

How to Configure a Stress Test Project for Microsoft Office SharePoint Server 2007 using Visual Studio Team Suite 2008.



By Steve Smith, MVP SharePoint Server, MCT
And Penny Coventry, MVP SharePoint Server, MCT

Combined Knowledge
www.combined-knowledge.com
steve@combined-knowledge.com



This 'How to' guide will take you through configuring a Stress Test Project for your Microsoft Office SharePoint Server 2007 farm using Visual Studio Team Suite 2008.

Scenario

The scenario that this whitepaper addresses is where an organization is planning to use SharePoint Server 2007 as well as those who currently use SharePoint Server 2007. Organizations can use this whitepaper to test whether their planned or current hardware will support the envisaged user scenarios.

Note It is not the intention of this whitepaper to show you real world benchmark results the intention is to show you how to get started with configuring and managing your stress tests.

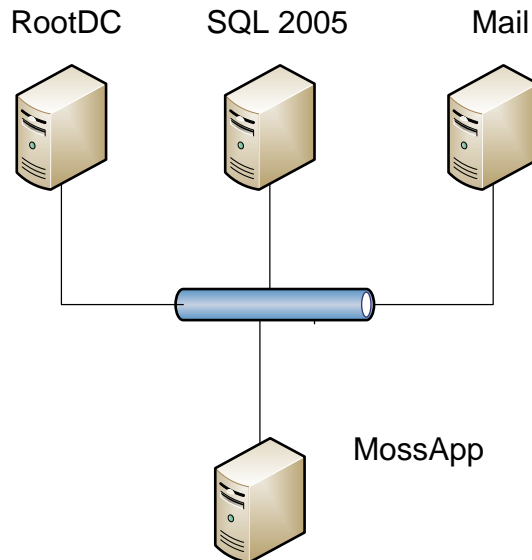
Configuration

The server topology that this guide uses as its starting point is as follows:

Server Name	Role	Products Installed
ROOTDC	Domain Controller DNS Server Mail Server	Windows Server 2003 SP1 (32-bit) Exchange 2003 Enterprise Visual Studio 2008 Team Suite
SQL2005	Database server	Windows Server 2003 SP1 (32-bit) SQL 2005 SP1
MOSSAPP	SharePoint Server (Application and Web Server with Index and Query Search service and Excel services started).	Windows Server 2003 SP1 (32-bit) SharePoint Server 2007 Enterprise SP1 (32-bit)
XPCLIENT	Client	Windows XP Pro (32-bit) with Office 2007 Enterprise (SP1)

This paper uses the internal domain name configured for the Active Directory is trainsbydave.com. You will need to replace trainsbydave.com with your own testing domain name.

The diagram below shows the environment at the start of the process:



The next step is to ensure that the Web applications are distributed according to the following table:

Web Application	Initial Location	Description
My Site	MOSSAPP	Hosted the My Site site collections for this environment.
Corp	MOSSAPP	Top-level site created from the Collaboration portal site template that can be accessed using corp.trainsbydave.com
SharePoint Central Administration v3	MOSSAPP	Site Collection created when SharePoint Server first installed.
SSP1	MOSSAPP	Web application used to host the SSP Administration site collection.

About Visual Studio Team System 2008

The Visual Studio Team System 2008 (VSTS) testing capabilities replace the Application Centre Test (ACT) Stress tool that was available in Visual Studio 2005 (VS 2005).

Note: You can also use Visual Studio Test Edition to run the same stress tests. (Thanks [Spencer](#))

If you wish to produce network traffic as part of your stress test scenarios, unlike VS 2005, VSTS 2008 does not require client software to be installed on other machines. You still need desktop computers to generate traffic to your SharePoint Servers that are hosting your SharePoint sites, however, VSTS provides agents to those designated desktops computers, which specify the actions that the desktop computers will complete to produce the network traffic.

Ok so let's get started

Step 1 – Prepare Visual Studio Teams Suite 2008

The following procedure assumes that the VSTS edition is installed with the Load Test option selected. You need to install the VSTS on a server to control the web tests. It should be installed on a server other than the SharePoint servers. In the environment created for this White Paper, VSTS was installed on the domain controller.

We have also created a share on the domain controller, called VSTS Solutions, where we are keeping our VSTS test files, and so other IT Pros can access them once they are created. By default, VSTS does not trust any location other than local drives, therefore if you wish to store your test solutions on a file share you will need to complete the following procedures to modify the policy affecting a file share. This gives a specific file share Full Trust permission using Mscorcfg.msc. or Caspol.exe You must be an administrator on the computer to make this change. In our example we will use Mscorcfg.msc

How to Config the Trusted Location on ROOTDC:

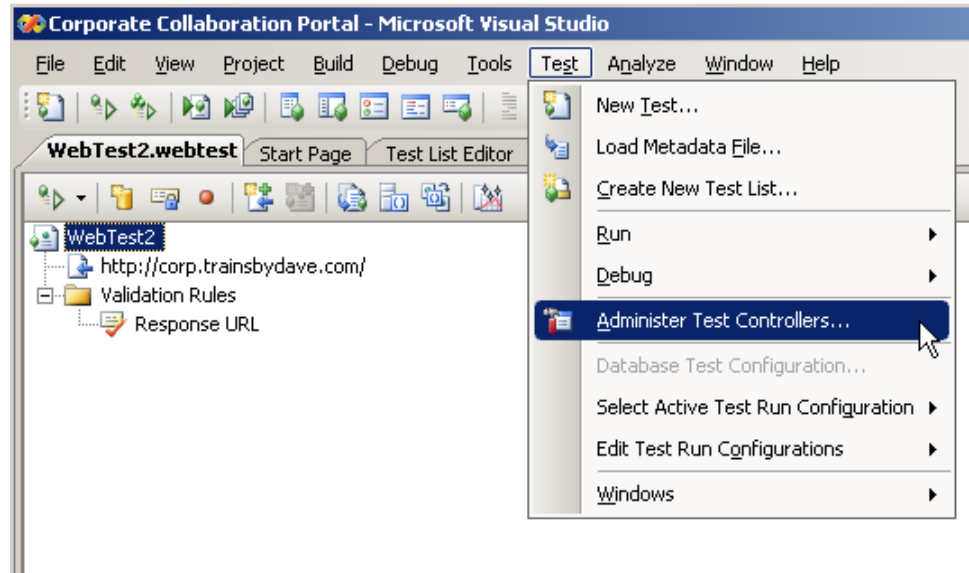
1. Start Mscorcfg.msc. or from the **Start Menu**, click **Admin Tools, Microsoft .NET Framework x.x Configuration**.
2. Expand the **Runtime Security Policy** node, the **Machine** node, the **Code Groups** node, and the **All Code node**.
3. In the right pane, click **Add a Child Code Group**.
4. Choose **Create a new code group**, type a name, for example **VS Stress Test** for the code group, and then click **Next**.
5. Choose a condition type of **URL**, and enter the UNC path to the share location of your project, using the format `\\servername\sharename*`. In our example, we typed `\\rootdc\VSTS solutions`.
6. Click **Next**.
7. Choose **Use existing permission set** and select **Full Trust**.
8. Click **Next**.
9. Click **Finish**.
10. Select your new code group: **VS Stress Test**, right click, and select **Properties**.
11. Under the **General** tab, check the two boxes under the **If the membership condition is met** statement, then click **Apply**.
12. Close Mscorcfg,
13. Reboot Machine

Optional - How to configure the SQL repository database for use by Visual Studio

The data from web tests can be stored in a SQL Server database you must however manually create this database using the file, Loadtestrepository.sql, stored in Program Files\MsVS9.0\Common7\IDE on the server where you installed VSTS. Copy the file, Loadtestrepository.sql, to your SQL server and double click to open the file in SQL management studio and then Execute the script to create the web test SQL database called 'Loadtest'. You can then point VSTS to your SQL database as follows:

How to Configure the SQL Server as the Load Test Repository in VSTS :

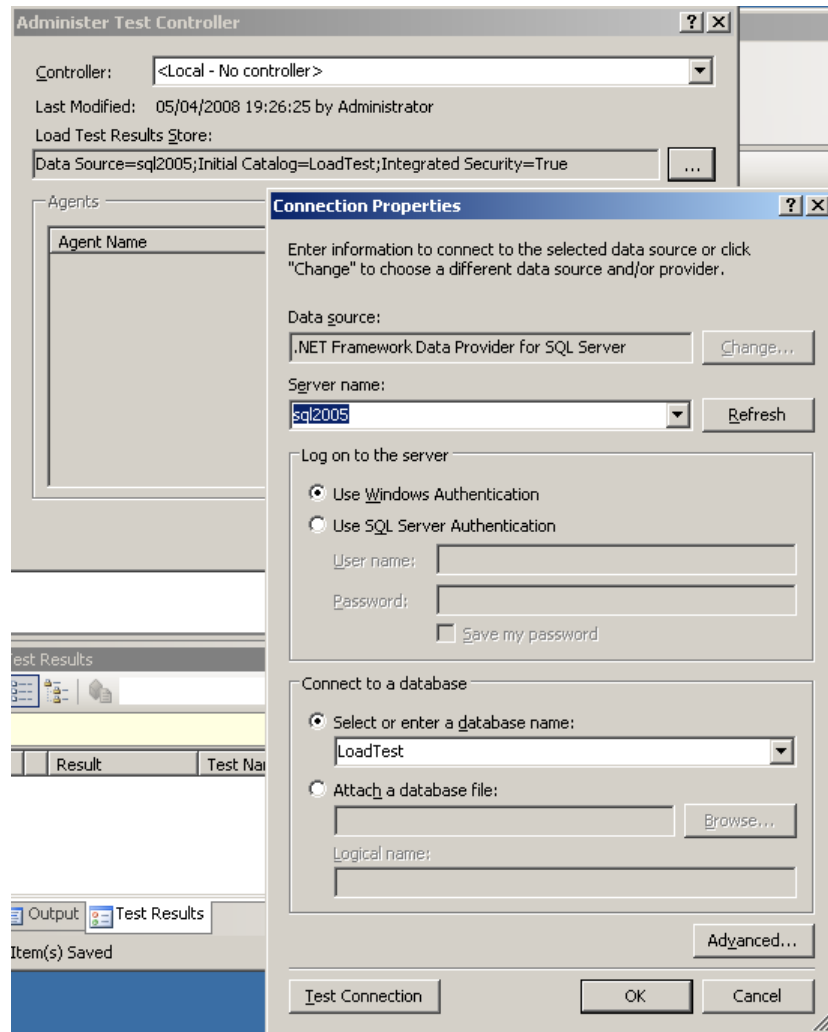
1. Open Microsoft Visual Studio 2008 (**Start -> All Programs -> Microsoft Visual Studio 2008 -> Microsoft Visual Studio 2008.**)
2. On the standard menu bar, click **Test** and then click **Administer Test Controllers** as shown in the following figure.



The Administer Test Controller dialog box opens.

3. Click the ellipse button to the right of the **Controller** text box.
The Connection Properties dialog box opens.
4. In the Server name text box, type the name of your SQL server, e.g., **sql2005** and then click the **Refresh** button.
5. In the **Connect to database** section, select **Select or enter a database name** and select **LoadTest** from the drop down list as shown in the following figure.

