The page will take a long time to load. When it does, quickly refresh the page about 5 times.
4. You will likely be presented with a blank page or 403 error, indicating that the server throttled your request.

Note: Your refreshes may not end up generating a throttle from the server. The HTTP Request Throttling monitor checks the server performance every 5 seconds, and if it receives more than 3 HTTP Get requests in that time the requests will be throttled. If it receives less than 3 requests in 5 seconds during throttling, the throttling will be disabled. It is difficult to fully test this in a single-user VM environment.

* 1. Right-click the PowerShell logo in the Taskbar and select Close all windows to stop the script and return the CPU to normal utilization.



1. Turn HTTP Request Throttling Off and try loading the page.
2. Head back over to Central Administration and click **Manage web applications** from the home page.
3. Select **Wingtip Intranet** from the list of Web Applications.
4. In the ribbon, click the dropdown under **General Settings** and select **Resource Throttling**.
5. Scroll down to the HTTP Request Monitoring and Throttling section and set HTTP request throttling to **Off**.
6. Click **OK**.
7. Fire up as many instances of the CPU script as it takes to reach 100% CPU utilization.
8. When CPU utilization is at 100%, wait 20 seconds then navigate to http://intranet.wingitp.com.
9. The page will likely time out before it loads or you will receive a 403 error. The server load is too high to display the page.

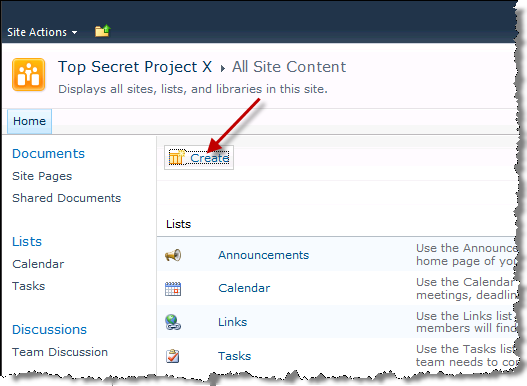
You may be wondering what the difference is between having the Request Throttling on or off, since the response from the server is similar. The idea with having Request Throttling on is that any work being done on the server should still be available after the server is operating under normal load. There isn’t a guarantee that this will be the case with Request Throttling off.

HOWEVER: A point to consider when using Request Throttling is that there is the potential for Search to max out the CPU cycles during a crawl, triggering the Throttling. If this were to happen, search would be presented with nothing but errors during its crawl.

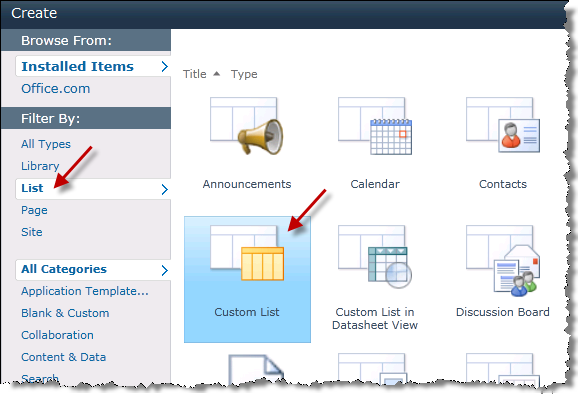
### Exercise 2: Fun with Large Lists

SharePoint 2010’s list capacity has been increased enormously – up to 50 million items in a single list. But the larger a list gets in SharePoint, the longer it takes to render which can end up affecting server performance if too many people try to access the same huge list. Luckily, a mechanism is in place to keep users from bringing down the server. This feature is called Large List Throttling, and much like the HTTP Request Throttling feature throttles requests for pages, its job is to throttle requests for views of lists that are set to display more items than are allowed by the server administrator.

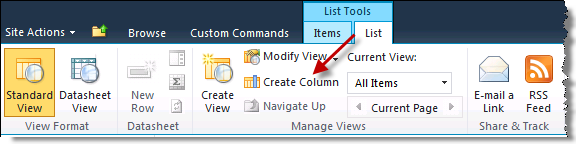
1. Set up the list view threshold for the Portal site.
2. Open **Central Administration**.
3. On the home page, click **Manage web applications**.
4. Select the **Wingtip Intranet** Web Application.
5. As before, open the **Resource Throttling** window by clicking the **General Settings** dropdown and selecting **Resource Throttling**.
6. Change the **List View Threshold** to **2000** (this is the lowest it can be set at).
7. Click **OK**.
8. Build a list on the Product Management site.
9. Navigate to http://intranet.wingtip.com.
10. In the **Site Actions** menu, click **View All Site Content**.
11. Click **Create**.



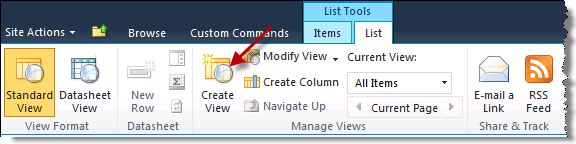
1. Choose List in the left pane, then click **Custom List**.



