White paper

Using BurnInTest in a production line environment

Abstract:
This white paper describes how BurnInTest 7.0 can be used in a production line environment. This paper is targeted at companies that manufacture, integrate or repair a large number of PC’s on a daily basis. Companies or individuals that run smaller operations may also benefit from some of the information in this document however.
Contents

Overview................................................................................................................................. 3
Configuration files .................................................................................................................... 4
What are configuration files................................................................................................. 4
How are configuration files created ..................................................................................... 4
How are configuration files used .......................................................................................... 4
Selecting a test configuration ............................................................................................... 6
Carrying out tasks before the test run .................................................................................. 7
Carrying out tasks after the test run ...................................................................................... 8
Example Autostop actions ..................................................................................................... 8
Report information - (Machine identification) ........................................................................ 9
General report information .................................................................................................... 9
Certificate report information ................................................................................................ 9
Test reporting - logging to disk .............................................................................................. 10
Installation and execution options ....................................................................................... 12
Standard Install ..................................................................................................................... 12
Run from a removable drive or network drive ....................................................................... 12
Externally controlling BurnInTest ......................................................................................... 16
Running BurnInTest on a System without an operating system ........................................... 17
Restricting a user from changing the tests selected .............................................................. 18
Rebooter .................................................................................................................................. 19
Reboot cycling ....................................................................................................................... 20
Sleeper ..................................................................................................................................... 22
Test hardware - Loopback plugs .......................................................................................... 23
Testing specialized hardware .............................................................................................. 24
Un-installer command line arguments ................................................................................. 25
Feedback ................................................................................................................................. 26
Contacting PassMark Software ............................................................................................. 26
Appendix 1: Command line arguments ................................................................................ 27
Appendix 2: Scripting ............................................................................................................. 30
Appendix 3 – Example reports .............................................................................................. 50
Text report ............................................................................................................................. 50
HTML report .......................................................................................................................... 53
BurnInTest Certificate .......................................................................................................... 59
Overview

This white paper describes how BurnInTest can be used in a production line environment. The main aim in this environment is to devise a system to increase the level of automation and maintain adequate records of the testing carried out.

This paper is applicable to both the Standard and Professional Windows editions of BurnInTest. More testing will be possible with the Professional edition however. This document does not cover the Linux version of BurnInTest.
Configuration files

What are configuration files
A configuration file contains all the parameters available in the test duty cycles and test preferences windows. That is to say it contains information about which tests to perform, the duty cycle setting for each test and other preferences, such as drive letters and device names.

Configuration files end in the extension "bitcfg. For example the current configuration is stored automatically in the file ‘LastUsed.bitcfg’, which is located in the BurnInTest installation directory.

Configuration files aid with the automation of the testing process and also allow the maintenance of several different test configurations. For example a configuration file could be created for each different model of computer to be tested.

By saving a particular configuration to a file, you can avoid resetting all the parameters each time BurnInTest is used.

How are configuration files created
A configuration file can be created by first selecting all the required settings in the test duty cycles and test preferences windows, then use the menu option, ‘Save Test Configuration As’. This will create a new configuration file in the location selected by the user.

After the configuration files have been created they can be transferred between computers where BurnInTest is installed.

How are configuration files used
Files can be used in three ways
1. They can be loaded from the command line.
2. They can be loaded by using the menu option, ‘Load Test Configuration’.

Figure 1 – Duty cycle window. These settings along with other preferences are saved in configuration files.
3. They can be loaded by a BurnInTest script file. A script file can load multiple test configurations during the one test run.

For the purpose of test automation options 2 and 3 are of most interest. (see ‘Command line arguments’ and ‘Scripting’ sections below).
Selecting a test configuration

It is possible to adjust a wide range of values from the ‘test duty cycles’ and ‘test preferences’ windows. The following general tips can be used to help select the right combination of tests to perform and their intensity level.