* See screen shot on next page *



6. Click **OK**.
The Connection Properties dialog box closes.
7. Click **Close**.
The Administer Test Controller dialog box closes

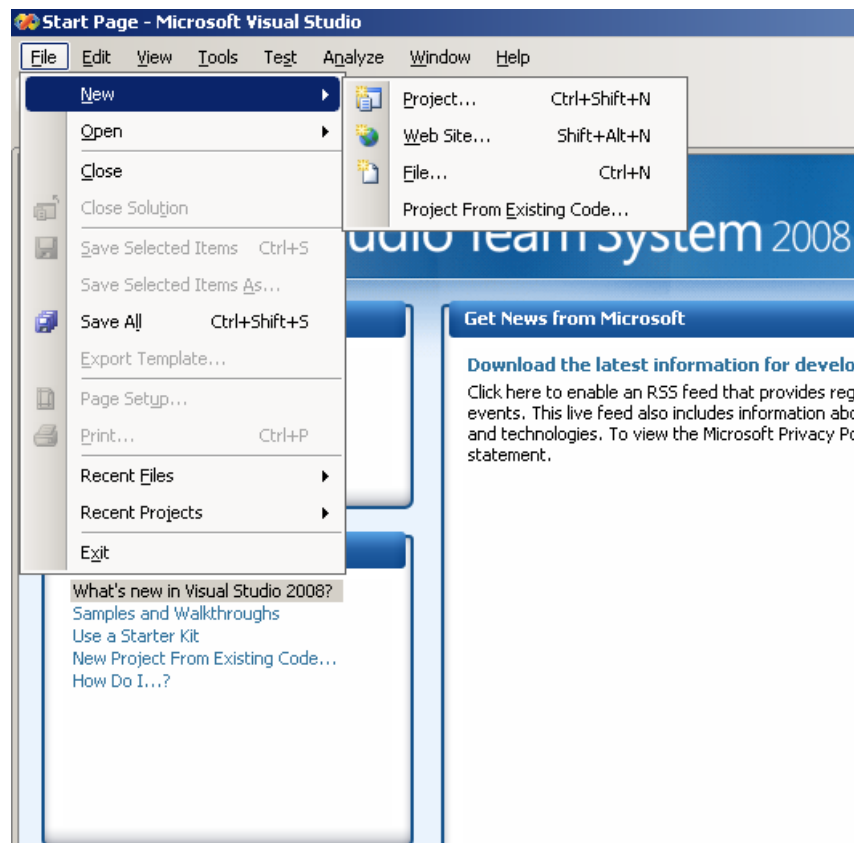
If you choose not to use a SQL server as the repository you can use the local SQL Express database installed along with Visual Studio. Use the same method as shown above just point the location to the local server instead of the SQL server in the server name field.

Step 2 – Build a Solution/Project

VSTS has the notion of solutions and projects. You can have many projects in a solution, where each project can contain a number of tests. You need to plan the number of tests you need to complete and how you are going to group your tests. In our scenario, we will create one test solution, where we will store all our tests against the SharePoint Web application Corp, hence why we will call our VSTS test solution Corporate Collaboration Portal. Against this Corp web application we intend to run a number of groups of Web Tests. We will group these **Web Tests** into a single **Load Test** in the **Project**. The first Project we will create is a basic set of tests where we will visit the Corp web application home, visit department team site and display a list with over 2000 items. Once the test project is created we will run the test multiple times with varying user loads.

In future white papers we will create other projects to test other scenarios, such as, using complex search queries and using different userids when displaying My Sites.

1. Open Microsoft Visual Studio 2008 if it is not already open. (**Start -> All Programs -> Microsoft Visual Studio 2008 ->Microsoft Visual Studio 2008.**)
2. On the standard menu click **File**, click **New** and then click **Project**.



The New Project dialog box opens.

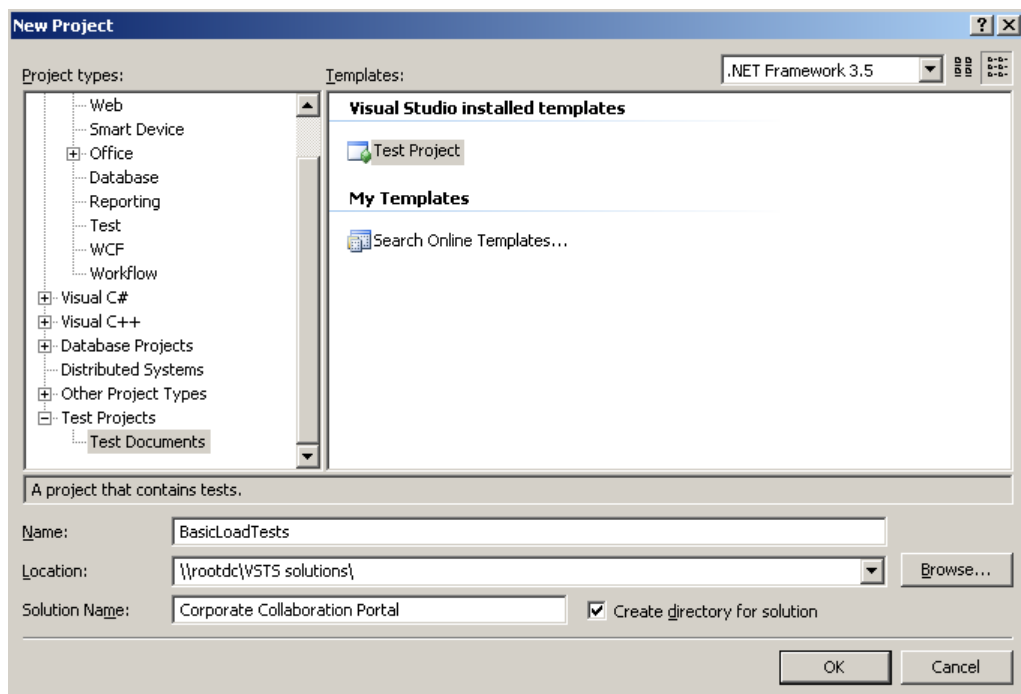
3. Under **Project Type**, scroll down, expand **Test Projects** and click **Test Documents**. Under Templates two templates appear.
4. Under **Visual Studio install templates**, click **Test Projects**.

Note: As VSTS does not contain any specific SharePoint related test template projects, we use the generic test project template as our starting point.

5. In the **Name** text box, type **BasicLoadTests**.
This is the name of the test project. You should type in a name that represents the set of tests you will create in this solution.
6. In the **Location** text box, type **\\rootdc\VSTS solutions**.

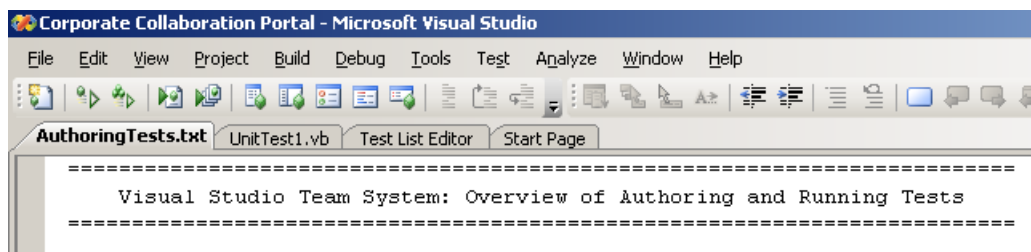
Note: It is not necessary to use a UNC Share if you wish you can use the default location which will be in your documents and settings folder. If you do choose to use the local folder then there is no need to add the trusted folder location as we did earlier in this Whitepaper as your local folder is already trusted for storing the projects data.

7. In the **Solution Name** text box, type **Corporate Collaboration Portal** as shown in the following figure.



8. Leave the **Create directory for solution** check box selected.
9. Click **OK**.

The New Project dialog box closes, and two new files are created, `AuthoringTest.txt`, `UnitTest1.vb`, and displayed in the document window as shown in the following figure. These specify the basic properties of this test project. You can delete these two files if you wish as we will not need them.



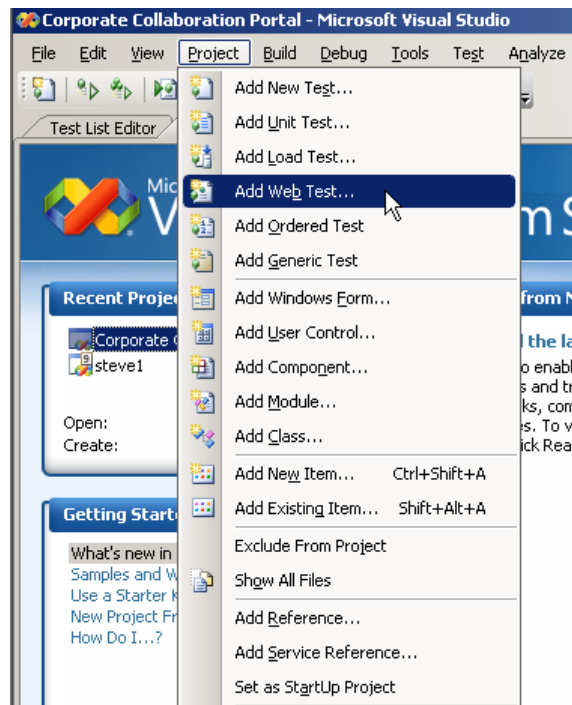
Step 3 – Build Web Tests

There are two components required to complete the test: Web test(s) and a Load test. A Load test can contain one or many web tests. When you create Web Tests in a project they are made available to all Load Test that you create in the project. This is very useful as you may wish to create multiple Load Tests using a mixture of different Web Tests for unique user load or condition testing.

How to Create a Web Test

In this step we will show you how to create a Web test. In this test we will visit the Corp web application home and then create a second test that displays the home page of a team site, which contains a List View Web Part (LVWP) of a list that contains over 2000 list items.

