



## Web Load Stress Testing

### Overview

A Web load stress test is a diagnostic tool that helps predict how a website will respond to various traffic levels. This test can answer critical questions such as:

- How will website performance be affected by the number of users?
- How many simultaneous users can my website handle?
- Where are the bottlenecks in my Web applications?
- What's the breaking point for my website, or when will I run out of resources?

Dotcom-Monitor provides external on-demand, self-administered website load test services, implemented and executed within minutes using an online control panel. The service does not require an investment in any hardware or software infrastructure.

Website traffic or loads are simulated from multiple Internet backbones. In addition to testing the Web service, the service tests the Internet connections, routers, firewalls, switches, and anything else that may be part of the Internet infrastructure that handles Web traffic. This approach mimics real user behavior and provides critical insight into potential performance issues.

### How it works

Dotcom-Monitor uses a global network of synthetic monitoring stations (agents) to generate user loads for website load stress testing.

As the test administrator, you can configure:

- Test targets
- Virtual user loads
- Test duration
- User behaviors to simulate



Once the stress test is configured, you can initiate a stress test at any time from the control panel by logging on to your Dotcom-Monitor account. Once a stress test is completed, the results are uploaded to the control panel and available for analysis and reporting.

## Setup

To set up a stress test, you need either a Web load test account or a regular monitoring account. The stress test option is not supported by free or trial accounts.

You can either stress test an existing device or create a new device. A device is a virtual grouping of tasks that are executed sequentially. If you have not created a device, follow these steps.

### Step 1: Creating device

Click the **Add New Device** button on the Device List screen to bring up the device wizard. You will need to create a name for the device, and the name can be anything that identifies the item to be tested. (For example, Webserver1: Shopping Cart.) You can ignore most of the other settings on this screen, as they pertain to monitoring only. Scroll down and click **Continue**.

**:: Set device options. Step 1 out of 3. ::**  
This Wizard will Guide you through the process of editing a current, and setting up a new device for monitoring  
**Name:**  **Frequency**   **Postponed**  
**Owner Device:**

### Step 2: Adding Tasks

On the next screen, select the items to stress test. If you are stress testing a single URL or few separate URLs, you can select the HTTP or HTTPS task. If you plan to stress test a Web application, we recommend using the Macro Recorder to build the stress test scenario. For HTTP/HTTPS tasks, enter the URL and click **Add**. Multiple URLs can be added at this point.

**Update Device Wizard**  
Add/Edit Tasks for **test** Device. Step 2 out of 3.  
**Please choose type of monitoring you would like to perform:**  
**1. Non-secure Web Sites (HTTP)**  
**2. Secure Web Sites (HTTPS)**  
**3. Multiple Web Pages Sequential Monitoring**



To create a script for a Web application, click the Macro Recorder tab on the main navigation menu. From here, you can download the Macro Recorder application to create a test script. You will find a short video tutorial on setting up the Macro Recorder on the same page.



Once a device with tasks is created, it is available for stress testing.

## Stress Test Scenarios

Dotcom-Monitor provides two options for setting up stress tests: cycles and time.

### *Cycles*

Cycles is the simplest option and allows you to specify the total number of virtual users and the number of times that each virtual user will go through a transaction. For example, you may want to test a shopping-cart transaction that has seven steps and is being completed by 200 simultaneous users, where each user completes the transaction 100 times. The repetition of the shopping transaction will provide you with an excellent gauge of whether the application will work correctly under 200 user load.

Number of users:		10
Number of circles:		10
Maximum Processing Time (in minutes):		10

In this scenario example, Dotcom-Monitor agents will execute the transaction 20,000 times (200 users x 100 cycles), and since it is a seven-step transaction, 140,000 Web server requests will be generated. When the cycle option is executed, there are no delays between hits, so Dotcom-Monitor puts the maximum load at any given time to your target environment.

### *Time*

The time option allows more flexibility to create different testing scenarios and approximate environments that are closer to real-life loads. This option allows for steps within a load test. Each step can specify virtual user load, the number of minutes to run under such load, and the minimum and maximum time delays between hits.



Work Time	Number of users	Delay Min (ms)	Delay Max (ms)	Action
00:05:00	10	2	7	<a href="#">Remove</a>
00:10:00	20	1	6	<a href="#">Remove</a>
00:20:00	30	0	0	<a href="#">Remove</a>
<input type="text" value="20 min"/>	<input type="text" value="10"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<a href="#">Add</a>

As an example, let's continue using the seven-step shopping cart scenario. From experience, you know that users usually spend on average between three to seven seconds between clicks and that the application should handle approximately 100 simultaneous users. To measure the application response times under different user loads, the following test can be executed using the time option.

Step	Time to Run (Min)	Users	Min Delay (ms)	Max Delay (ms)
1	5	50	3000	7000
2	5	100	3000	7000
3	5	100	1000	2000
4	5	200	0	0

In this example, the stress test will run for 20 minutes. For the first five minutes, it will run with 50 users generating random delays between three to seven seconds. This should establish a base line for the application. Then, the test will scale up to 100 users with the same delays. This is the target test. From here, you can deviate to see how the application performance changes under different conditions. Note that in Step 3, the user's time decreased; and in Step 5, there was a maximum load of 200 users without any delays.

### Starting a Stress Test

A stress test can be started at any time from the control panel by simply clicking the **Stress** link next to the device name. This launches the Stress Test Setup screen. As discussed above, there are two options for running a stress test. You can run the test by cycles or time and scenario.