- Only turn on tests for which there is some corresponding hardware. For example, there is obviously no point in turning on the tape drive test if there is no tape drive installed.
- If the disk drive has a fairly large capacity, you will probably need to set the duty cycle for the disk test to 100% in order to make sure the maximum amount of disk surface is tested over several cycles.
- Monitor the CPU utilization during the 1st couple of test runs. If the CPU is not running at 100% all the time, raise the duty cycle of one or more tests. Conversely, if the CPU consistently runs at 100%, lower the duty cycle of one or more tests.
- If the machine has booted to Windows there is a reasonable chance that the CPU is working reasonably reliably. Give preference to the RAM test over the CPU test when setting the duty cycle. RAM tends to fail more often than a CPU.
- Don’t be scared of pushing the machine too hard. It’s better that it fails on the production line than after you have shipped it to your customer.

You should test for at least a few hours. An absolute minimum is one complete cycle of the disk test. Leaving BurnInTest to run on the PC overnight should give the machine a good work out.
Carrying out tasks before the test run

From the Pre-Test tab in the preferences window, BurnInTest allows the user to select the option of running an external application before the test run proper starts. For instance you may want to launch a report of the previous test run before beginning the current one. The test run proper will not begin until the external application is exited.
Carrying out tasks after the test run

From the Timers tab in the preferences window, it is possible to define certain actions to take on successful or unsuccessful test completion. In each case (successful or unsuccessful completion), there are three options.

1. Stop the tests normally (this is the default behavior).
2. Stop the tests, Close BurnInTest and run a specified external application. This could be used, for example, to call an application or batch file that would clean up the hard disk after the completion of testing.
3. Reboot the PC using the current Reboot options. This option uses the PassMark Rebooter application to perform the reboot. If used in conjunction with a startup shortcut to BurnInTest, it’s possible to create a reboot cycle. See the ‘Rebooter’ section below for more details.

![Timers tab in preferences window.](image)

As each environment will be different, the function of an external batch file will vary between companies. Some possibilities are described below.

**Example Autostop actions**
- Copy the log files (see below) to a network drive
- Call an external program to sound an audible alarm
- Start another test program to run a sequence of tests
- Call a batch file to clean up the hard disk of the machine
Report information - (Machine identification)

General report information
The identity of the machine can be entered in to the ‘Machine Identity’ window, (accessible from the Edit menu in BurnInTest).

Certificate report information
The fields in this section allow the entry of customer specific information for the system under test. This information is used in the Customer Test Certificate report.

This information can be saved to a file using the ‘Save as Defaults’ button. The name of the file is always ‘machineid.dat’. This file will be saved in the BurnInTest installation directory. Each time BurnInTest starts it checks to see if this file exists on the disk and loads the contents of the file.
Test reporting - logging to disk

BurnInTest can create report files automatically. Using the setting in the preferences dialog the user can select:

- If logging to disk is on or off.
- The log file directory and base file name. Note: Log files are prefixed as described below and suffixed with the created date and time.
- The required detail level of the log file. Either a Results Summary or Normal (recommended).
- Whether a very detailed activity trace log is produced and the level. It is recommended that No trace log be specified under normal circumstances.
- The approximate size of the log (and trace) files.
- The format of the log file. Plain ASCII text, HTML web page or HTML Customer certificate.
- A prefix to the log file name, including a number of system Environment variables.
- Whether to clear all log files at the start of each test run, or to accumulate the log files over test runs.

Note: Log entries are written to disk in real time. You can also log periodic interim result reports.

The log file format can be ASCII (plain text), HTML (for the web) or an HTML Customer certificate (1 page HTML report for your customer). Log files can be opened in a text editor or a word processor application. See the example reports Appendix 1. HTML files can be opened in a browser and posted on the web. Both formats can be attached to E-Mail messages.

Figure 4 – Logging preferences window
New log/trace files will be created when BurnInTest is started and logging is turned on, or any logging options are changed (eg. Turned on, Log directory changed, format changed). If BurnInTest is restarted with scripting commands of REBOOT or REBOOTEND, entries will be concatenated to the previous log file (ie. That relate to the same script being run).