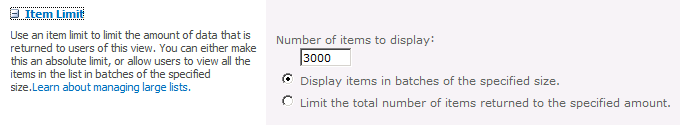
1. In the Name field to the right, type **PartsList**.
2. Click **Create**.
3. Add Columns to the **PartsList**.
4. When the list opens, click **Create Column** in the ribbon.



1. For the **Column Name**, type **PartNumber**.
2. Select **Number (1, 1.0, 100)** as the type of column.
3. Scroll down and click **OK**.
4. In the ribbon, click **Create Column**.
5. For the Column Name, type **Model**.
6. Select **Single Line of Text** as the type of column.
7. Scroll down and click **OK**.
8. Make a view for the **PartsList**. You will be populating this list with 6000 items and it would be a pain to page through 30 items at a time.
9. In the ribbon, click **Create View**.



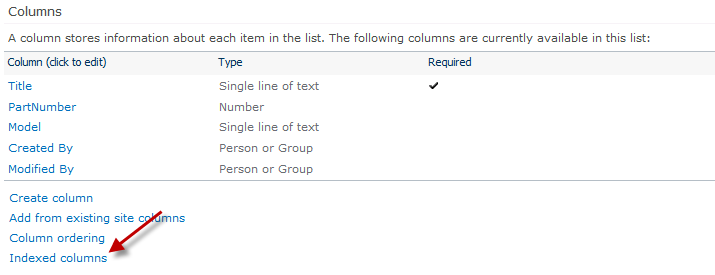
1. Click **Standard View**.
2. For the **View Name**, type **BigView**.
3. Scroll down and expand the **Item Limit** section.
4. Set the Number of items to display: to **3000**.



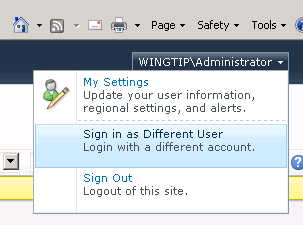
1. Scroll down and click **OK**.
2. Use a script to quickly populate this list with 6000 items.
3. Navigate to [[Lab Files]]\Starter Files.
4. Right-click on the CreateListItems.ps1 file and select **Run with PowerShell**.
5. A PowerShell window will open. The list will automatically be populated with 6000 items. The script takes around 2-3 minutes to run and will close automatically when it is finished.
6. Return to the **PartsList** and **refresh** the page and click **Retry** when prompted. Notice that 3000 items are returned for the view, but performance is very slow.
7. Because this list has so many items, make the **Model** column an indexed column.
8. Click the **List** tab in the ribbon.
9. Click the **List Settings** button.
10. On the list settings page, notice that there is a warning about the number of items in the list.



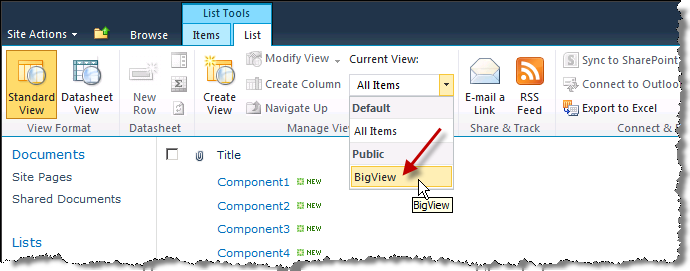
1. Scroll down the page and under Columns, click the link **Indexed columns**.



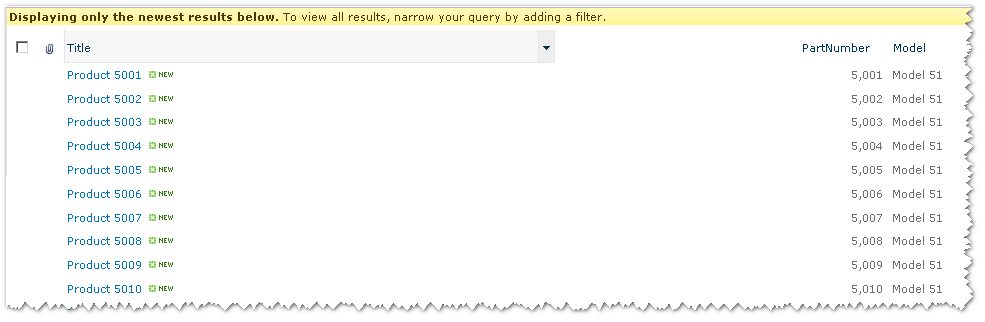
1. Click **Create a new index**.
2. On the **Primary column for this index:** dropdown, select **Model**.
3. Click **Create**.
4. Grant Ken Sanchez access to the site as a reader.
5. From the http://intranet.wingtip.com site, select **Site Actions » Site Permissions**.
6. Click the **Grant Permissions** button in the **Edit** tab within the ribbon.
7. In the **Select Users** box, enter **Ken Sanchez**.
8. In the **Grant Permissions** selector, assign **Ken** to the **Visitors** group for the site that has **Read** access.
9. Test the Large List Throttling as Ken Sanchez.
10. Navigate back to the **PartsList** using the Quick Launch bar.
11. Using the user control in the upper right corner of the browser, signin as **Ken Sanchez** (WINGTIP\KenS)



1. Click the **List** tab in the ribbon.
2. Click the **All Items** dropdown and change the view to **BigView**.



1. Ken only sees the newest 1000 items in the list instead of all 3000 items in the list, even though the **BigView** view is set to show 3000 items at once.



