TMurgent Technologies

DeNamiK LoadGen
Beta Product Review

TMurgent White Paper

White Paper by Tim Mangan
Founder, TMurgent Technologies

May, 2006

Introduction

This White Paper provides a review of the "*DeNamiK LoadGen"* from DeNamiK (www.denamik.com). This technical review is based on the very first Beta version of the product, as well as discussions with the product architects on what we should expect with the final release.

LoadGen is a load/stress testing platform for the Citrix marketplace. It supports the centralized management of load testing a Citrix Server or Farm using multiple clients.

In this way, the product most directly competes with *Scapa Test and Performance Platform* from Scapa Technologies (www.scapatech.com), as well as products from Mercury Interactive. In writing this white paper I have to admit that I have not used Scapa nor the Mercury products. I tend to write my own test scripts and test bed when I need one rather than pay a premium amount of money for a general purpose tool. DeNamiK says their product not only competes with these tools, but also complements Scapa and LoadRunner if you already own them. In this case you could use LoadGen to ease the test process and these other tools for their more advanced reporting capabilities.

LoadGen also "competes" with another test bed that I, and probably most of you, have tried – the "Citrix Test Server Kit" (CTSK). This test kit, although free, has frustrated many a tester with the seemingly inability for most reasonably educated professionals to produce tests using anything other than the stupid sample test scripts that come with it. It was through their own frustrations with CTSK and cost of the more expensive tools, which they used as part of their consulting business, which led the guys at DeNamiK to conceive this product.

In the testing LoadGen I found a number of small issues that I will try not detail in this paper. The purpose of their Beta, after all, was to get an early version of the product into the hands of others who would try to use it differently than they intended. In this they succeeded, and I expect that small things I noticed will either be fixed with the release or are too insignificant to deal with right now. So I will try to stick with the "big picture" items.

General Impressions

Beta means many things to different Development oriented companies. DeNamiK calls this beta a "Release Candidate", but make no mistake about it; I was testing a beta product. Clearly, in this case, Beta means that this is an early look at a brand-new product. While, at least with this Beta version[1], it still needs a lot of spit-and-polish, it shows itself to be an important new product for this market.

The product clearly aims to replace Scapa, but at this point is somewhere between CTSK and Scapa. DeNamiK has not announced pricing, but if they charged half of what Scapa costs it would be a bargain.

The goal of the tool is to help you automate repeatable load test scenarios using multiple client machines. Their approach is to use a centralized console to define user accounts, tests to be run, and machines to use as clients.

LoadGen does a good job of integrating with Active Directory to help with the users. It leverages Citrix Published Applications for the test scripts (however leaving the publishing to you), and effectively uses networking tools to locate and interact with the client machines.

One limitation of the product is that it only works with Citrix (at least at this point) and is of no use in testing Terminal Services without Citrix (or with other products like Jetro, Tarantella, and the like).

Most lacking in the Beta is good documentation. But what else is new? If the initial Beta of any product had good documentation I would probably be skeptical about the technical ability of the company. A good overview of the entire process would go a long way and after talking with them I am sure they will have that in place prior to release.

What It Takes

We will break this into three parts, the Server Farm, the test console machine, and the test client PCs.

You need a Citrix Server or Farm to test. I used both an FR3 server as well as an MPS 4 server in my tests and it didn’t matter. DeNamiK LoadGen does nothing to the Citrix Server itself except for to use it.

[1] DeNamiK LoadGen Release Candidate (0.9.2290.35129)

The DeNamiK LoadGen console (they call it the "Director") can be placed on any machine. You probably do not want to put it on the terminal server under test, but if you did it would work. DeNamiK recommends Windows 2000 SP3 as a minimum operating system. I tried it on a 2000 system without any service packs and had some issues working with Active Directory which is not surprising considering the changes AD[2] has gone through. I also tried it on a server with SP4 and another 2003 server with SP1 where it worked great. There is probably no reason you couldn't install it on a desktop OS also, although I didn't have time to try it. The LoadGen console has two requirements other than the OS version. You need .NET Framework 1.1 and you need a Citrix client (version 7.0 or above) installed. The LoadGen console requires a lot of screen space, so you need a 1024x768 desktop to run it in. The Director consists of a windows service and the GUI elements.