BurnInTest will automatically modify the file name selected to include the current date and time, e.g. If you use the file c:\temp\logfile.txt as your log directory then BurnInTest will create files like,

```
c:\temp\logfile_081209_113751.log
```
or
```
c:\temp\logfile_081209_113751.trace (for a trace file)
```

051209 is the date (in Year Month Day format), the 9th of December 2008 in this case. 113751 is the time (in Hour Minute Second format)

Logging to disk may be useful if your system is unstable. You could create a log, and then even if the computer crashed, you would have a record of how much testing was completed before the crash. Logging can also be used as a method of keeping quality assurance records in a manufacturing environment.
Installation and execution options

There are several different ways to install and run BurnInTest. Some of them are more suited to a mass production environment.

Standard Install
This is what happens when the standard package is installed. The files are expanded from the compressed archive and the install program is run to prompt the user for various information, like the installation directory. Files are then copied from the archive to the installation directory and some information is written to the registry to support the security software used to protect BurnInTest and the uninstallation of the package. After a standard install, uninstallation should be done from the Add / Remove programs icon in the Windows control panel. In a mass production environment this option can be time consuming.

Run from a removable drive or network drive
The installation step can be avoided entirely by running BurnInTest directly from a CD, USB drive or Network drive. There are some issues with each method that bear consideration.

- Running from a CD
  Normally BurnInTest uses the BurnInTest installation directory to store log files and configuration settings. Running BurnInTest from a read only CD will mean that it will not be possible to create / update these files. To run BurnInTest from a CD when no Operating System is installed, please see the section on “Running BurnInTest from a bootable CD”

- Running from a USB Key drive
  This method is recommended over running BurnInTest from CD as configuration and log files can be saved.

- Running from a Network drive
  This can be a viable option but if multiple instances of BurnInTest are started from the same network directory at the same time, there is a risk that configuration and log files will be overwritten by the various different instances. BurnInTest also uses a custom driver for certain purposes, such as accessing Parallel ports. BurnInTest will not have sufficient privileges to install the driver from a network.

- Running from a Floppy disk
  From V5.0, BurnInTest is too large to run be from a floppy.

Examples

Example 1: Use the following process to run BurnInTest from a CD or USB key drive.

It is possible to install BurnInTest Professional onto a USB drive or CD/DVD/BD such that no installation is required on the test system. This can be useful in a number of scenarios, such as field staff testing PC's without installing BurnInTest on the test system.

When running BurnInTest this way, there will be no files left on the PC after BurnInTest has finished. The procedure is the same for optical disks, but as it is not writeable for log files, logging should be either turned off or specified to be on a writeable drive. You can also get more sophisticated by having multiple test configurations, there is some information about this in the help file.

When BurnInTest is run from a removable drive when installed in this way, the default directory for the users files (like reports and the configuration file) is the BurnInTest directory, rather than the normal default directory of the users Documents directory. The command line parameter “/p” required in V5.x in this scenario is now turned on automatically, and is no longer required for a USB drive installation.

Installing BurnInTest to a USB drive
This installation process can be performed for a USB drive installation (any writable drive) using the menu option "File"->"Install BurnInTest to a USB drive". This option was added to the normal BurnInTest Professional V6.0.1002 package (and the "zip" build as required in V5.x is no longer required or available).

From the "Install BurnInTest to a USB drive" Window, you need to specify:
1) The USB drive and directory you want to install BurnInTest to. For example, "F:\BurnInTest". BurnInTest will create the directory if it does not exist.
2) The type of installation. If you have a license key, then select Licensed, otherwise select Evaluation for a trial period.
3) If you selected a "Licensed" installation type, then enter the Username/Key.

When you select install, BurnInTest will create the directory on the USB drive (e.g. F:\BurnInTest), copy all of the files from the BurnInTest directory (e.g. C:\Program Files\BurnInTest) to the USB drive (e.g. F:\BurnInTest) and install the license information onto the USB drive.

Installing BurnInTest to an optical disk

To install BurnInTest on an optical disk (CD/DVD/BD) follow the process above, but specify a writable temporary directory in step 1 (e.g. C:\BurnInTest). On completing the installation to the temporary directory, burn the created directory to the optical disk.