1. Make a new view for Ken so she can see all items in the list.
2. In the ribbon, click **Create View**.
3. Click **Standard View**.
4. For the View Name, type **Ken’s View** (since Ken isn’t a contributor on the site she can’t make public views).
5. Scroll down and **expand** the **Item View** section and type **2000**.
6. Click **OK**.
7. Now Ken’s new view is set to show 2000 items, which meets the 2000 item limit set in Central Administration.

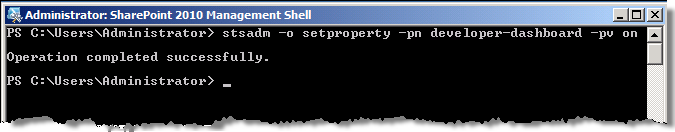
So why is the administrator able to log in and see the BigView without being throttled? The limit set in Central Administration can be overridden by the local administrator account.

### Exercise 3: Play with the Developer Dashboard

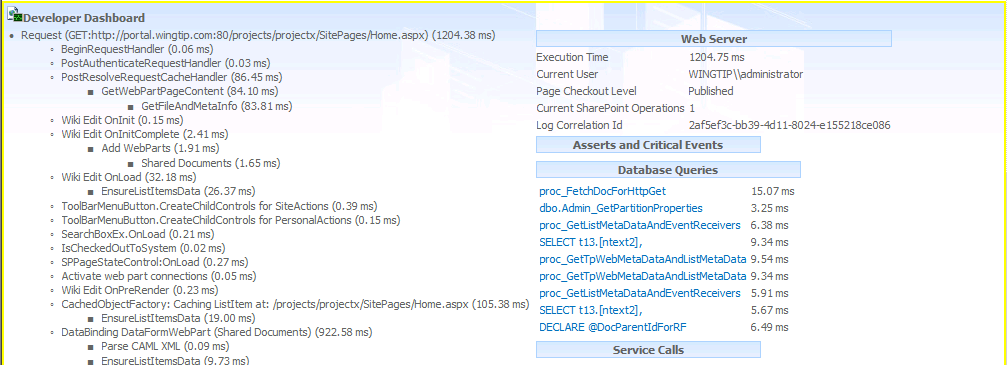
The Developer Dashboard is a new performance tool that administrators and developers can use to track page performance.

1. Turn the Developer Dashboard on.
2. Open the **SharePoint 2010 Management Shell**.
   1. Type the following cmdlet and press [Enter].

stsadm -o setproperty -pn developer-dashboard -pv on



* 1. When you get the Operation completed successfully message, switch back to the browser and **refresh** the page (it may take a bit for the command to complete since this is a farm-wide setting you are enabling here).
  2. Scroll down to see the Developer Dashboard.



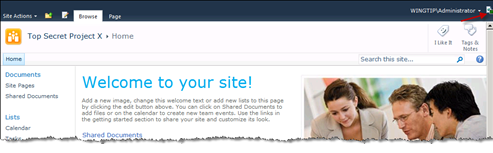
* 1. In the Quick Launch click **PartsList**.
  2. Scroll down to see the Developer Dashboard and note the **Execution Time** under the Web Server section on the right.
  3. Switch the view to **BigView**.
  4. When the page loads, scroll down to see the Developer Dashboard and the **Execution Time** for **BigView**.

This is great and all, but what if you don’t want the Developer Dashboard loading with every single page?

1. Turn the Developer Dashboard to on-demand mode.
2. Switch to the **SharePoint 2010 Management Shell**.
3. Type the following command and press [Enter].

stsadm -o setproperty -pn developer-dashboard -pv ondemand

* 1. Switch back to the browser and open http://intranet.wingtip.com.
  2. Look in the upper right corner of the screen for the Developer Dashboard icon.



* 1. Click the icon to refresh the page and load the Dashboard.
  2. Scroll down to see the Developer Dashboard.
  3. Click the icon again to reload the page without the Developer Dashboard.

Note that the Developer Dashboard is only available to users with at least Contribute permission.

1. Turn the Developer Dashboard off.
2. Switch back to the **SharePoint 2010 Management Shell**.
   1. Type the following command and press [Enter].

stsadm -o setproperty -pn developer-dashboard -pv off

* 1. Switch back to the browser and refresh the page. Notice that the Developer Dashboard icon is gone.