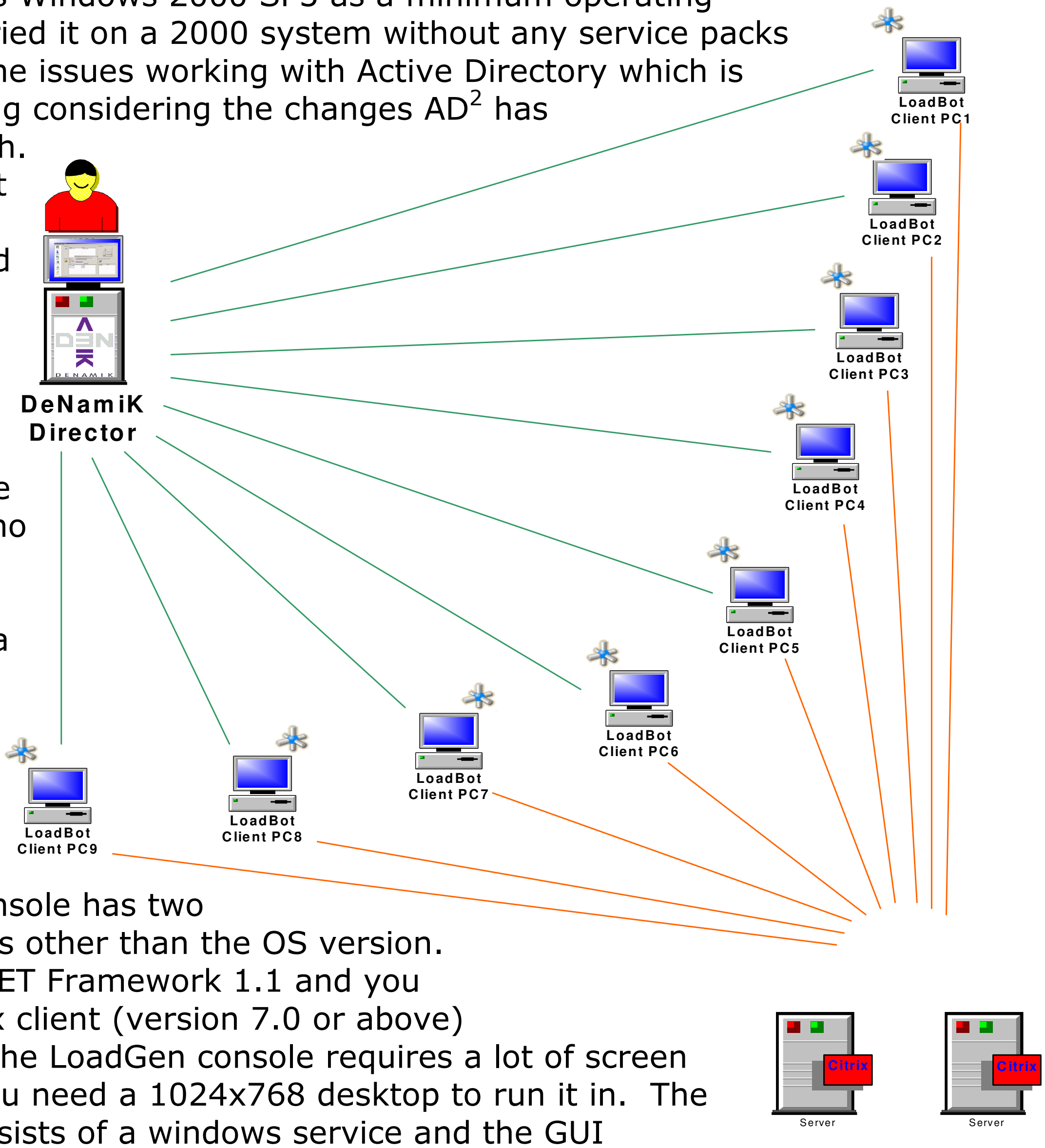


Figure 1 - Typical LoadGen Deployment

LoadGen utilizes multiple client PCs to launch the Citrix sessions from (See Figure 1). The software that runs those sessions is called a "LoadBot". Bot is an appropriate term as the LoadGen Director truly automates the client end using "Robotic" techniques. You never have to touch these client PCs! The LoadGen Director takes care of

[2] AD: Microsoft Active Directory

installing[3] the software. The Director also takes care of what it calls "Activating" and "Deactivating" the LoadBot, which really means it is starting, and stopping the DNKexec.exe and DNKLoadBot.exe executables on the client machine.

Finally, the Director talks with the LoadBot via TCP to start/stop/monitor the individual user session tests. DNKexec runs as a local system service and controls the DNKLoadBot, which in turn launches Citrix sessions via wfcica32.exe. These client machines also need to be Windows 2000 SP3 or above with a Citrix Client and .Net 1.1 installed. Sorry, no WinCE or Linux clients! In addition to testing the usual Win2k, WinXP, and Win2003 machines as clients; I wanted to try a Vista (Beta) Desktop[4] and a Windows 2003 x64 Server[5] as clients but they are not (yet) supported. Vista, in particular may be problematic for many management applications, such as LoadGen, due to the improvements in security[6].

Many, but not all of the machines in my test lab are Virtual Machines, hosted either on Microsoft Virtual Server or VMware Server. This made no difference to LoadGen[7].

Testing with LoadGen

The one missing ingredient in LoadGen would be the load tests to run on the server itself. But heck, we already know how to load up a server! All you need are some self running applications or scripts that are set up as published applications.

For this testing I used my test applications taken from my ToolCrib. These are a set of self-running applications that stress servers in

[3] Actually more like copying the files. There are no dlls or registry use; the program does not show up in "add/remove" programs on the client.

[4] We tried to get Vista to go (the Citrix client works OK), but the OS Beta was unstable with AD itself so remote authentication was failing and we were unable to determine if was the Beta OS or the Beta LoadGen at issue.

[5] LoadGen should be able to work as a 32-bit app on this OS, but we had issues running the LoadBot once installed. Probably WOW related, but we didn't have time to investigate.

[6] Microsoft has hinted that some of the more sever restrictions contained in the Vista Beta will be turned off by default in the final release of the OS. This may help.

[7] One exception to this is that the Director required a slightly larger desktop than 1024x768, however this is probably just a Beta issue that will be cleared up.

particular ways. I released them to the community[8] earlier this year at my website (www.tmurgent.com). You can also use any scripting techniques you like (vb/cs/ws-script, kix, perl, or WinTask, for example). One other tool you may want to consider for automating applications like office as part of your test would be AutoIT[9].

We can break up the process of setting up a testing scenario into the following ten steps (you might find this helpful until the release documentation is available):

1. Get your Infrastructure ready.
2. Install DeNamiK LoadGen on the Director (console) machine.
3. Create test user accounts and "Profiles"[10]. These test user accounts will live in your Active Directory and are the logon credentials that will be used to run applications on your Citrix Server. The "Profile" ties together a group of like named users and a published application that they will run. As will be described later, you have two choices in creating/selecting user accounts and making the "Profiles".
4. Verify the "Profiles".
5. Define LoadBot Administrators
6. Select LoadBot Machines and get Bot loaded on it.
7. Verify LoadBot Machines.
8. Define a "Test Scenario"
9. Run a Test.
10. Create a report on the results.

We can walk through each of these steps in more detail to get to know the product better.

1. Get Your Infrastructure Ready

Identify or install Citrix Servers and Active Directory. Now is the time to add a test group to your AD schema if desired. This doesn't help LoadGen, but it does make publishing applications to your testers via Citrix easier.

Identify the machine to host the LoadGen Director and get a Citrix client installed on it.

[8] Free for VAR or Enterprise non-commercial use, ISVs need to obtain an inexpensive license to use.
[9] www.autoitscript.com. AutoIT is freeware, although you can make a donation to help defray website costs.
[10] Some of the naming used in the product may change by release.

Identify the machines to be the client LoadBot machines and get a Citrix client installed on them.

2. LoadGen Console Installation

The product installer is nothing out of the ordinary (which is always a good thing). It is an MSI based installer, which is always a good thing. If .NET 1.1 is not installed it will let you know and offer to take you to a Microsoft web-site for the install. Microsoft wants you to install .NET 2.0, which will not help you; you will have to search the Microsoft site for the 1.1 installer.

When you first run the console, you will find that you need to email a code and receive an activation key in return to use the product. If you do not have a Citrix Client installed it will require you to do so now.

Once this is complete, LoadGen starts up with a fairly simple menu, as shown in Figure 2.

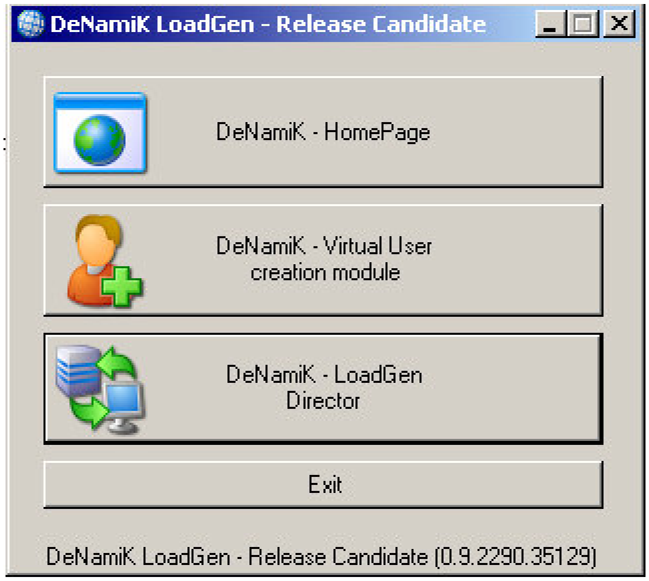

DeNamiK LoadGen - Release Candidate
DeNamiK - HomePage
DeNamiK - Virtual User creation module
DeNamiK - LoadGen Director
Exit
DeNamiK LoadGen - Release Candidate (0.9.2290.35129)