Example 2: An alternate example of running BurnInTest from a CD is provided below.
This example assumes the following.
- That log files need to be kept
- That each PC to be tested has a CD drive (Drive F), a USB key drive (Drive G) and no network connection.
- That the machine needs to be left in a clean state.

1. Create the required configuration files
From within BurnInTest, select the settings that suit your specific test requirements from the test duty cycle window and the test preferences window. Save this configuration, using the ‘Save Config As’ menu option. This can be repeated several times if several test configurations are required.

For this example we assume the following...
- The default ‘Stop Tests’ on testing Passed and ‘Stop Tests’ on testing Failed options are selected from the Post-test tab of the preferences window.
- Automatic logging is turned ON from the Logging tab of the preferences window.
The log file directory is, c:\burnintest
The name of the log file is, ‘bitlogfile.txt’
- The name of the saved configuration file is, 'heavyload.bitcfg'

2. Create the required batch file
Using Notepad or another text editor, create an installation batch file ('startburn.bat'), eg.

```
mkdir c:\burnintest

copy f:\*.* c:\burnintest
c:\burnintest\bit.exe /r /p heavyload.bitcfg

copy c:\burnintest\bitlogfile*. * g:\
copy c:\burnintest\error*. * g:\
del c:\burnintest\*. *
rmdir c:\burnintest
```

This batch file installs BurnInTest then runs the software with the configuration file that was prepared in advance. It then saves the log files to a USB Key drive and cleans up.
3. Create a fast install CD
Follow the steps given above to create a CD (in section ‘Installation and execution options’), but when burning the CD include the configuration files and batch file created above. Optionally a machine identification file, ‘machineid.dat’, (see above), could also be included on the CD.

4. Run the tests
To run the tests insert the CD in a machine to be tested and run the batch file.

5. Collect the results
If the test is completed without error there will be a number of log files on the USB Key drive. These can then be transferred to a central data store (along with the serial number of the machine). If the test is completed with at least one error you will see something that looks like the following.

![Figure 5 – Flashing Test failure window](image)

Example 3: Other examples of using command line arguments
There are a number of different ways to specify command line arguments and this can depend on what you are trying to achieve. The most common options for specifying a command line argument are:

1) You can create a shortcut to bit.exe and then specify the command line in the shortcut target.

   *Example 3.1: Start BurnInTest from a shortcut and run tests automatically.*

   Right click on the bit.exe file and select Create Shortcut.

   Right click on the new shortcut file and select Properties and specify a Target with the command line argument.

   "C:\BurnInTest\bit.exe" -r

   Now to run BurnInTest with these command line parameters, just double click the BurnInTest shortcut. You could also copy the shortcut to the Windows Startup folder to run BurnInTest automatically with the current default settings after Windows has booted.
2) You can open a command window, Start-R, "Cmd". This will open a command window where you can type commands directly, such as shown below.

   Example 3.2: Start BurnInTest from a command window and run a pre-defined script of tests

   Select Windows Start-R, "Cmd"

   Once the Windows command window is open, type in the commands:

   `cd C:\Program Files\BurnInTest`
   `bit.exe -s MyScript.bits`

3) You can write a batch file to execute BurnInTest with Command line arguments.

   Example 3.3: Start BurnInTest from a batch file with a pre-defined configuration file

   Create a file with an text editor (like Notepad) called Test.bat and include the lines in the file c:

   `cd "\program files\BurnInTest"`
   `bit.exe –c "heavyload.bitcfg"`

Example 4: Running BurnInTest from a network server

Install BurnInTest into a directory on the server using "File->Install BurnInTest to a USB drive". This will create a directory, e.g. \My Server\BurnInTest, and this directory will include a BurnInTest installation. When BurnInTest is installed this way, it can be run from a network server in 2 ways:

1) I can be just run directly from the server, e.g. \My Server\BurnInTest\bit.exe. In this case you won't have the privileges to install the BurnInTest device driver for the Parallel port test and to collect some system information. You can ignore this by starting BurnInTest with the "-a" command line parameter and by just not running the parallel port test.

