

SCSI TOOLBOX, LLC

Remote Manufacturing Engine

Contents

Overview	3
Requirements.....	3
Pre-Setup	4
Running RME.....	6
STEP 1: Select a computer to execute a script on.....	7
STEP 2: Discover the devices on the selected computer	7
STEP 3: Select a script to run.....	7
STEP 4: Build your list of devices you wish to test.....	8
STEP 5: Execute the script.....	9
STEP 6: Get runtime status of the tests running.....	9

Overview

The Remote Manufacturing Engine is a manufacturing tool that executes tests on remote machines unattended. The available tests are identical to all the tests available in the SCSI Toolbox's Disk Manufacturing Module (DMM). From a Main Console on a server, the technician can select which client computer testing will be executed on, which devices need to be tested, and what type of testing is to be done. With the simple clicks of the mouse, the testing is launched on the client computer.

On the unattended client computers, RME has an engine, the Disk Manufacturing Engine (DME). The DME accepts a script, which are defined in DMM, and a list of devices to test. Silently, DME executes the script as if DMM were running on the client machine! While the script is being executed, DME stores status information (success, failure, ...) concerning each of the tests in the script, and this status information is remotely retrieved from the Main Console on the server machine, giving the user real-time progress and status information on the tests.

Requirements

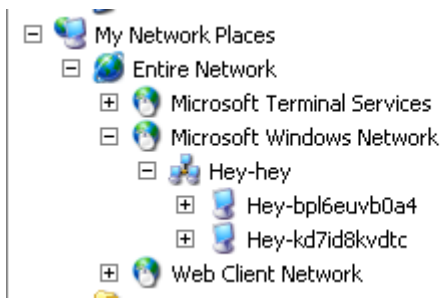
The Main Console will have the capability to:

- Select a computer to begin testing on
- Define which devices DME should begin testing on. The definition of which devices need to be tested can be defined as being all devices on a HBA, or as a list of particular devices. By way of an example, it will be capable to do testing on HBAs 2, 3, and 7 and Device with address 4:2:0 (simultaneously)
- Select which script is to be executed by DME
- The capability to get real-time status information on the testing being executed on a particular computer
- The capability to Stop testing on a particular computer

Pre-Setup

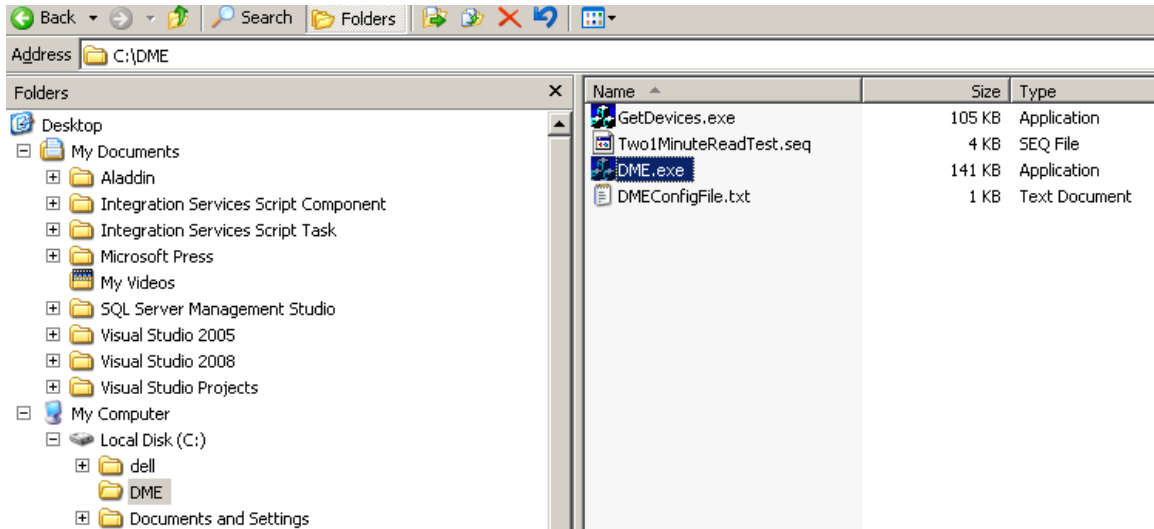
Prior to running the **Remote Manufacturing Engine (RME)**, the user must accomplish the following:

- Have a network set up with one or more client machines, each “visible” to each other. By way of an example, one may have a “server” machine and one “client” machine. The pic below is helpful:



In the pic above, the server is “Hey-kd7id8kvdtc” and the client is “Hey-bpl6euv60a4”. It is very important that you have the computers visible on the Network. If “My Network Places” => “Entire Network” => “Microsoft Windows Network” does not show your computers, **RME** will NOT function!