1. On the main menu, click **Project** and then click **Add Web Test**, as shown in the following figure.

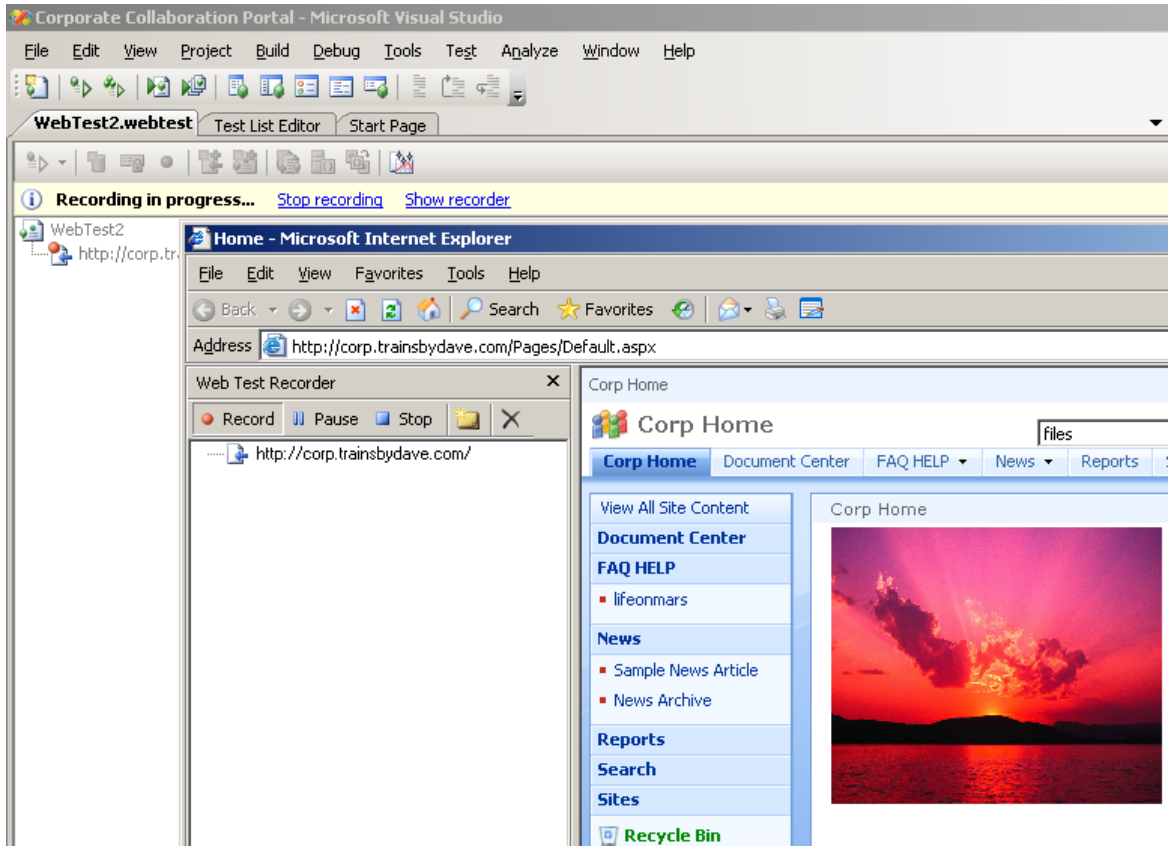


The file WebTest2.webtest file is created and opened in the VSTS document window, and then a new browser window opens.

Note: Your Web Test may say WebTest1.webtest it depends if you have created any web tests previously. You can delete any previous web tests in the solution explorer window.

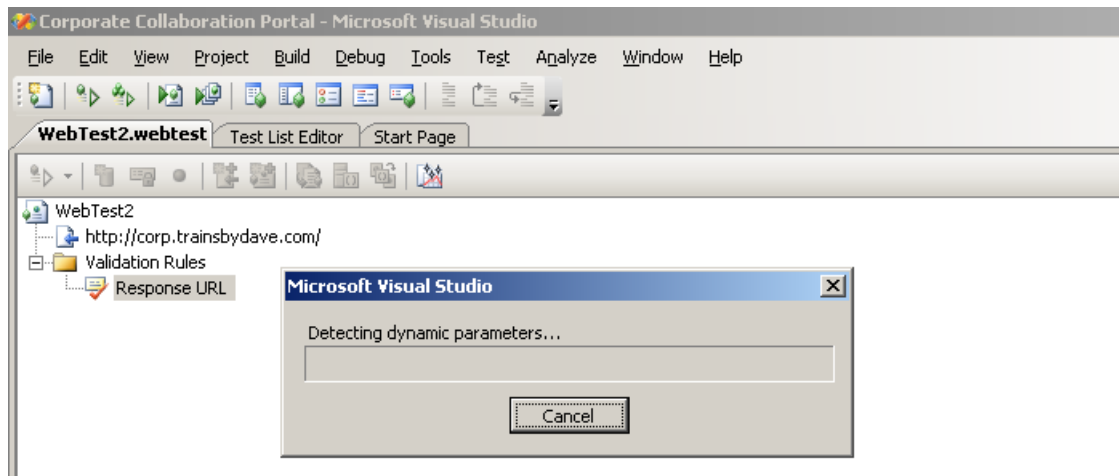
2. In the browser window, click the **Pause** button.
3. In the **Address** text box, type **<http://corp.trainsbydave.com>**.
4. Click **Record** button and then press **Enter**.
If the Enter Password dialog box opens type in a valid userid and password.

The browser renders the home page of the Corp SharePoint site as shown in the following figure.



5. Click **Stop**.

A Microsoft Visual Studio dialog box appears, stating that it is detecting dynamic parameters. VSTS is looking for dynamic elements on the page that required parameters as shown in the following figure.

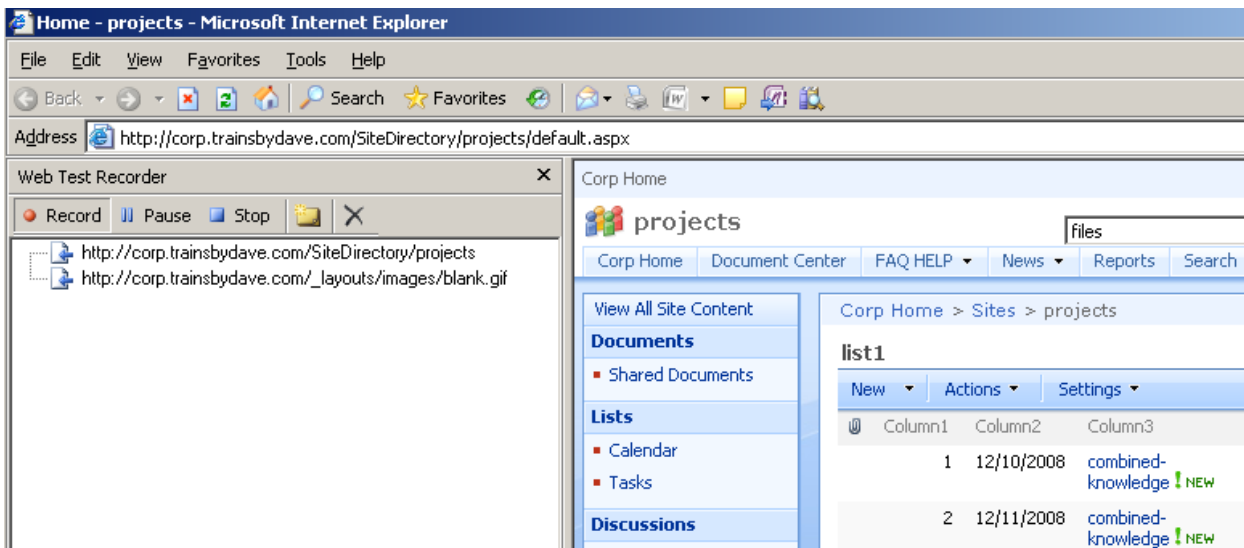


When the Microsoft Visual Studio dialog box closes, the Web test details are displayed in the WebTest2.webtest window, showing the resulting of creating and then running the test.

Tip: Using the Solution Explorer you can rename the file WebTest2.webtest to a Useful name

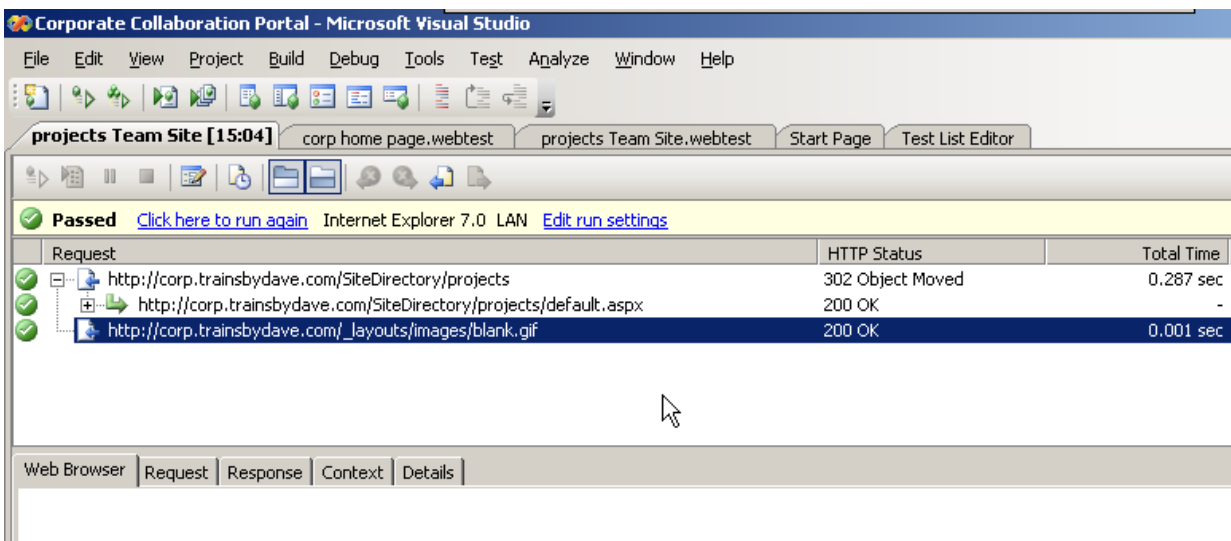
Now we will create the second Web Test

6. On the main menu, click **Project** and then click **Add Web Test**.
The file WebTest2.webtest file is created and opened in the VSTS document window, then a new browser window opens.
7. In the browser window, click the **Pause** button.
8. In the **Address** text box, type **http://corp.trainsbydave.com/SiteDirectory/Projects**.
9. Click **Record** button and then press **Enter**.
If the Enter Password dialog box opens, type in a valid userid and password.
The browser renders the home page of the team site, which contains a LVWP that contains over 2000 list items. (Obviously I could not fit them all on the screen grab but trust me 😊)



10. Click **Stop**.
A Microsoft Visual Studio dialog box appears, stating that it is detecting dynamic parameters. When the Microsoft Visual Studio dialog box closes, the Web test details are displayed in the window.

Note – Try It : You do not need to stop after hitting one URL if you wish you can continue to record as you navigate several sites and lists in the test. Try it and See 😊