Figure 2 - Main Menu

3. “Test User” Creation and “Profile” Generation

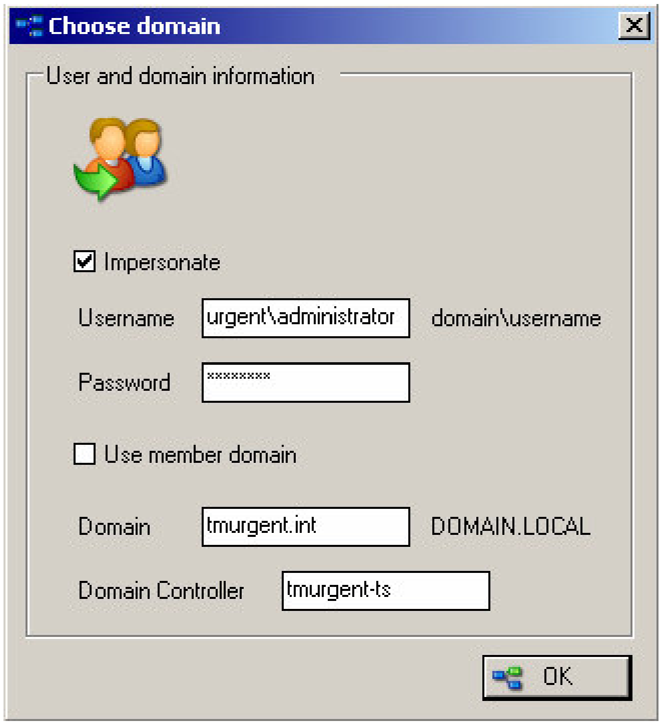
This is two different things handed in one step.

LoadGen needs to have test user account names in Active Directory that all start the same and end with numbers, where the numbers all have the same number of digits by zero filling (like test001, test002, test003, etc…).

A “Profile” is the combination of a numbered set of these users and a published application.

You can have LoadGen create the test user accounts for you, or you can use existing AD users. This dictates how you take the next step, as you would use two different methods to accomplish this depending on if you need to create the users.

When you need to Create New Users



☑ Impersonate

Username: urgent\administrator (domain\username)

Password: *********

☐ Use member domain

Domain: tmurgent.int (DOMAIN.LOCAL)

Domain Controller: tmurgent-ts

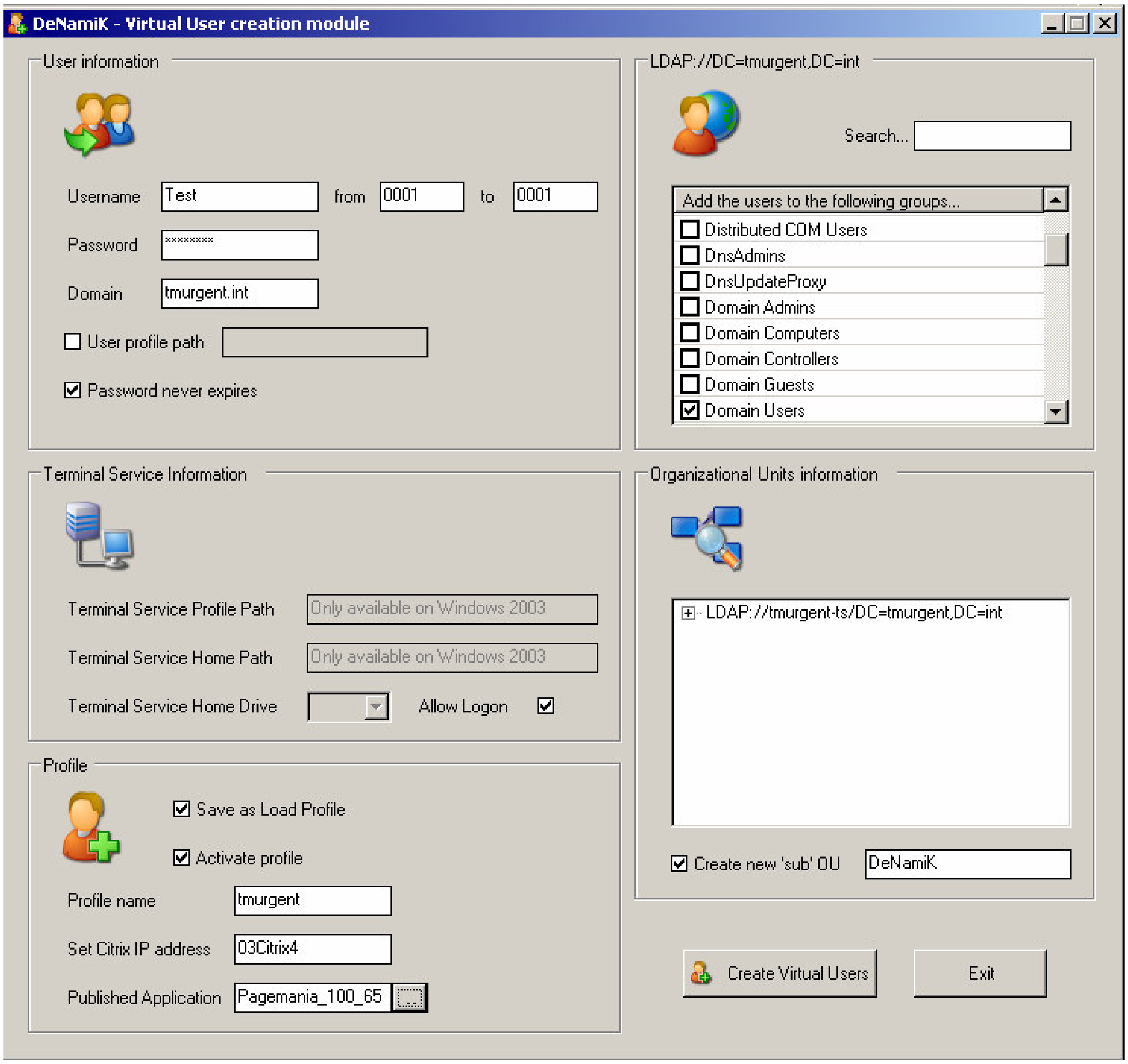
Figure 3 – Create New User: AD Account Credentials

From the main menu, click on the “Virtual User Creation Module” to perform this function.

You first supply the name of the AD server and domain, as well as a user account with permissions to add users to the domain (see Figure 3).

Normally, you can just use the defaults on this dialog (Impersonation disabled and “Use member domain” selected.) This was more complicated for me because I was using multiple domains. If you are working with a single domain and are already logged in with credentials that allow you to add users to Active Directory, then the default values presented by this interface are what you would want.

After you supply the credentials in Figure 3, you are asked to supply the base name (see Figure 4) for test user accounts, for example “Test”, and starting and ending numbers (e.g.: 001 through 098) to form names such as “Test001”, “Test002”, and so on. The module gives you an option of adding these new user accounts directly to the domain or as a newly created sub-OU[11]. Also, you can add these test user accounts to additional user groups as part of the account creation, and set the "password never expires" setting. Note that you need to go to AD directly first to make a test user group before creating the users if you want them added to the group as part of the creation process.