- The following files must be stored on the client/remote machine: DME.exe, GetDevices.exe, the script you want to execute, DMEConfigFile.txt. The picture below shows the files needed on your client/remote machine:



DME.exe and GetDevices.exe are applications supplied by SCSIToolbox. The script you want to execute is any script that you create in our Disk Manufacturing Module (DMM) – SCSIToolbox does NOT supply this script, it is a script you create on your server machine. And finally the file DMEConfigFile.txt is a text file that contains the names of all the computers on your network. Here is an example DMEConfigFile.txt:

```

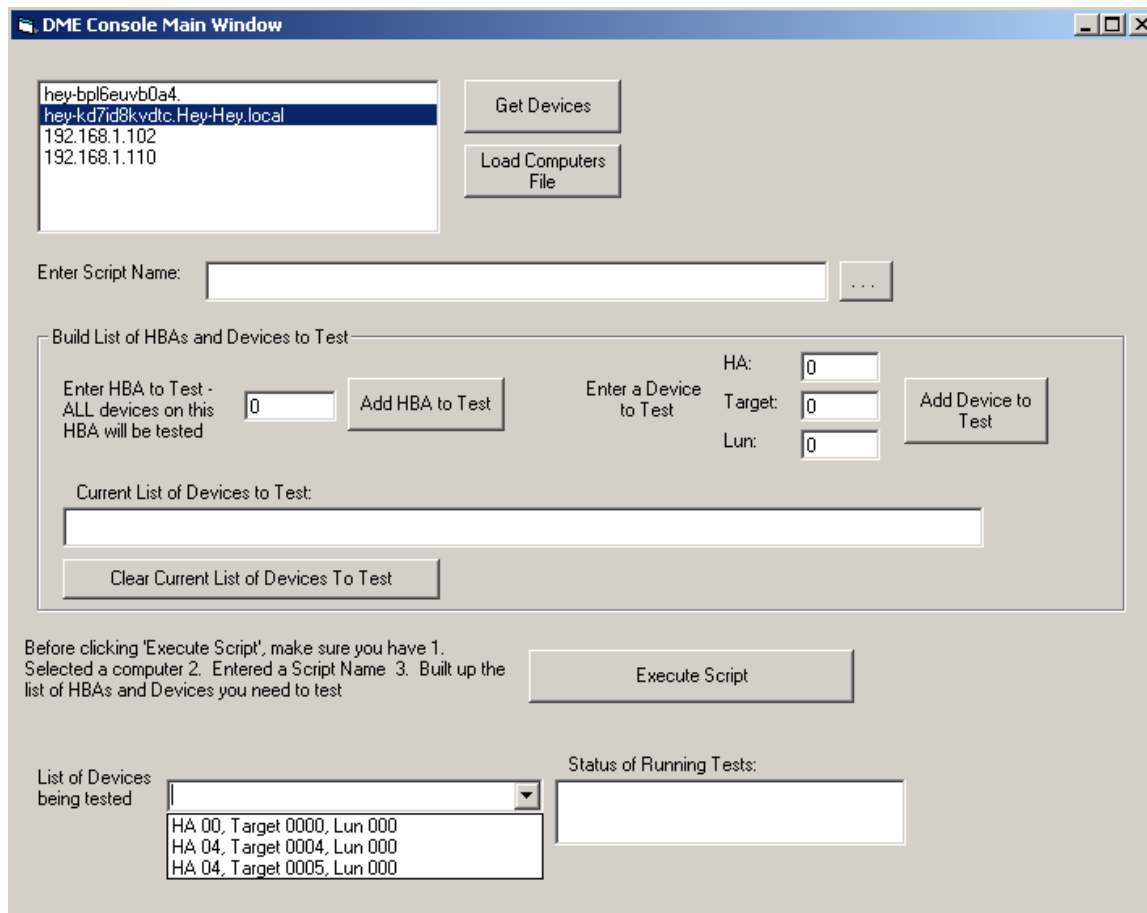
DMEConfigFile.txt - Notepad
File Edit Format View Help
hey-bp16euvb0a4.
hey-kd71d8kvdtc.Hey-Hey.1oca1
192.168.1.102
192.168.1.110
:

```

In the above screen capture, we have 4 connected computers. Note you can list the computers by their IP Address

Running RME

The Main Window for **RME** is displayed in the picture below:



To utilize **RME**, you basically work from “Top of the Screen” to the “Bottom of the Screen”. If you go from top to bottom of the Main Window for **RME**, you will do the following tasks in order:

- Select a computer to execute a script on
- Discover the devices on the selected computer
- Select a script to run
- Build your list of devices you wish to test
- Execute the script
- Get runtime status of the tests running

© Copyright 2010 SCSI Toolbox LLC

Sales: 720.249.2641

General: 303.972.2072

Let's go over each of the six items listed above.

STEP 1: Select a computer to execute a script on

RME will automatically load your DMEConfigFile.txt. If it cannot find this file, you will need to click the "Load Computers File" to browse to the location of DMEConfigFile.txt. Then select a computer

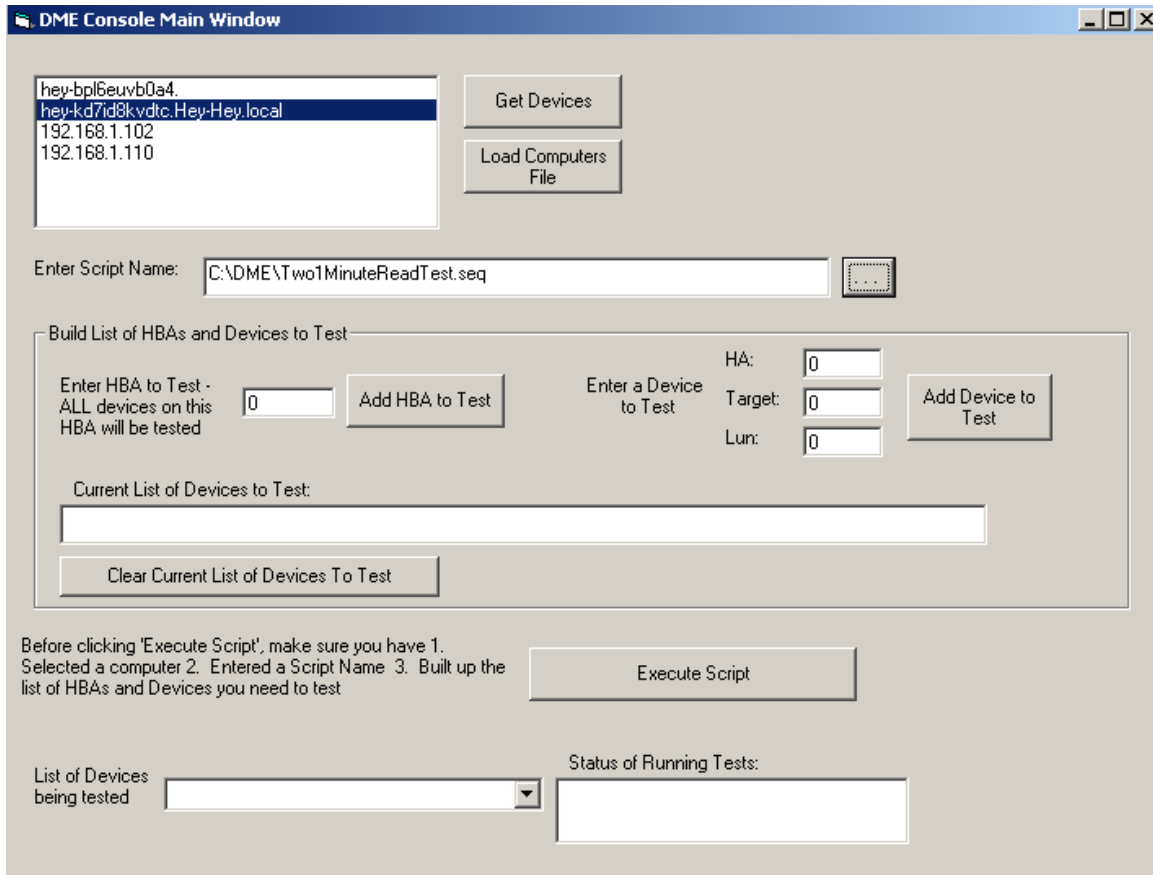
STEP 2: Discover the devices on the selected computer