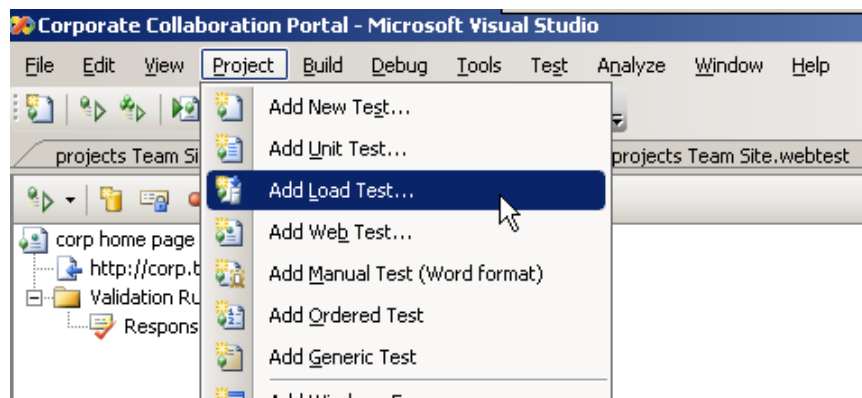


Step 4 – Build Load Tests

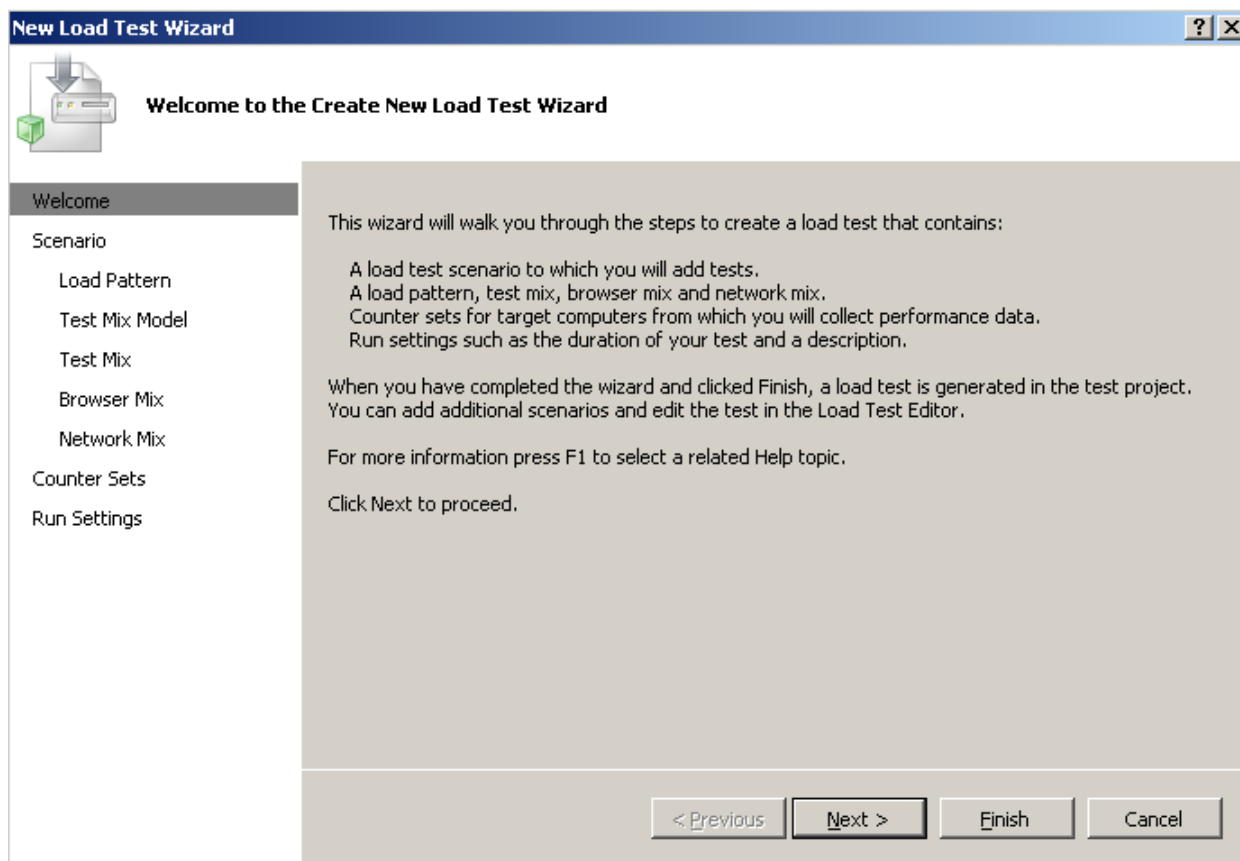
To complete the test we now need to create a Load test that will include both our Web Tests.

How to Create a Load Test

1. On the main menu, click **Project** and then click **Add Load Test** as shown in the following figure.



The New Load Test Wizard dialog box opens displaying the Welcome to the Create New Load Test Wizard page as shown in the following figure.



2. Click **Next**.

The Edit Settings for a Load Test Scenario page is displayed.

3. In the **Enter a for the load test scenario** text box, type **Mid Morning Load Test**.
4. In the **Think time profile** section, select **Do not use think times** as shown in the figure below.

Note: As our test does not include moving between pages, there is no think time needed.

The screenshot shows the 'New Load Test Wizard' dialog box with the title 'Edit settings for a load test scenario'. On the left is a navigation pane with options: Welcome, Scenario (selected), Load Pattern, Test Mix Model, Test Mix, Browser Mix, Network Mix, Counter Sets, and Run Settings. The main area contains the following fields and controls:

- Text box: Enter a name for the load test scenario:
- Section: Think time profile
 - Use recorded think times
 - Use normal distribution centered on recorded think times
 - Do not use think times
- Text box: Think time between test iterations: seconds
- Buttons: < Previous, Next >, Finish, Cancel

5. Click **Next**.

The Edit Load Pattern Settings for a Load Test Scenario page is displayed.

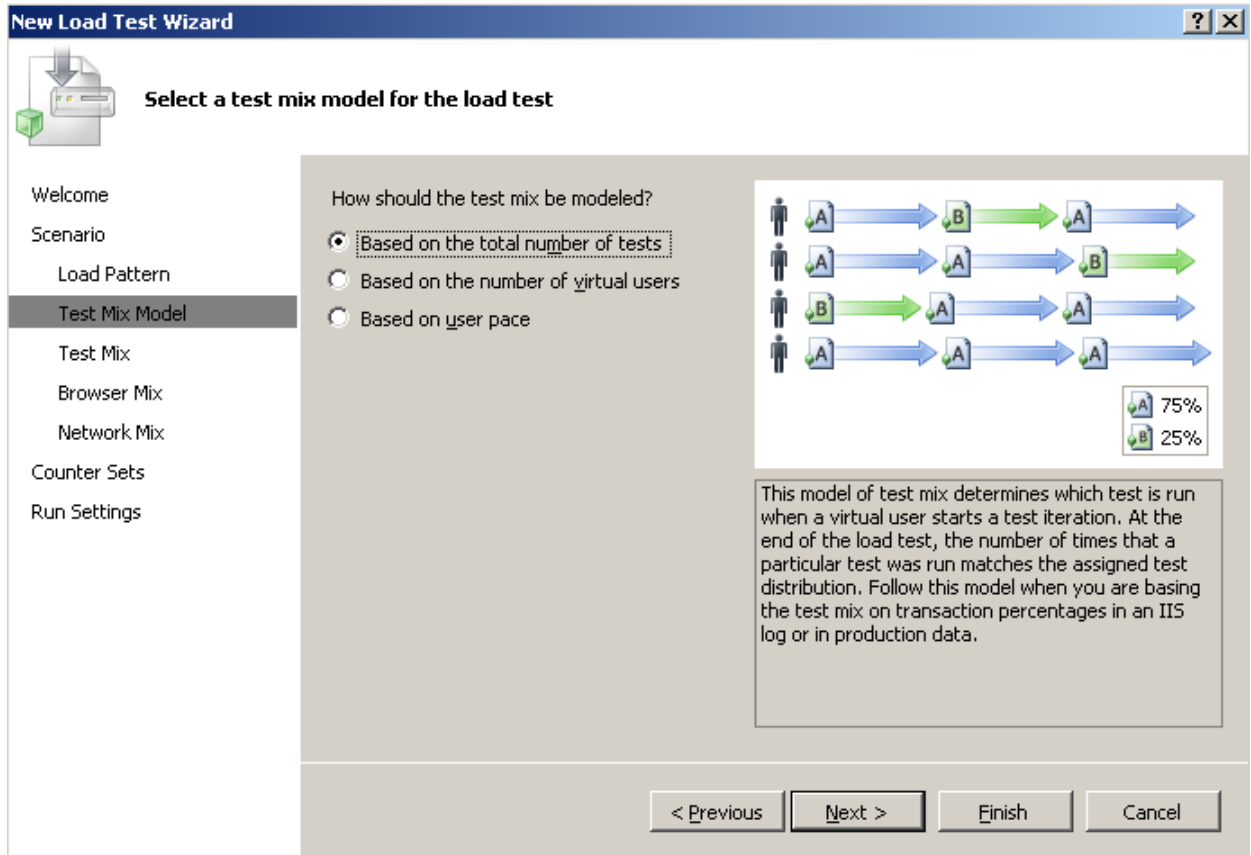
6. Select **Constant load** and set the **User Count** to **50**.

Note: In our test we want to simulate a constant load rather than users hitting the pages in stepped counts. Stepped counts would be useful for testing an environment such as people logging on in the morning as they arrive in office over a 1 hour period and as they switch on their PC's and logon the test can simulate this model.

7. Click **Next**.

The Select a Test Mix Model for the Load Test page is displayed.

8. Select the **Based on the total number of tests** as shown in the following figure.
For our scenario, we want the test to be used by the users evenly.



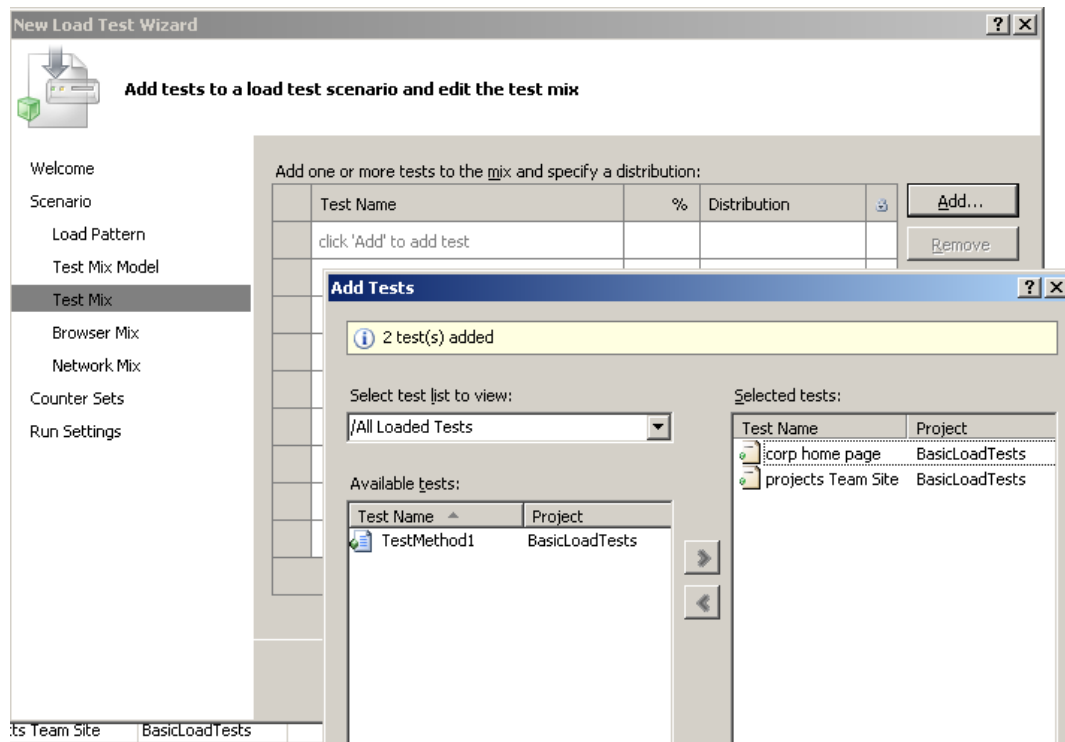
9. Click **Next**.

The Add Tests to a Load Test Scenario and Edit the Test Mix page is displayed.

10. Click **Add**.

The Add Test dialog box is displayed.