DeNamiK - Virtual User creation module
User information
LDAP://DC=tmurgent,DC=int
Search...
Terminal Service Information
Organizational Units information
Profile
Create Virtual Users
Exit

Username Test from 0001 to 0001

Password ********

Domain tmurgent.int

☐ User profile path

☑ Password never expires

Add the users to the following groups...

☐ Distributed COM Users
☐ DnsAdmins
☐ DnsUpdateProxy
☐ Domain Admins
☐ Domain Computers
☐ Domain Controllers
☐ Domain Guests
☑ Domain Users

Terminal Service Profile Path: Only available on Windows 2003

Terminal Service Home Path: Only available on Windows 2003

Terminal Service Home Drive — Allow Logon ☑

☑ Save as Load Profile

☑ Activate profile

Profile name tmurgent

Set Citrix IP address 03Citrix4

Published Application Pagemania_100_65

LDAP://tmurgent-ts/DC=tmurgent,DC=int

☑ Create new 'sub' OU DeNamiK

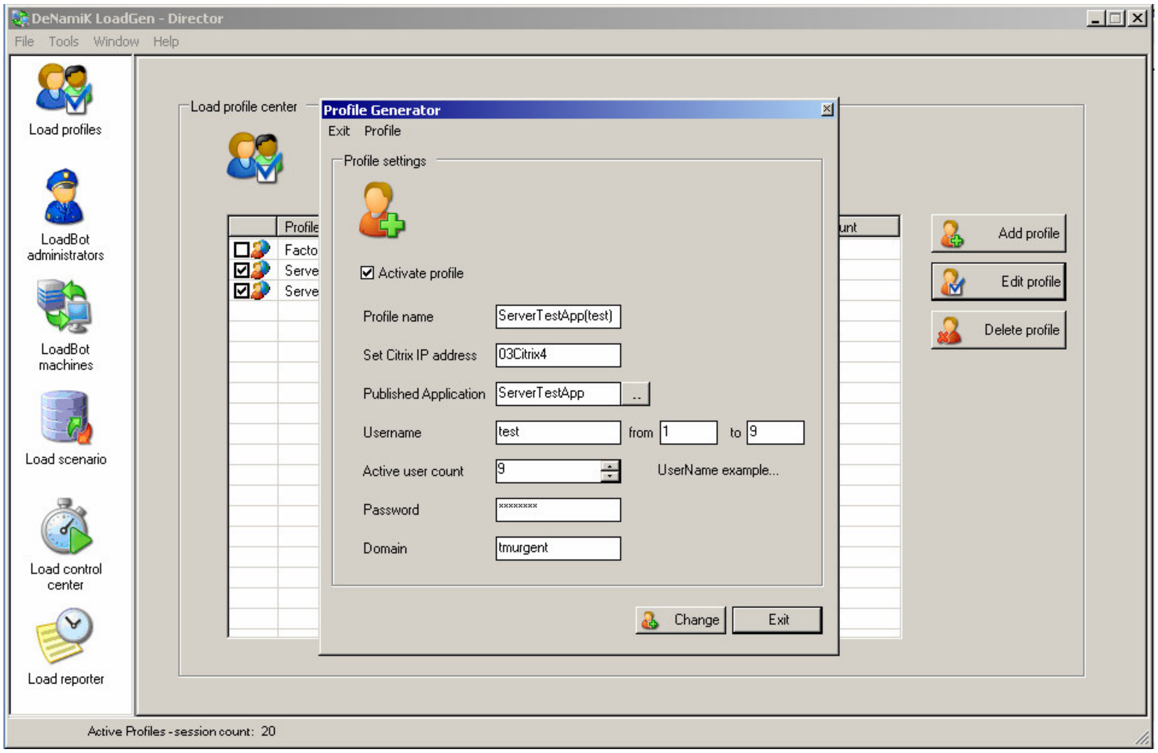
Figure 4 - Create User Accounts and "Profile".

In addition to generating these AD accounts, this module will also create the "Profile" (bottom left of Figure 4). The GUI works nicely with the Citrix web interface so that you can pick the published application name from the published list by clicking on the ".." button.

[11] OU: Organizational Unit. This is different than being a member of a group and can be skipped unless your AD administrator is, shall we will call it "anal retentive"?

When you already have AD Users

If you already have test accounts in your Active Directory (and they are reasonably named) you can use them instead. It may also be easier to do the Citrix Application Publishing if all the test user accounts are part of an AD user Group. To use existing accounts, skip the "Virtual User Creation Module" and jump right into the "Director" Module. Then select the "Load Profiles" (Figure 5) and add a new profile associating those user accounts with a published application.



☑ Activate profile

Profile name: ServerTestApp(test)

Set Citrix IP address: 03Citrix4

Published Application: ServerTestApp

Username: test from 1 to 9

Active user count: 9

Password: ********

Domain: tmurgent

Active Profiles - session count: 20

Figure 5 - Creating a "Profile" with existing User Accounts

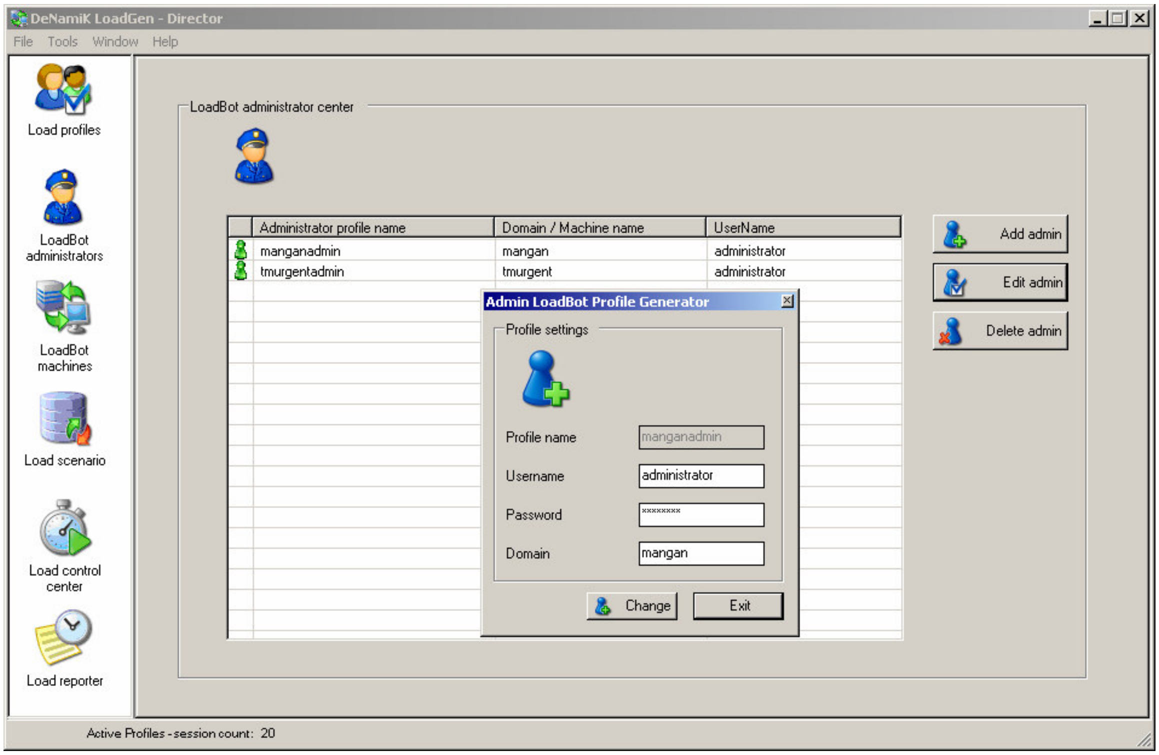
These Profiles are selectable later on in the testing process. Most situations probably only need a single Profile, but you can have multiple. In my case, because I had two server farms in different AD domains I created two different Profiles, one with users in different domains that I would use with the appropriate farm. I could also create profiles for different scenarios using the same users – a CPU memory scenario profile in one case, a profile using memory or I/O intensive apps in another.