Click the "Get Devices" button – **RME** will automatically discover the devices on the selected computer and then add these devices to a combo-box (see the label "List of Devices")

STEP 3: Select a script to run

Click the "Browse" button next to the edit box labeled "Enter Script Name" and browse to the script you wish to execute on the selected computer.

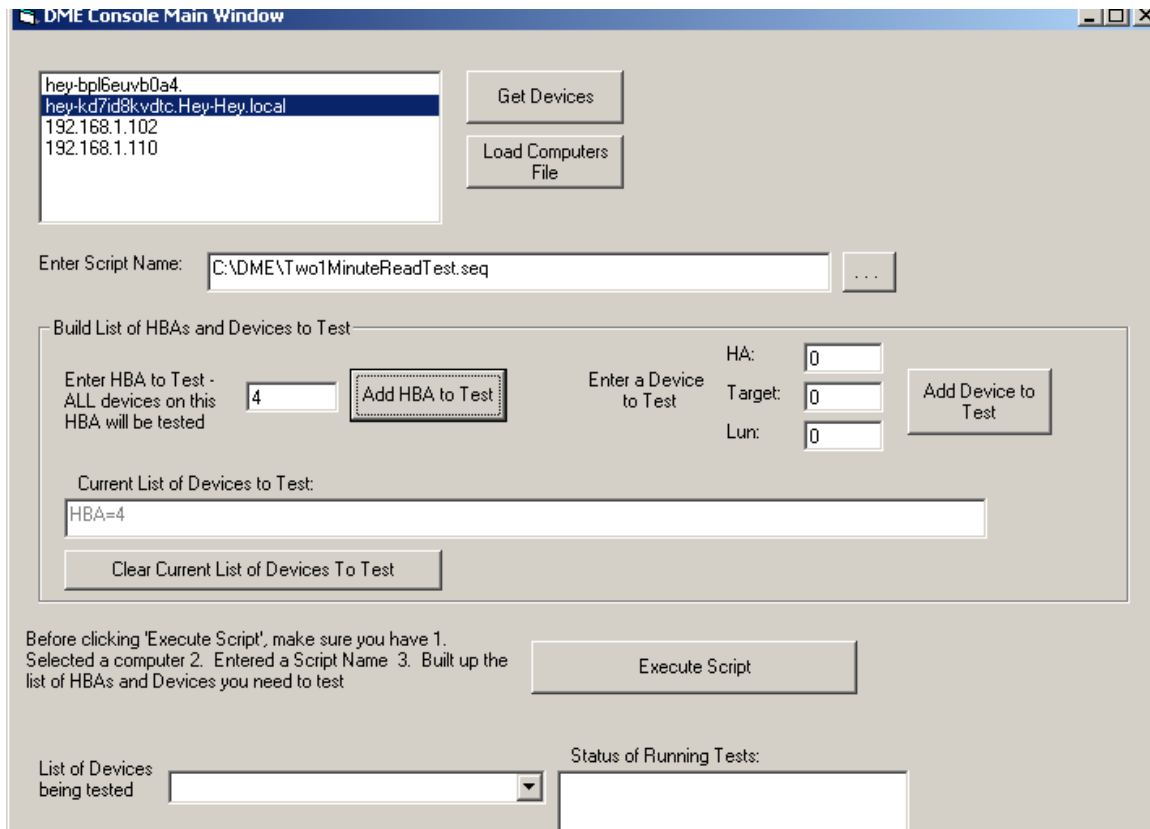
At this point your Main Window will look something like the screenshot below:



STEP 4: Build your list of devices you wish to test

You can enter a list of HBAs to test, and/or a list of individual devices to test. By way of an example, suppose you wanted to test all devices on HBA 4. You would enter “4” into the edit box labeled “Enter HBA to Test” and then click the “Add HBA to Test” button.