11. Add the two Web tests created earlier as shown in the following figure.

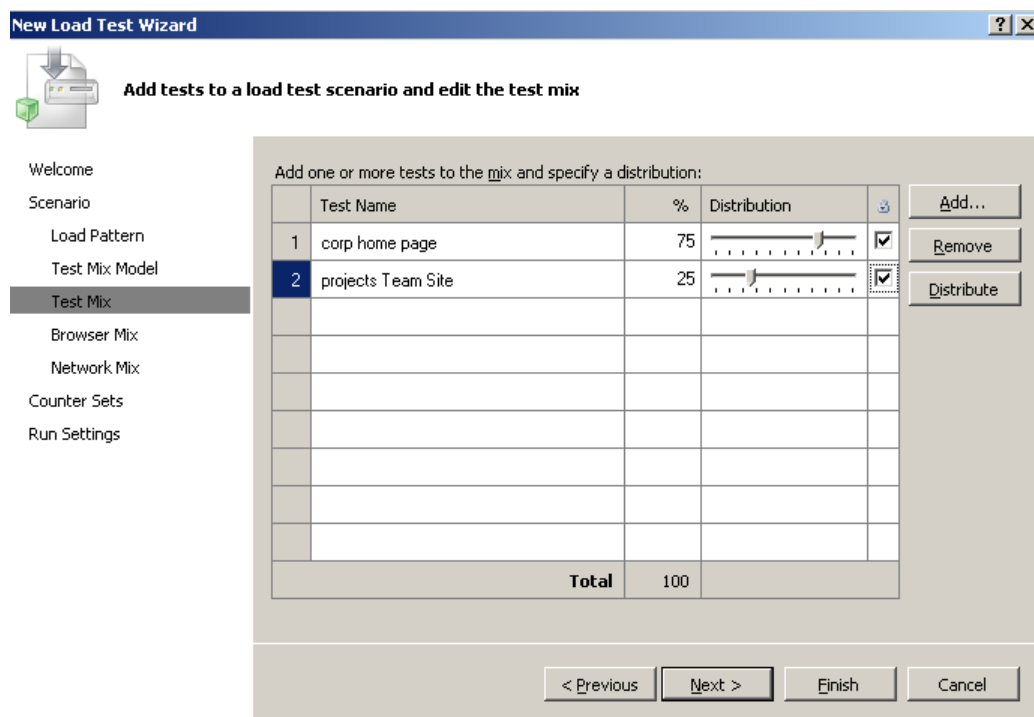


12. Click **OK**.

The Add Test dialog box closes, and the two tests are displayed under Add One Or More Tests To The Mix and Specify A Distribution

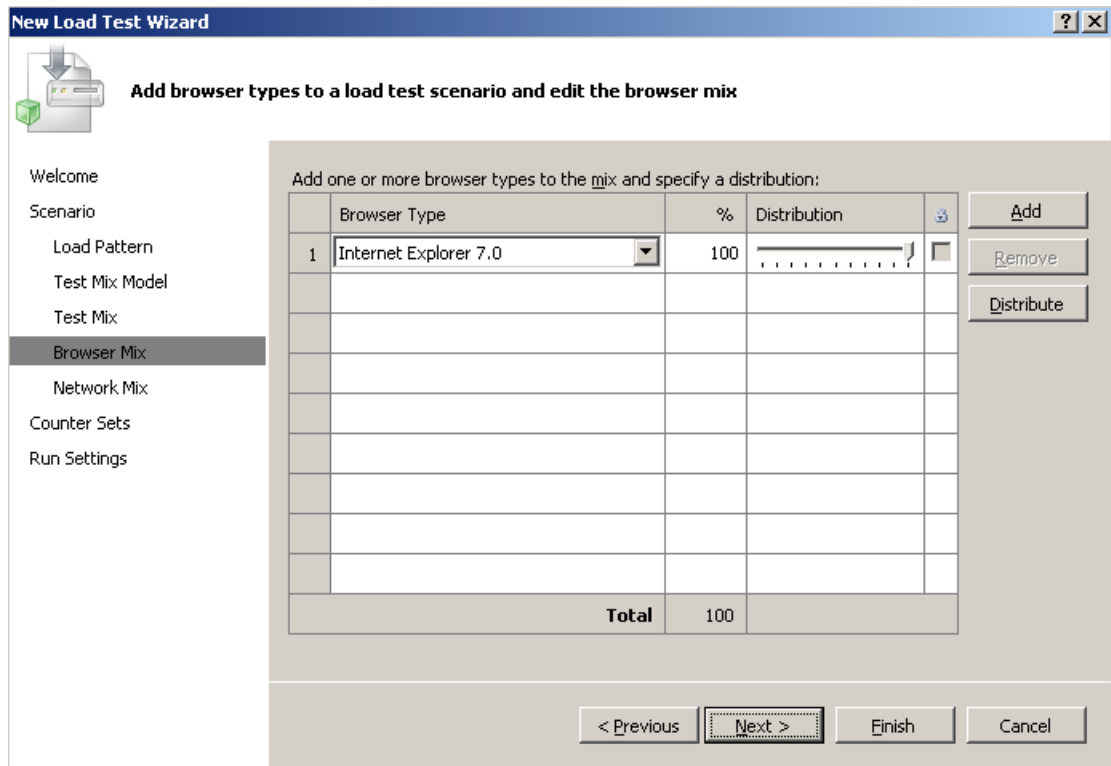
13. Move the sliders so that the Corp test is used 75% of the time and the projects team site test will be used 25% of the time as shown in the following figure.

We are doing this ratio to demonstrate the more users will hit the home page than people hitting the projects page Team Site home page. The more tests you add to a load test the more you will have to think about load distribution.

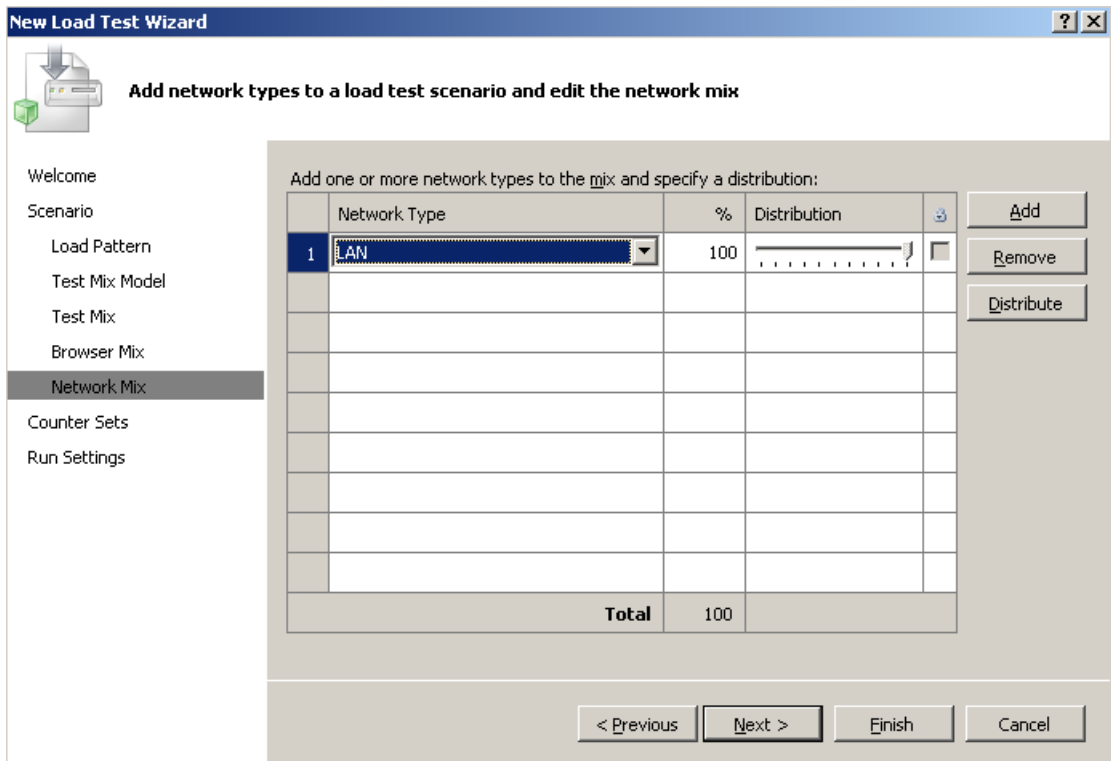


14. Click **Next**.

The Add Browser Types to a Load Test Scenario and Edit the Browser Mix page is displayed as shown in the following figure. We are only using IE 7 clients in our testing.

15. Click **Next**.

The Add Network Types to a Load Test Scenario and Edit the Network Mix page is displayed as shown in the following figure. We are only using Lan clients in this test.



16. Click **Next**.

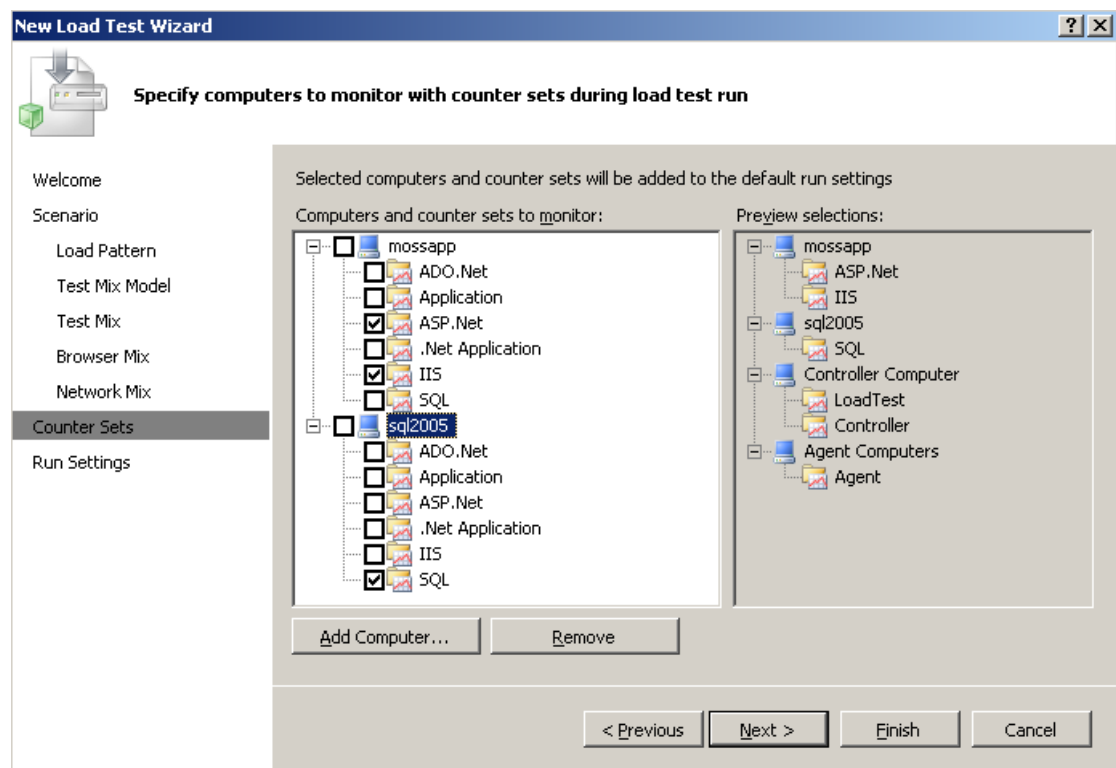
The Specify Computers to Monitor with Counter Sets during Load Test Run page.