4. Verify the “Profiles”

The **Tools** menu of the Directory allows you to select users from the profile and run a test of the published application. Unlike the load tests performed by the Bots, here you can see the published app run to verify correctness.

5. Define LoadBot Administrators

A LoadBot Administrator is the set of credentials (AD or local admin) needed to install and run the LoadBot software on client machines. This is the magic that makes it so you don’t have to visit all those client machines! Figure 6 shows the definition and parameters associated with a LoadBot Administrator.



Administrator profile name	Domain / Machine name	UserName
manganadmin	mangan	administrator
tmurgentadmin	tmurgent	administrator

Profile name: manganadmin

Username: administrator

Password: ********

Domain: mangan

Active Profiles - session count: 20

Figure 6 - LoadBot Admin Definition

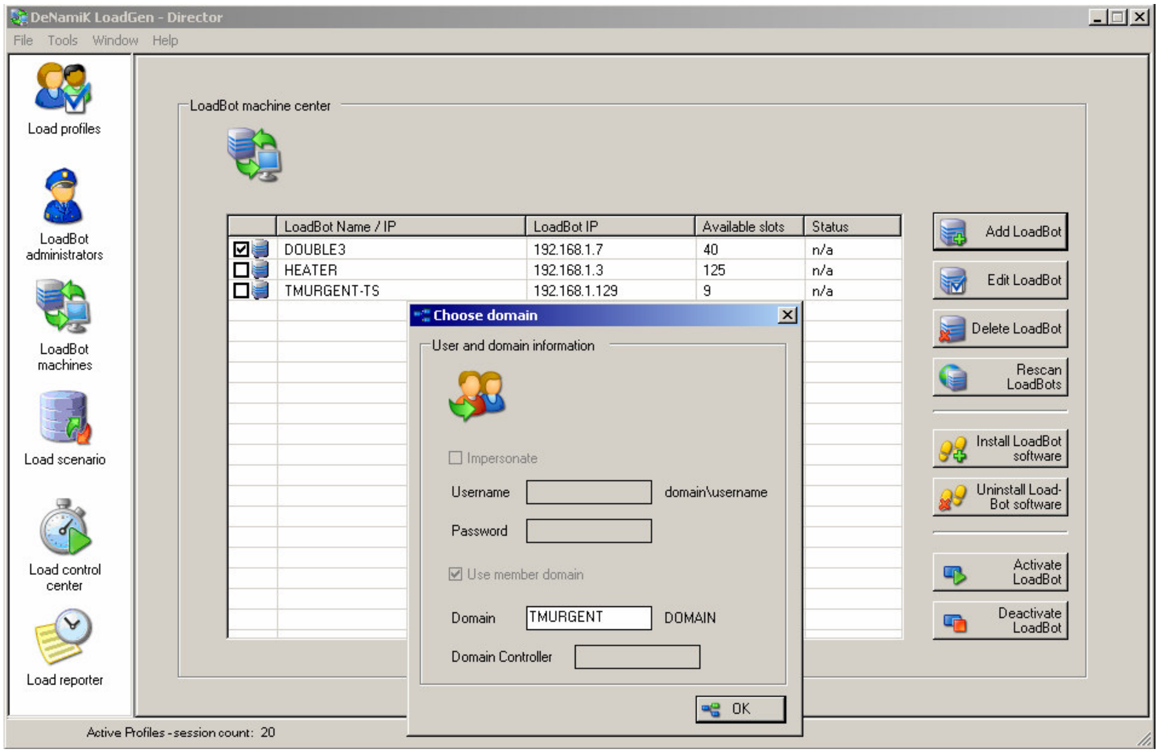
Again, a single domain scenario will only need a single LoadBot Administrator; as I was using two domains in the test I needed two LoadBot Administrators, one with credentials to install software on client machines in each domain.

6. Select LoadBot Client Machines and get LoadBot Installed

LoadBot Machines are the Client PCs used to launch published applications from as part of the test. The strength of DeNamiK over any scripting you and I might do on our own is it's ability to harness the power of multiple PCs around your lab into a single organized test.

The LoadBots themselves are software that is installed (via this dialog) on the target machine. When you run a test scenario later on, the console directs each of these LoadBots to do its part.

The console software integrates well with the network to help you locate these machines. Once you give it a Domain (or Workgroup) name, as in Figure 7, it locates member PCs and verifies the OS installed.



	LoadBot Name / IP	LoadBot IP	Available slots	Status
☑	DOUBLE3	192.168.1.7	40	n/a
□	HEATER	192.168.1.3	125	n/a
□	TMURGENT-TS	192.168.1.129	9	n/a