2) A better (but more complicated) way to run from the server is to create a batch file that copies the directory e.g. \\My Server\BurnInTest to a local directory e.g. c:\burnintest, runs BurnInTest, copies the log files back to the server and then deletes the local copy of BurnInTest. A similar example to this is shown in Example 2, item 2 "Create the required batch file Using Notepad or another text editor".

The default directory for configuration files and log files is the My documents directory on the system bit.exe is running on. If you use the "-p" command line parameter then the default directory will be changed to the same directory where bit.exe is.
Externally controlling BurnInTest

Controlling BurnInTest from another application

Users can run BurnInTest from an external controlling application using the BurnInTest command line options. This can be useful if BurnInTest testing is one of several steps in the system validation process and this process is to be controlled by another application.

BurnInTest can run for a specified duration, number of cycles or run a scripted series of tests.

There is an option with the licensed version of BurnInTest to allow the controlling application to stop the BurnInTest testing and BurnInTest can be configured to exit at the end of the testing.

To allow an external application to signal BurnInTest testing to stop, BurnInTest must be started with the "-T" command line parameter. Once the tests are running the application controlling BurnInTest can signal BurnInTest to stop testing by creating a file called "BURNINTEST_STOP" in the BurnInTest program files directory or the ".\<UserName>\Documents\PassMark\BurnInTest\' directory, where  <UserName> is the administrator account that BurnInTest is run under. The application that creates the stop file must delete this file. To configure BurnInTest to exit at the end of testing, specify Test Preferences->Post-Test 'Run external application and exit' and leave the application field blank.

Controlling other applications from BurnInTest

BurnInTest can run an external application (and wait for it to complete) before running tests. This can be used for a number of reasons such as changing the test environment before testing or having an external application provide test information to BurnInTest, such as the company name, tester's name, serial number etc. See Pre-Test preferences to see how an external application can pass a sub-script of commands to BurnInTest to configure test information such as the system serial number.

BurnInTest can run external applications (and wait for it to complete) after running tests. This can be used for a number of reasons such as resetting the test environment after testing or passing results to another application to have it perform another step in the system validation process. To see how to run an application from BurnInTest after testing and what parameters (e.g. $RESULT) can be passed to this application please see Post-Test preferences.
Running BurnInTest on a System without an operating system

A bootable USB Flash Drive or optical disk can be created with Microsoft Windows 7 and BurnInTest 7.0 Professional using Microsoft WinPE 3.0. Many BurnInTest users can benefit from testing PC hardware when there is no Operating system installed, or the Operating System is inoperable. This can be useful for testing PC hardware:
1. In a production line environment,
2. That is to be shipped with Linux,
3. In a known virus free environment and
4. To try to determine the cause of corruption of an Operating System.

A separate document has been produced to assist people in setting up an environment that allows PassMark BurnInTest to be used in these situations. This document is available from the PassMark Software website: 
BurnInTest on WinPE 3.0.
Restricting a user from changing the tests selected

In a production line environment, it may be required to remove the ability of the tester to modify which tests are to be used, either to force the use of pre-defined test configurations or simply to stop tests being removed from the test setup. The option to select and de-select tests in the BurnInTest “Test setup and duty cycles” window can be disabled per test. To do this a file "BITaccess.txt" needs to be included in the BurnInTest directory to specify which checkboxes in the setup window will be disabled. Please email us to request a sample "BITaccess.txt" file.
Rebooter

PassMark Rebooter is a small freeware application, which is distributed with BurnInTest. It offers various options, to provide automated rebooting of computers.

The ‘Reboot options’ section allows the selection of Reboot types (Reboot, Shutdown, Power Off, Logoff), Force shutdown behavior (Ask to close, Force to close, Force if hung), the Delay once Rebooter is triggered before rebooting and the number of Maximum Reboots, used for during reboot cycling (see below).

The ‘Auto-login’ section provides the option of setting the system to auto login on restart.

If Rebooter is run without any command line options, it launches the configuration window as above. If the command line option –reboot is used, Rebooter will automatically carry out the currently saved reboot option.