17. Click **Add Computer**, and type *mossapp*.

18. Click **Add Computer**, and type *sql2005*.

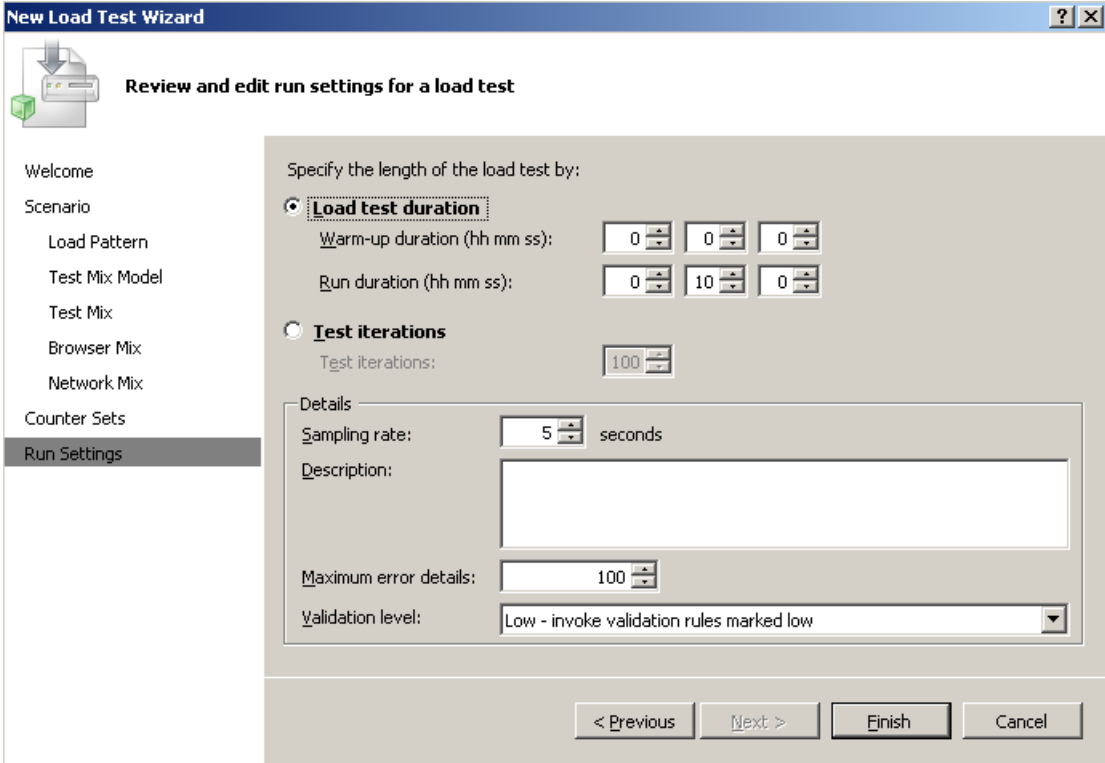
Note: At this point, we could also add domain controller and Web servers, as you would have in a medium server farm. Once the computers are added, select the counters that need to be monitored on the appropriate servers. These Counter Sets have been pre-created and designed by discussions with the product teams. IT Professional and Developers who have used Perfmon will be familiar with the use of performance counters. Additional counters can be added later.

19. Select the relevant counter set for each server, for example, on **mossapp**, select **IIS** and **ASP.NET**, and on **sql2005** select **SQL** as shown in the following figure.



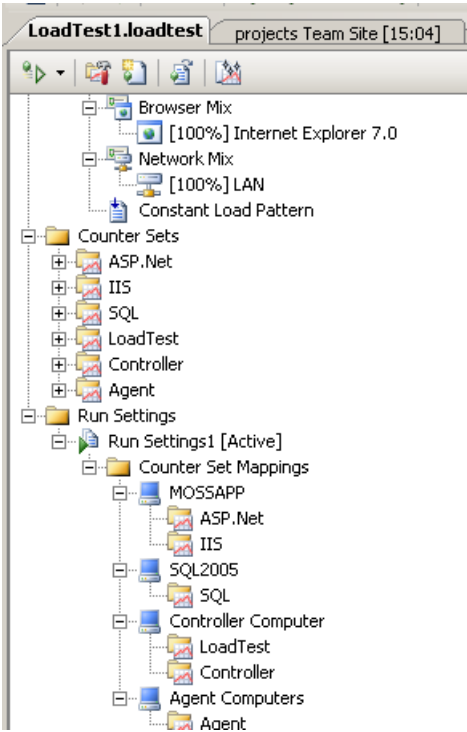
20. Click **Next**.

The Review and Edit Run Settings for a Load Test page is displayed. This page allows you to select the test mix, but as we are doing an evenly based test using 50 users leave selected the Load test duration option as shown in the following figure. You may wish to increase the test length beyond 10 minutes though.



21. Click **Finish**.

The Loadtest1.loadtest file opens where the test is built and displayed. The Counter Sets and Run settings are displayed as shown in the following figure.

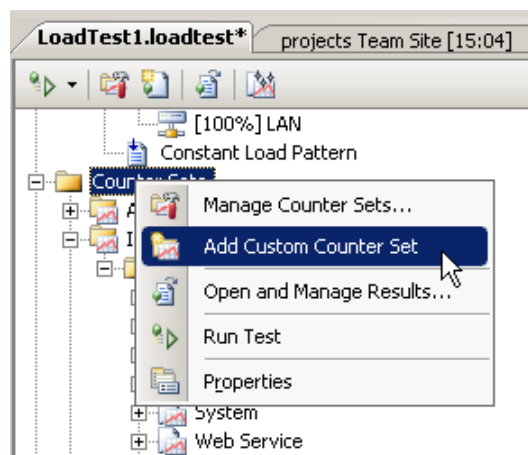


Step 5 – Create Custom Counter Sets and add Counters (Optional)

You can create new Counter Sets as well as adding Counters to the new Counter Set or existing ones. VSTS does not come pre-configured with any SharePoint related Counter Sets, so you may wish to create them. For more information on SharePoint related performance counters, refer to Joel Oleson's blog post: "Good List of Performance Counters" at: <http://blogs.msdn.com/joelo/archive/2007/01/16/good-list-of-performance-counters.aspx> and the Microsoft Press book, "Microsoft® Office SharePoint® Server 2007 Best Practices" by Ben Curry and Bill English, ISBN 0735625387, June 2008.

How to create a new Counter Set and to add more counters

1. Open the load test where you want to add the new Counter Set.
2. Right-click **Counter Sets** and click **Add Custom Counter Set** as shown in the next figure.



A new folder, Custom1 is created under the Counter Sets and under the Custom1 folder is a subfolder named Counter Categories.

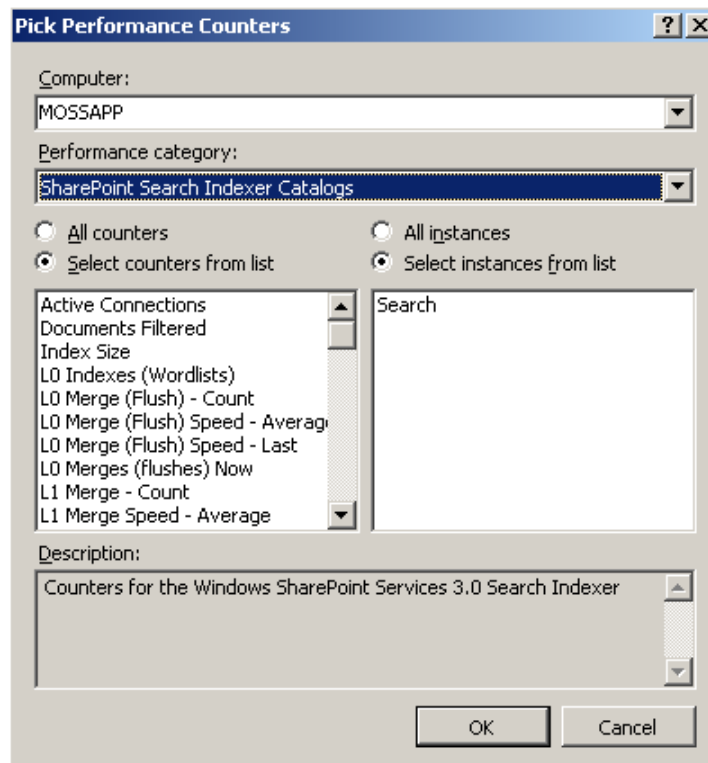
3. Right-click **Custom1** and click **Properties**.
The Properties task pane opens where you can rename the Custom1 counter set to an appropriate name, for example, SharePoint Search.

Next add counters to the counter set as follows:

4. Right click **SharePoint Search** and click **Add Counters**.
The Pick Performance Counters dialog box opens.
5. In the **Computer** drop down list, choose the Computer where the performance counters are installed. In this example, we are creating a SharePoint related Counter set, in the **Computer** drop down list we selected **MOSSAPP** and then from the **Performance Category** drop down list select **SharePoint Search Indexer Catalogs**.

The available counters are displayed.

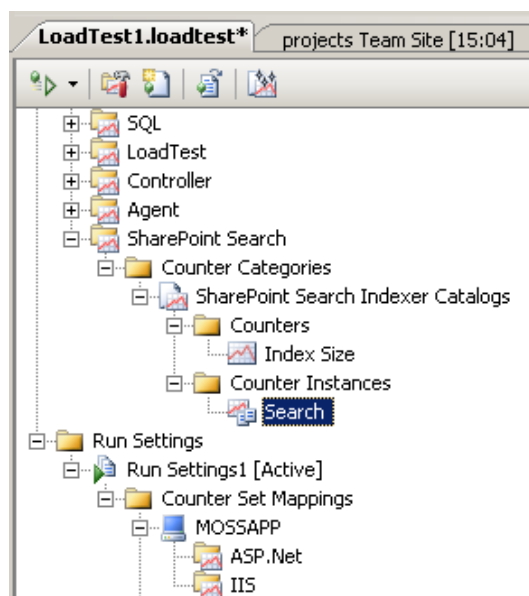
6. Select the required counters and counter instances as shown in the next figure.



Note: When adding new counters make sure you have selected the correct server in the computer field. For example in a medium server farm you will set the Query counters on the Web Servers and the Index counters on the Application Server.

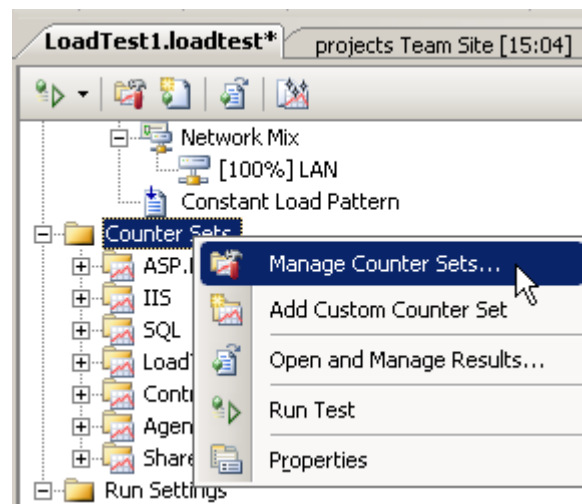
7. Click **OK**.

A new counter set is added to the Counter Categories folder as shown in the following figure.