□ Impersonate

Username ____ domain\username

Password ____

☑ Use member domain

Domain TMURGENT DOMAIN

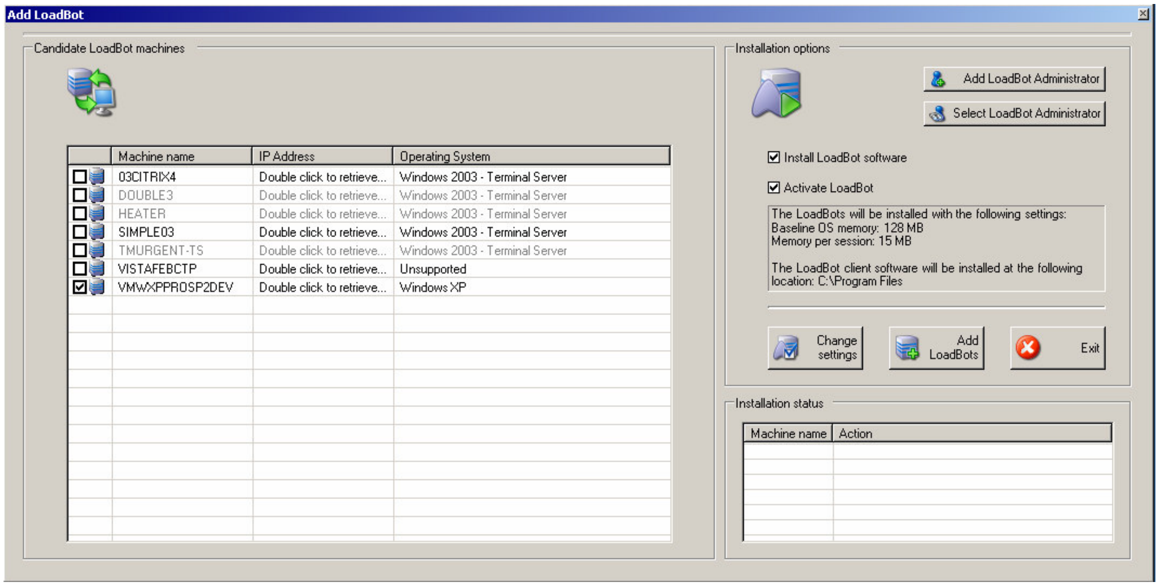
Domain Controller ____

Active Profiles - session count: 20

Figure 7 - Finding Clients for LoadBots

The found machines are presented in a list (see Figure 8) from which you can select machines. This search function found everything visible in the domain of the network neighborhood, including a non domain machine that was in a workgroup of the same name. This shows that the search is using network interfaces rather than AD. As it also detects the OS, you can see that it detected that the machine with the Windows Vista Beta is unsupported (at that time). I was unable to work with any Workgroup clients in my tests, which probably was related to defining the LoadBot Administrator properly – which I seemed unable to do for Workgroups.

After you highlight one or more machines, choose the "LoadBot Administrator", and click on the "add LoadBots" button to get the LoadBot software installed onto the LoadBot machine. Software updates can be centrally distributed from this interface as well.



	Machine name	IP Address	Operating System
☐	03CITRIX4	Double click to retrieve...	Windows 2003 - Terminal Server
☐	DOUBLE3	Double click to retrieve...	Windows 2003 - Terminal Server
☐	HEATER	Double click to retrieve...	Windows 2003 - Terminal Server
☐	SIMPLE03	Double click to retrieve...	Windows 2003 - Terminal Server
☐	TMURGENT-TS	Double click to retrieve...	Windows 2003 - Terminal Server
☐	VISTAFEBCTP	Double click to retrieve...	Unsupported
☑	VMWXPPROSP2DEV	Double click to retrieve...	Windows XP

☑ Install LoadBot software

☑ Activate LoadBot

The LoadBots will be installed with the following settings:
Baseline OS memory: 128 MB
Memory per session: 15 MB

The LoadBot client software will be installed at the following location: C:\Program Files

Machine name	Action

Figure 8 - Installing the LoadBots

As an alternative to this discovery (search) method, you can also import a CSV formatted list (via the tools menu), such as from your SMS or Altiris consoles if you have those. You still need to assign the LoadBot Administrator and Install/Activate the LoadBot machines via this interface.

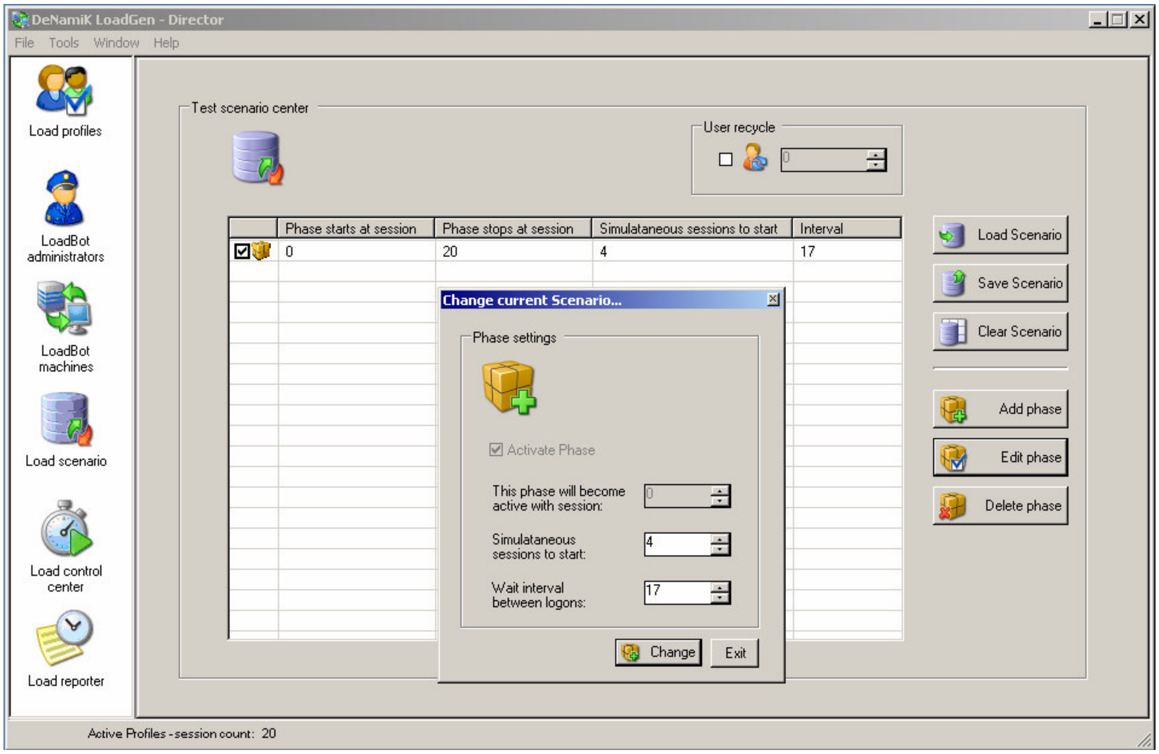
7. Verify LoadBot Machines

Right clicking on a LoadBot Machine (see window in background in Figure 7) you can verify the state of the LoadBot. This causes the LoadGen Director to contact the LoadBot service on the machine and ensure it is running and ready to accept request to run a test.

This is a step you have to do before you run any test scenario. For example, tomorrow you might want to run a test and if a LoadBot machine was reset the Bot will not be active. Running this verification will not only detect the situation but correct it.

8. Define a Scenario

A test "Scenario" consists of a set of "Phases", each of which defines a Combination of a set of "Profiles" that are run in some manner. It isn't obvious in the interface, as shown in Figure 9, but a phase is tied to the active “Profiles”. Those Profiles that are checked in the Load Profile Section of the Director are “active”.