During a cycles test, you can specify the maximum processing time in minutes (up to 60 minutes). By default, Dotcom-Monitor forces test termination after 60 minutes. During a time test, you can specify the work time (time to run for each user).



Once the test is configured, click **Submit**. You will be shown the total cost of the test. Once you click **Agree**, the stress test will be queued. We allow one global stress test per system. If there are other load tests are running, your test will be entered into the line, and the exact time for the start of the test will be displayed. If there are no other tests, it usually starts within 30–90 seconds. You can click **Continue** to see real-time test progress. If something goes wrong, you can terminate the test by clicking **Cancel**. Termination may take 60–120 seconds before the load stops.

### Internal Systems Monitoring

Dotcom-Monitor recommends internal systems monitoring that follows the path of your Web traffic. These monitors can show potential problems or pinpoint potential application bottlenecks. Monitored items may include:

- Router bandwidth
- Firewall utilization
- CPU
- Memory
- Disk usage on servers

Monitoring these areas will allow you to correlate different loads with effects on your infrastructure.

### Results

Once a stress test is completed, you will receive an e-mail with several reports, if e-mail was specified under reporting options. If you did not specify email or somehow never received this report, all results are saved online. To access these reports, log on to your account and click on **Device List**. From there, you can click on the **Stress** link next to your device. Follow the Stress Test History link on the next page to access the results page.

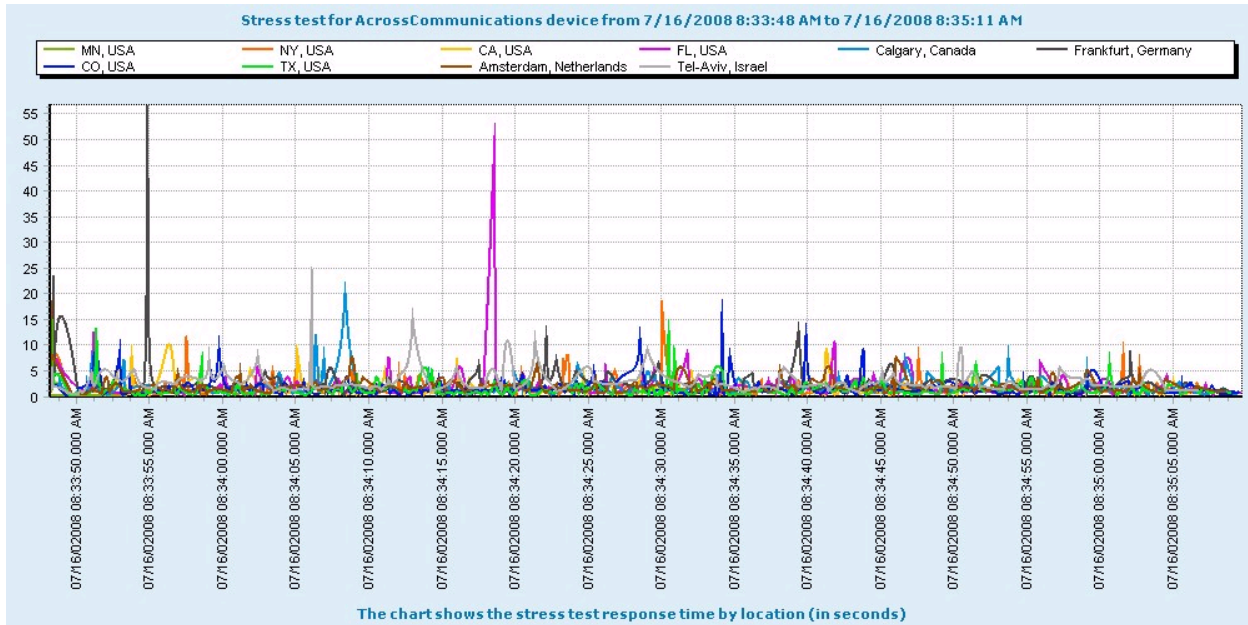
12/12/2007 11:29:39 AM	12/12/2007 11:34:40 AM	Failed	1	1	txt,html,css,gif	vadimm@dana-net.com	\$ 0	<a href="#">Summary</a>	<a href="#">Csv</a>	<a href="#">!</a>
3/21/2007 9:48:17 AM	3/21/2007 12:14:56 PM	Finished	500	500	txt,html,css,gif	vadimm@dana-net.com	\$ 576.46	<a href="#">Summary</a>	<a href="#">Csv</a>	<a href="#">!</a>
3/21/2007 9:08:23 AM	3/21/2007 9:16:47 AM	Finished	10	10	txt,html,css,gif	vadimm@dana-net.com	\$ 1	<a href="#">Summary</a>	<a href="#">Csv</a>	<a href="#">!</a>

We provide three options to review results.

- A summary link that shows aggregated statistics for performance, errors for the overall transaction, and each step of the transaction.
- A CSV file in an Excel format with all data collected during the test. This provides information for each hit to your target site. You can use this data to create your own reports or to drill down on why and when there were delays and from what locations.



A chart shows the response time for transaction per location. This is useful in determining whether all locations responded similarly or if some networks have larger latency accessing your site versus others.



#### Additional Help:

Video Tutorial On Setting up and Executing Stress Test:

<http://www.dotcom-monitor.com/tutorials.asp>

#### FAQ:

<http://www.dotcom-monitor.com/faq.asp>

#### Support:

<http://www.dotcom-monitor.com/contacts.asp>