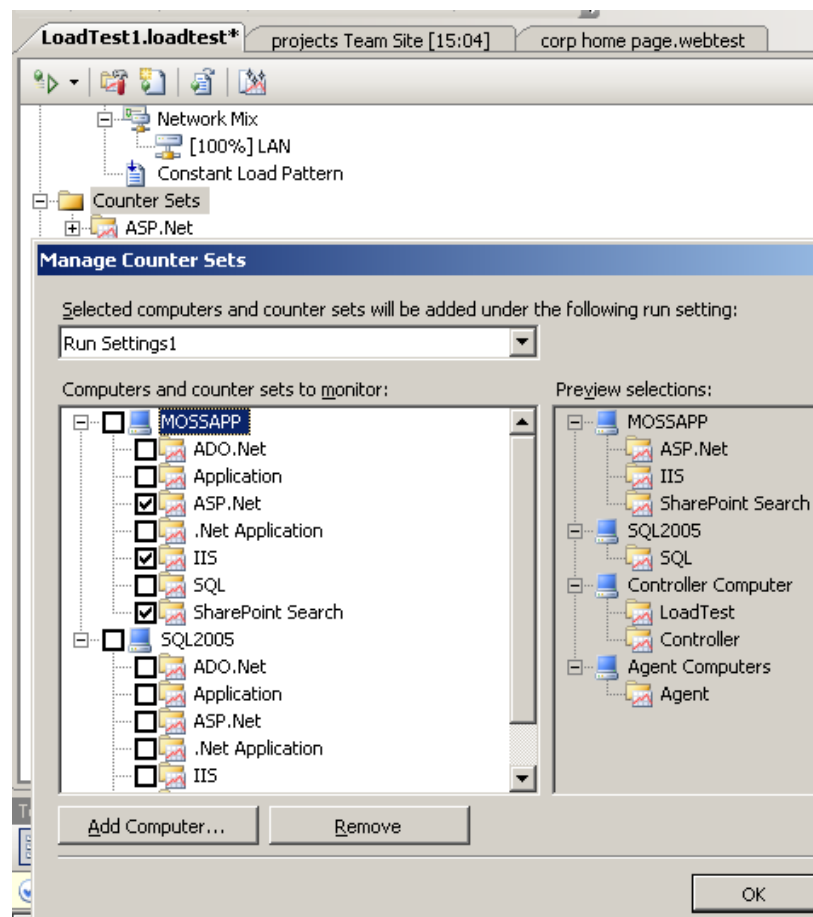
You then need to add this new Counter Set to the specific machine(s):

8. Right click **Counter Sets** and click **Manage Counter Sets** as shown in the following figure:



The Manage Counter Set dialog box opens showing the existing counter sets selected for each computer that is to be monitored.

9. Under **MOSSAPP**, select the **SharePoint Search** check box as shown in the following figure.

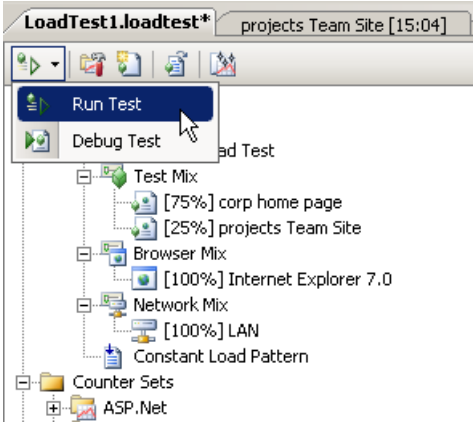


10. Click **OK**.

The Manage Counter Set dialog box closes.

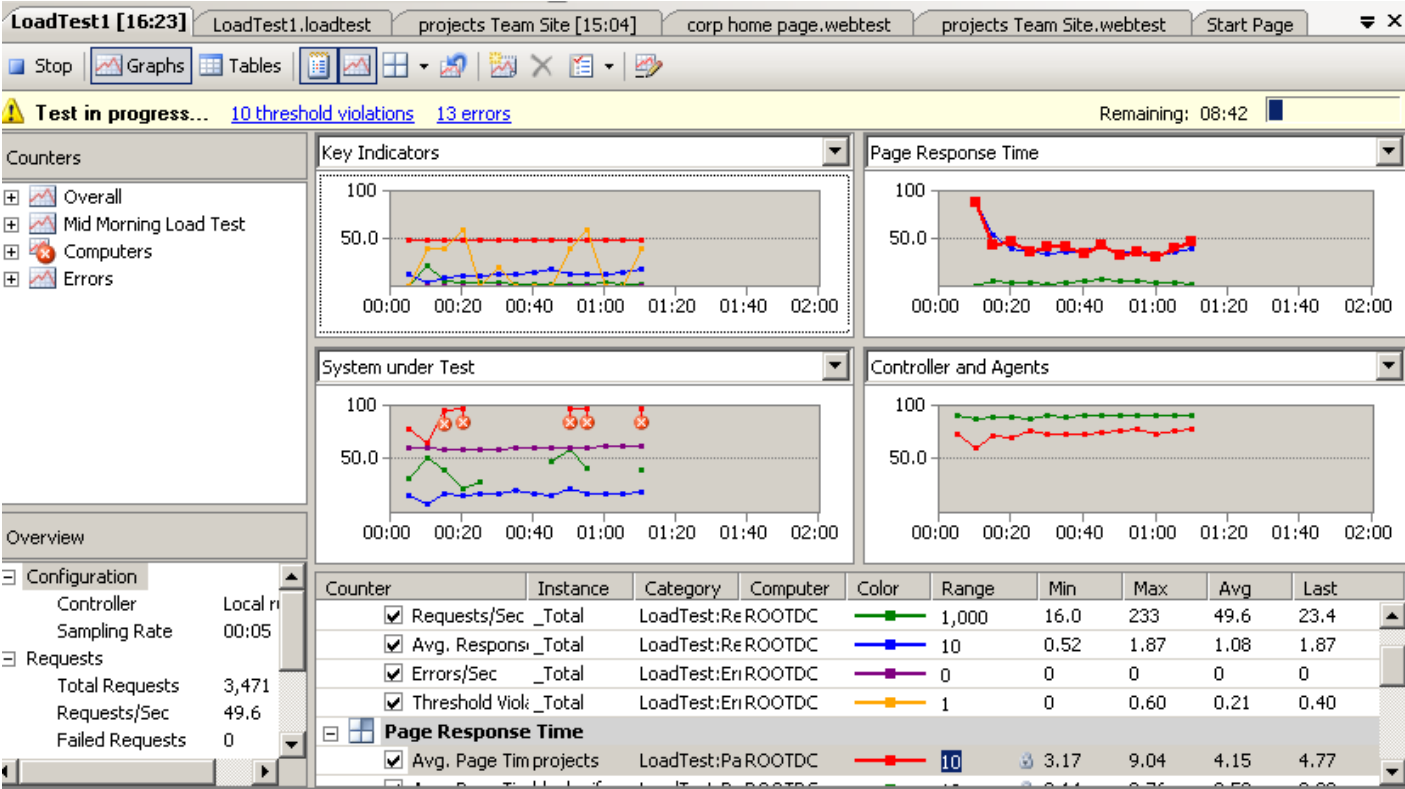
Step 6 – Run Load Tests

1. Click the down arrow to the right of the **Run Test** icon and select **Run Test** as shown in the following figure.



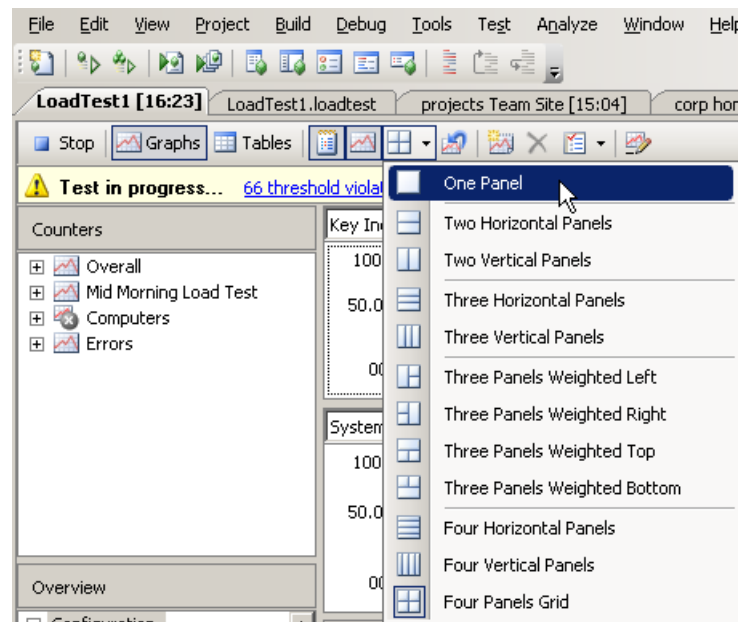
At the bottom of the LoadTest1.loadtest window, the progress of the Test run is displayed.

Once the test is loaded the Test in Progress screen is displayed populated with a number of graphs as shown in the following figure.



The performance counters that indicate that a resource is under stress are displayed as a red circle containing white cross. You can however manually define these warning levels.

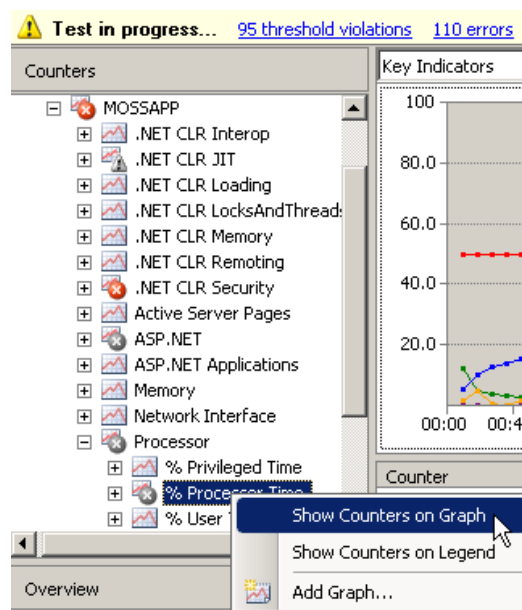
- To display one graph in place of a number graphs, click the down arrow to the right of the **Four Panels Grid** icon, and click **One Panel** as shown in the next figure.



The Key Indicators graph is displayed in one pane.

To add counters to the graph displayed in the one panel, for example to add the % Process Timer Counter for the server MOSSAPP:

- In the left hand **Counters** section of the page, expand **Computers** -> **MOSSAPP** -> **Processor**.
- Right click **% Process Timer** and click **Show Counters on Graph** as shown in the following figure.