User recycle: □ 0

	Phase starts at session	Phase stops at session	Simulataneous sessions to start	Interval
☑	0	20	4	17

☑ Activate Phase

This phase will become active with session: 0

Simulataneous sessions to start: 4

Wait interval between logons: 17

Active Profiles - session count: 20

Figure 9 - Edit a Scenario Phase

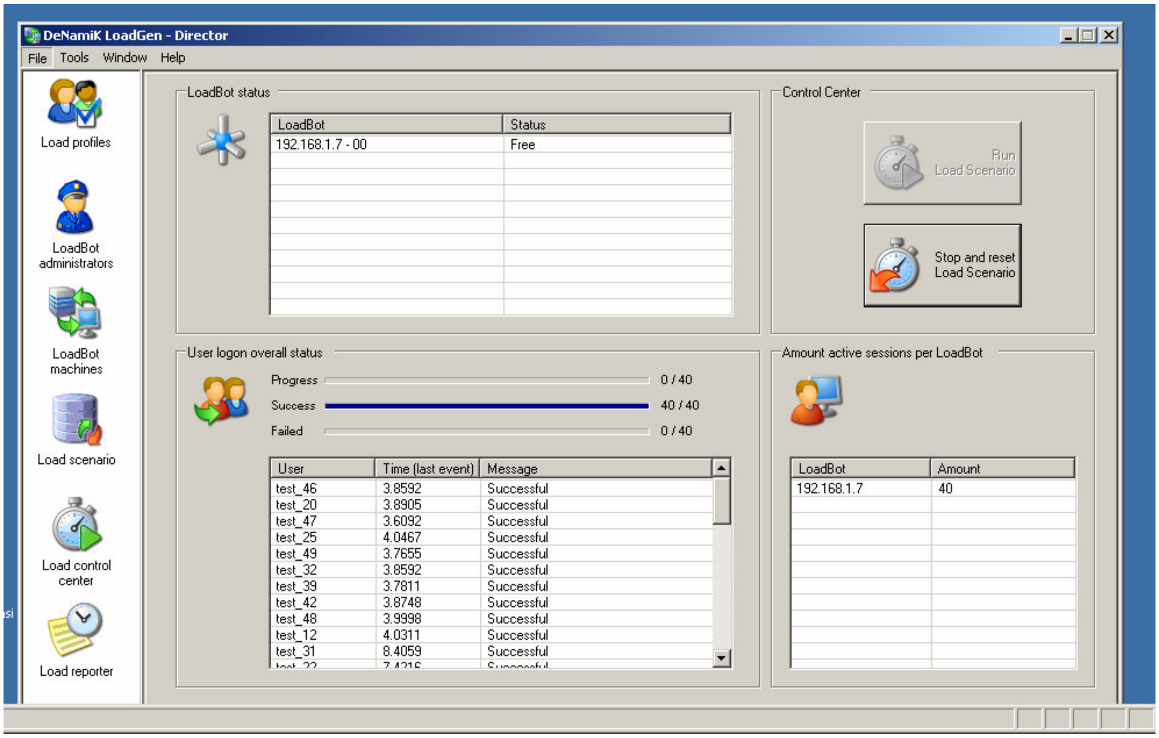
You define how many user sessions to launch at a time and the time to wait between launchings. That latter time appears to be between when the application start is confirmed to the start of attempting to launch the next set, so you can get more overlapping sessions if the test sessions run longer than your wait period.

So with these multiple phases I can set up a profile for 100 users, and then phase in the application launching. The first 10 might launch one at a time with a 20 second interval; Phase 2 might cover the next 20 launching 2 at a time; Phase 7 might launch 10 at a time with a 2 second interval!

At present I find this interface a little clumsy to use, however it is clear that they have aim at a very flexible solution that will suit anyone’s needs. Given more time and polish this has all the signs of being an excellent product! Oh, and don’t forget to use that Save button or you will have to re-enter the phases again if you shut down.

9. Run Scenario(s)

The "Load Control Center" is the control and monitoring panel for actually running the load tests. You go back to the "Load Profiles", "LoadBot Machines", and "Load Scenarios" screens and use the check boxes to select what your test will consist of. Then you go to the Load Control Center (see Figure 10) and click on the Run button.



LoadBot	Status
192.168.1.7 - 00	Free

Progress 0 / 40

Success 40 / 40

Failed 0 / 40

User	Time (last event)	Message
test_46	3.8592	Successful
test_20	3.8905	Successful
test_47	3.6092	Successful
test_25	4.0467	Successful
test_49	3.7655	Successful
test_32	3.8592	Successful
test_39	3.7811	Successful
test_42	3.8748	Successful
test_48	3.9998	Successful
test_12	4.0311	Successful
test_31	8.4059	Successful
test_22	7.4216	Successful

LoadBot	Amount
192.168.1.7	40

Figure 10 - Running a Scenario

At this point the Director Control Center works with the background service (DeNamiK.exe) on the client over TCP port 8000 to control the tests. DNKLoadBot.exe on the client talks TCP on ports 4840[12] and 4841.

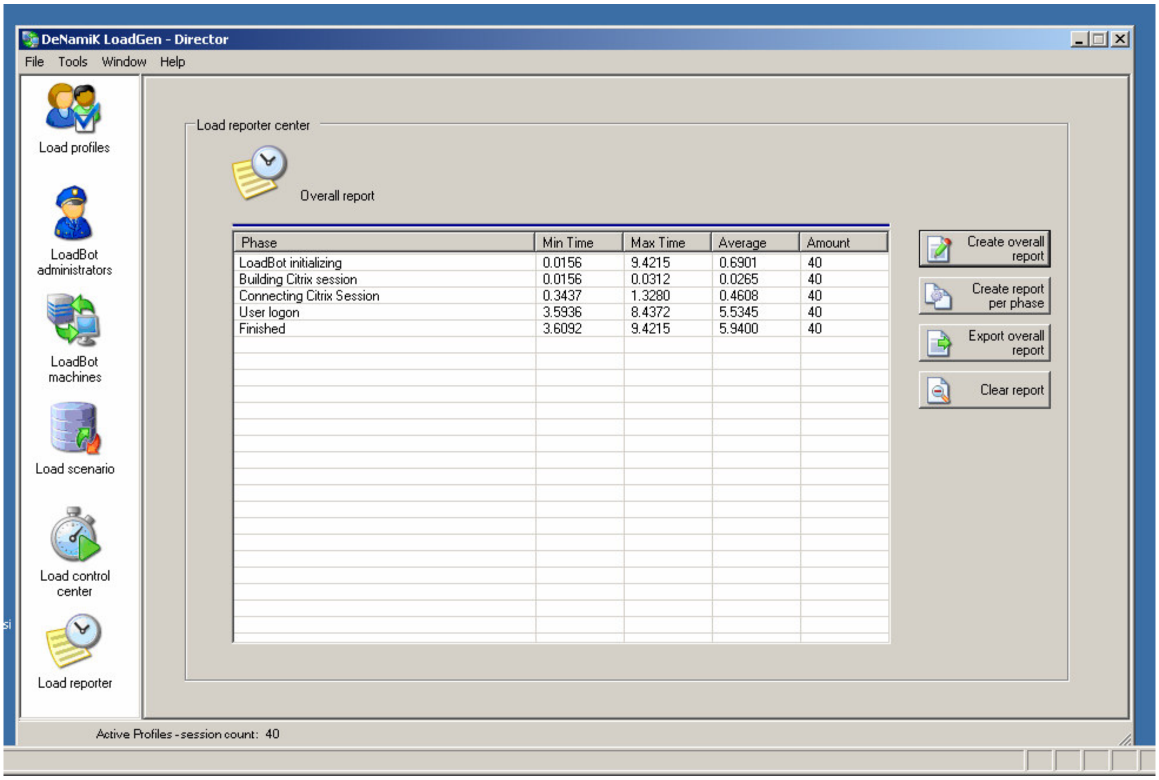
The control center is your only indication of progress through the test. The "User logon overall status" monitor shows you tests that have been attempted to be launched. "Successful" only means that the LoadBot assigned has successfully logged in the user to the Citrix Server and launched the application. No attempt at verifying correctness of the application is attempted.