Your screen will now look like this:



Notice that the edit box labeled “Current List of Devices to Test” now has the string “HBA=4”

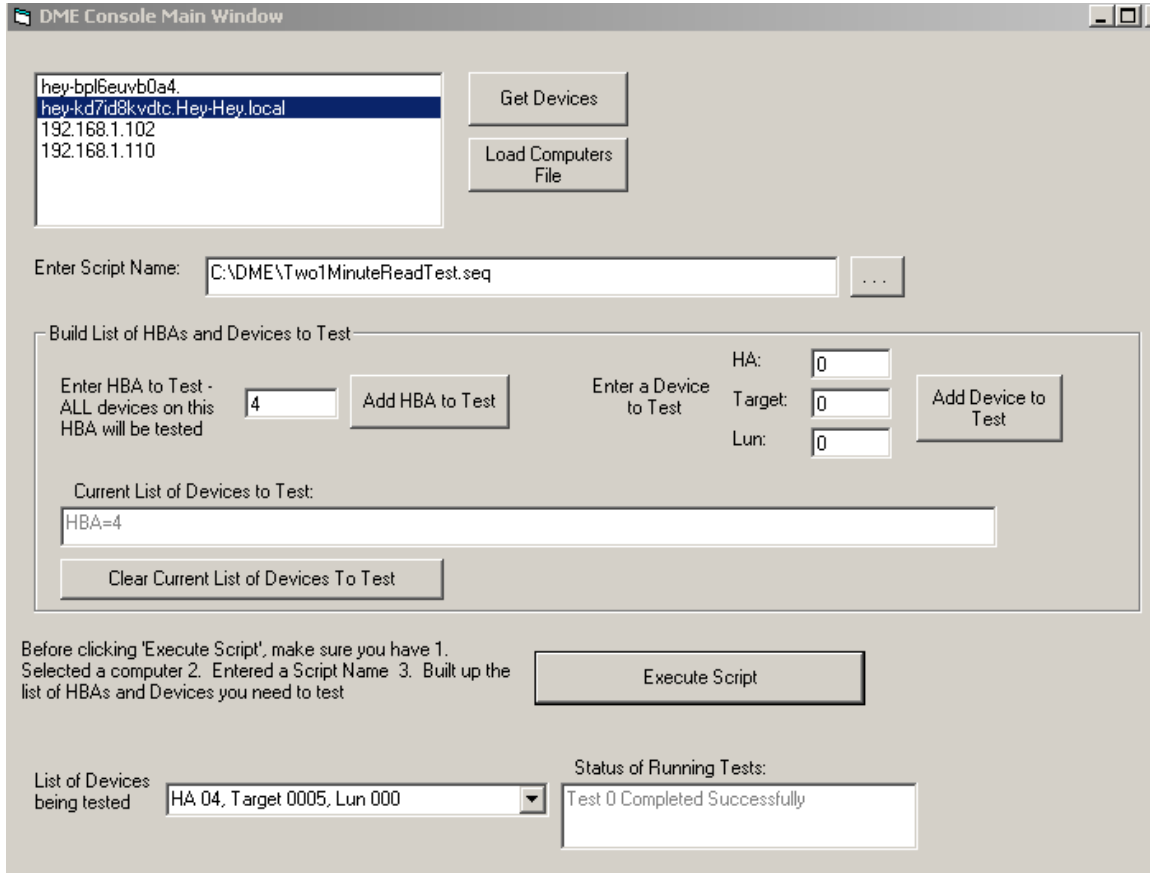
STEP 5: Execute the script

To begin testing, click the “Execute Script” button

STEP 6: Get runtime status of the tests running

To get status of a particular device, select an entry in the combo-box labeled “List of Devices”. The status of the testing being executed on that device will automatically be updated. In the screenshot

below, we have selected device HA 4, Target 5, LUN 0 and the status is shown as “Test 0 Completed Successfully”



As testing proceeds from Test 0 to Test 1, to Test 2, and so on, the status will automatically be updated (there is no button to click to get status, it is done automatically at a periodic interval)