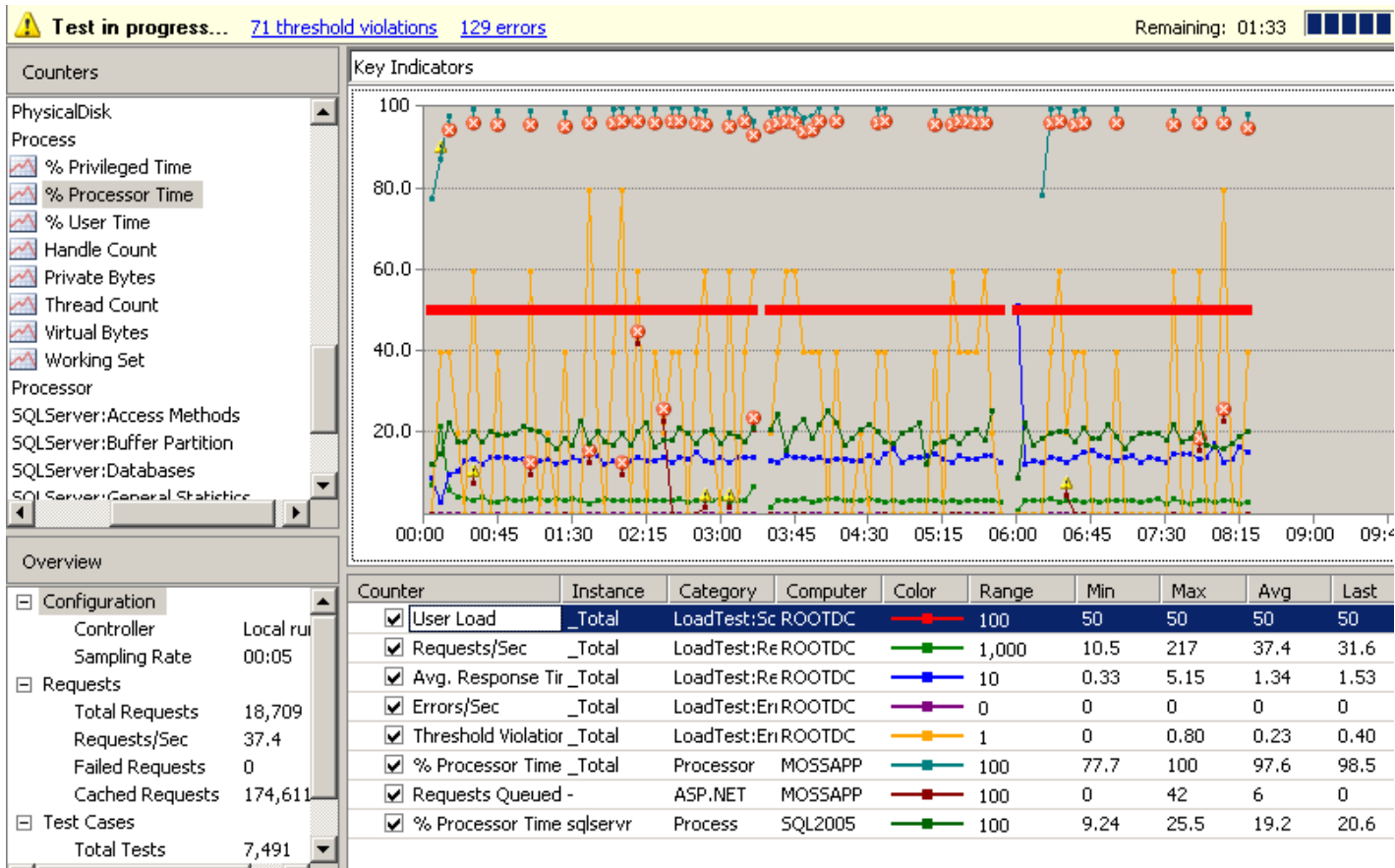


The Key Indicator graph now displays the counter. Add additional counters as you require. You can add counters to the graph, for any of the servers monitored during the load test, for example, we could add SharePoint counter from MOSSAPP and SQL counters from the server, sql2005.

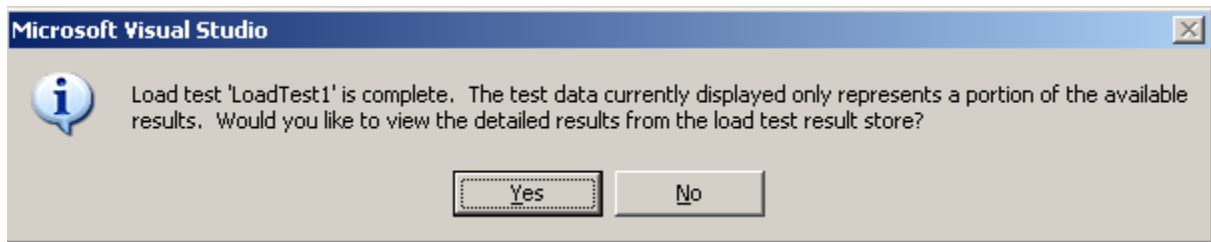
This enables us to compare the affect of the load on the different machines, for example in the figure below the SQL server shows % processor usage at 25%, whereas the load on MOSSAPP was at 100% processor usage.

To start with try displaying only a few key counters rather than too many to spot certain trends. If you are load testing against two web front ends try disabling one web front end and see how that affects the test. Also another nice trick it to add some new content to the tested pages as the test is being conducted. For example try adding a large graphic to the corporate home page and see what effect it has on the monitored counters.

By Adding more web servers and/or optimizing IIS you should see the ration between SQL and the webs changing as the more IIS servers you have throwing requests at SQL the more concurrent user requests your farm will handle.



After the defined load test time expires the test will complete. A dialog box is displayed stating that there is more test data than was displayed on the graphs, providing you with the option of viewing the detailed results as shown in the figure below.



5. Click **Yes**.

The Load Test Summary page is displayed as shown in the next figure. Scroll down to see all the results of the test. Notice how you can see information like slowest page rendering time and average requests per second.

LoadTest1 [16:23] LoadTest1.loadtest projects Team Site [15:04] corp home page.webtest projects Team Site.webtest Start Page

Summary Graphs Tables

Test Completed 100 threshold violations 122 errors

Counters

- Overall
- Mid Morning Load Test
- Computers
- Errors

Load Test Summary

Test Run Information

Load test name	LoadTest1
Description	
Start time	10/04/2008 16:23:52
End time	10/04/2008 16:33:52
Warm-up duration	00:00:00
Duration	00:10:00
Controller	Local run
Number of agents	1
Run settings used	Run Settings1

Overall Results

Max User Load	50
Requests/Sec	33.0
Requests Failed	0
Requests Cached Percentage	90.3
Avg. Response Time (sec)	1.36
Avg. Content Length (bytes)	41,806
Tests/Sec	13.1
Tests Failed	0

Key Statistic: Top 5 Slowest Pages

URL (Link to More Details)	Avg. Page Time (sec)
http://corp.trainsbydave.com/	3.25
http://corp.trainsbydave.com/SiteDirecto...	3.25
http://corp.trainsbydave.com/layouts/im...	0.49

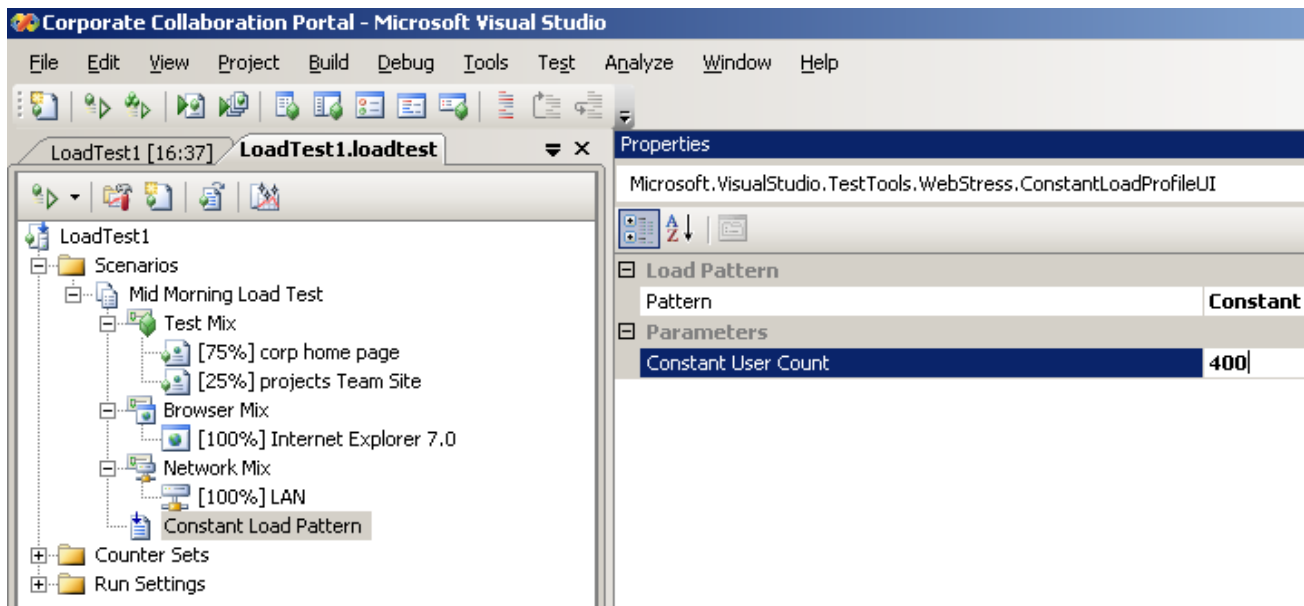
Key Statistic: Top 5 Slowest Tests

Name	Avg. Test Time (sec)
projects Team Site	4.22
corp home page	3.64

Step 6 – Rerun Load Tests

Should you wish to run this load test again, but this time with a different user load :

1. Open the Load test and expand scenarios and then left click **Constant Load Pattern**. The Properties task pane displays.
2. Modify the **Constant User Count** to **400** as shown in the following figure.



This allows you to test the test scenario with a different user counts, simulating peak and non-peak times.

3. Click the **Run Test** icon.

Baseline – Baseline – Baseline

So how do you know that your servers are being stressed by SharePoint traffic? The simple answer is creating your initial traffic analysis on core components like processor and memory without any SharePoint activity. You can only know if your servers are performing badly if you first knew how the performing at the start of fresh builds. Create multiple baselines using core counters on all the servers to enable you to do true analysis over time against user load and SharePoint development. For example what is the affect of a new 3rd party web part at 9.00am compared to the same time last week?

This concludes this white paper. The next white paper in this series will show you how to add additional web test to the project and then add them to the load test.

References

- Stress Testing Microsoft Office SharePoint Server 2007/Windows SharePoint Services 3.0
<http://blogs.technet.com/wbaer/archive/2007/08/02/stress-testing-microsoft-office-sharepoint-server-2007-windows-sharepoint-services-3-0.aspx>
- Good List of Performance Counters
<http://blogs.msdn.com/joelo/archive/2007/01/16/good-list-of-performance-counters.aspx>
- MOSS2007 Testing Using Visual Studio Team System 2008
<http://www.helloitsliam.com/archive/2008/03/10/moss2007-testing-using-visual-studio-team-system-2008.aspx>
- MOSS2007 – Visual Studio 2008 Testing (Creating Team Sites)
<http://www.helloitsliam.com/archive/2008/04/01/moss2007---visual-studio-2008-testing-creating-team-sites.aspx>
- Plan for performance and capacity (Office SharePoint Server)
<http://technet.microsoft.com/en-us/library/cc262971.aspx>
- Plan for performance and capacity (Windows SharePoint Services)
<http://technet.microsoft.com/en-us/library/cc288426.aspx>
- SharePoint Capacity Planning Tool
<http://technet.microsoft.com/en-us/library/bb961988.aspx>
- SharePoint Capacity Planning Tool Released - behind the scenes
<http://blogs.msdn.com/joelo/archive/2008/02/06/sharepoint-capacity-planning-tool-released-behind-the-scenes.aspx>
- SharePoint Capacity Planning Key Info
<http://blogs.msdn.com/joelo/archive/2007/07/09/capacity-planning-key-links-and-info.aspx>

I hope you found this article useful, please do feel free to send me comments and keep checking our web site for more downloads on configuring some of the cool new features of Moss2007.

www.combined-knowledge.com

Steve@combined-knowledge.com



Security Solutions
Advanced Infrastructure Solutions
Information Worker Solutions
Custom Development Solutions



Systems Administration
Systems Engineering