[12] Odd trivia item. Port 4840 is also used in the Linux world for a "Pseudo Random Number Generator Daemon" associated with SSL. Should not be a conflict.

The director seems to choose individual user/apps from the profiles randomly, and I am yet undecided if that is a good thing. Sometimes I want randomness in my testing and sometimes I want a repeatable test.

10. Report Results

In the Beta, results from the Load reporter were limited to recording the time periods up to and including the application launch, but not the application termination. I didn't have time to figure out if this was per design or a bug. As seen in Figure 11, the report center would show the minimum, maximum, and average times for the test, with an option to report on individual phases of the test scenario.



Phase	Min Time	Max Time	Average	Amount
LoadBot initializing	0.0156	9.4215	0.6901	40
Building Citrix session	0.0156	0.0312	0.0265	40
Connecting Citrix Session	0.3437	1.3280	0.4608	40
User logon	3.5936	8.4372	5.5345	40
Finished	3.6092	9.4215	5.9400	40

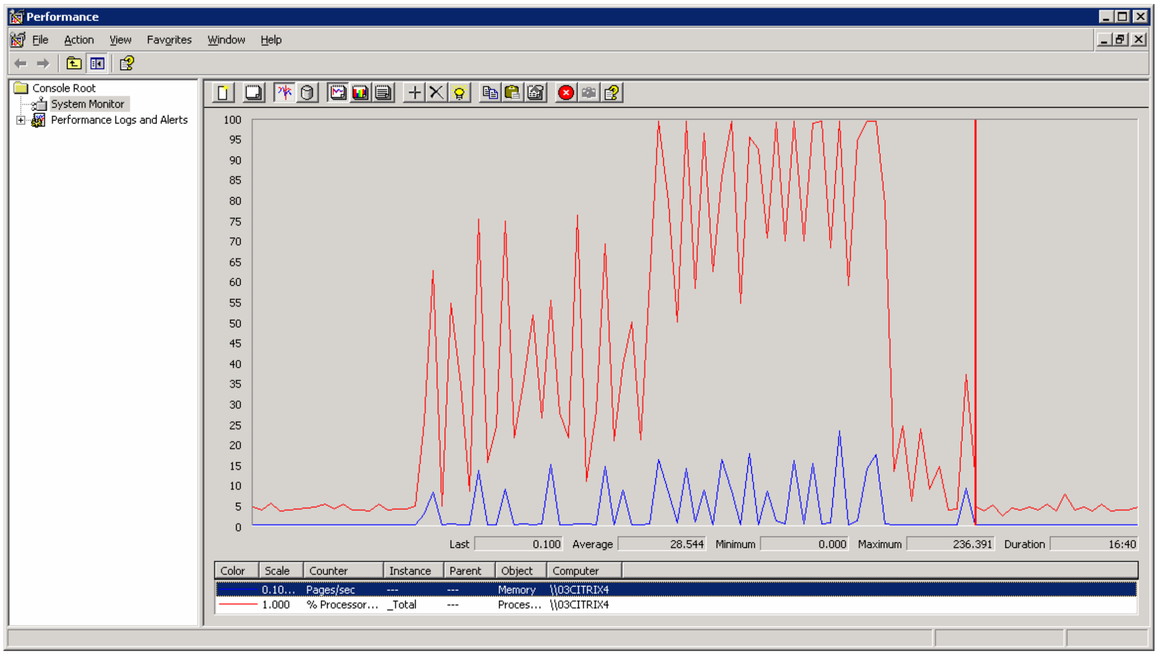
Active Profiles - session count: 40

Figure 11 - Reports Interface

I believe that the content of this report, although in raw tabular form, is a little weak today to use LoadGen as a platform to measure "Perceived Performance[13]" results (which I am so fond of) under differing load conditions, but with a little more work I think they can easily get there.

[13] See the White Paper section at www.tmurgent.com for information about Perceived Performance.

For looking at the computational performance of the server under test, the Windows Performance Monitor, or other such server management tool (including the reporting end of Scapa or LoadRunner), usually may be used to provide you reasonable statistics from the loaded server as well. Figure 12 shows the results of a phased test that I ran in action.



Last 0.100 Average 28.544 Minimum 0.000 Maximum 236.391 Duration 16:40

Color	Scale	Counter	Instance	Parent	Object	Computer
🟦	0.10...	Pages/sec	---	---	Memory	\\03CITRIX4
🟥	1.000	% Processor...	_Total	---	Proces...	\\03CITRIX4

Figure 12- Performance Monitor Results

In that example, the red line is CPU and you can clearly see the more lightly loaded first phase of the test on the left, followed by the second phase which was more heavily loaded.

Summary

Architecturally, LoadGen is designed quite well. It seeks to automate everything you need to have a first class load test environment, and is mostly there.

Because the product I tested was the first Beta, it lacks in Documentation, and a bit of "flow" in the user interface. Names of menus or buttons need a bit of polish, and I wonder if portions of the console would not be better suited with "Wizards" that are so popular in GUIs today.

Some of the integrations are still rough – again because this is an early Beta and I tend to test the outer edges of things. Diagnostics for when things go wrong are lacking as well. I would expect those rough edges to get smoothed out soon.

The additional features that I would want to see added to the product seem to be on the roadmap that was shared with me. That would include RDP support, better run-time monitoring/diagnostics, reporting improvements, and a scripting solution that would send keystrokes from the client machines.

While exact pricing is not available, DeNamiK has indicated their pricing model, which is exciting. The model is based on a per-"test user" basis, and the first 150 test user licenses will be free. They really want this to replace the CTSK for folks that need a simple low-end test. For larger tests you need to purchase licenses, which is only fair. In addition to purchasing a permanent license, their may also be a "leasing" model that will allow you to lease test user licenses for a period of time – appropriate for sizing an upgrade project to an existing farm.

It was surprising for the first Beta of a new product to automate so much of the testing in my chaotic multi-faceted lab. Congratulations especially go to DeNamiK for handling a mixed 2000 and 2003 Active Directory Domain without having to work around it. DeNamiK's goal of replacing the CTSK as an affordable test platform with something usable and scalable for massive testing is clearly met with this product. When the final release becomes available I expect it will be a welcome addition to the lab.

About TMurgent White Papers

At TMurgent, we provide information to the Terminal Server community, as well as working with Software vendors with products for this space. Please visit our website at www.tmurgent.com for more information about the strange things we do.