E.g.

rebooter.exe -reboot
Reboot cycling

BurnInTest can be setup so that the PC will reboot itself in a cycle, with test runs occurring between each reboot. BurnInTest uses another software utility called Rebooter to reboot / restart a PC. Rebooter can be used from within BurnInTest or it can be run by itself (look for the executable called, rebooter.exe in the BurnInTest installation directory).

To use Rebooter from within BurnInTest, go to the BurnInTest Preferences window and then select the "Post-Test" tab. Clicking on the 'Reboot options' button will open the Rebooter configuration window. You can get additional help about Rebooter options by clicking on the help button in the Rebooter configuration window.

Here is a brief description of how a BurnInTest can be set-up to reboot itself in a cycle.

**Step1 – Select and save Rebooter settings**

Set the ‘Maximum Reboots’ value to the number of cycles required.

Set the other parameters in Rebooter. (Reboot type, Delay, etc). Don’t set the delay value to be too short, as you want BurnInTest to fully stop before rebooting. 20 Seconds is a good value.

Save the settings, with the ‘Save Options’ button. Then close the Window.

**Step2 – Configure the settings you want in BurnInTest**

From the Preferences and Duty Cycle windows enter all the settings that you want for your test runs. When you close these windows, the settings are saved to disk and will become the new default values.

Alternatively a separate configuration file could be created (using the ‘Save As Config’ menu option) and used on the command line.

Make sure you

- Set a test period with the Auto-Stop option in the preferences window.
- Select ‘Exit & reboot PC’ in the Action after Auto-Stop check boxes.
- Have the log file accumulation option set in the Logging section of the preferences window.

**Step3 – Create an auto run shortcut**

Create an auto run short cut that points to the BurnInTest executable. You need to do this manually in Windows. The command line in the shortcut should use the ‘/r’ option. This will start tests executing in BurnInTest automatically. The command line for the shortcut should be something like

```
C:\Program Files\BurnInTest\bit.exe /r
```

The Rebooter help file (Rebooter.hlp ) also contains more details about how to start programs automatically with Windows.

**Step4 – Start the cycle**

Start the 1st test run from within BurnInTest, with the “Start Tests” button. At the end of the test period you have entered the PC will reboot according to the settings in Rebooter and after the reboot, BurnInTest will automatically re-start and do another test run, then Reboot again.

**Note:**

When Rebooter is started from BurnInTest, the Rebooter setting of "Auto load Rebooter at startup" is not applied. This allows BurnInTest to be setup as the auto restart program, and avoids the conflict of both BurnInTest and Rebooter autostarting after a reboot. This means that only a single reboot will be performed when rebooter is run from BurnInTest. To perform multiple reboots from within BurnInTest a script should be used with multiple REBOOT commands.

**More about Rebooter**
Rebooter is a small utility program developed by PassMark Software to help automate the PC hardware testing process. It has been designed to work with PassMark BurnInTest but will also work with 3rd party application. Rebooter allows you to,

- Shutdown, Reboot or Logout of a PC.
- Reboot a PC from the command line
- Set a timer so that the PC will reboot after a certain amount of time
- Setup a reboot loop, to reboot a PC over and over again in a cycle.
- Force a shutdown or request a shutdown.
- Enable and disable the Windows auto-login feature. (Windows 2000, XP and later)
- Include reboots into your hardware stress testing plan, (when used with BurnInTest).
PassMark Sleeper is a small freeware application, which allows the user to put their system into various states of sleep and hibernation. Sleeper can be called from BurnInTest using the ‘Close BurnInTest and run external file’ option from the Post test tab of the BurnInTest preferences window.

If Sleeper is called without any command line, the configuration screen above is displayed allowing the user to select options such as the Sleep state to enter, the duration of the sleep or hibernation etc. It is then possible to begin the sleep by clicking the Sleep Now button.

All settings can also be specified from the command line. See the Sleeper online help for more details on command line options.

Like Rebooter, Sleeper can be called from within BurnInTest by specifying the required command line from within the Pre-Test and/or Timers tabs on the preferences window.

Test hardware - Loopback plugs

The BurnInTest optical drive test provides the best test when used with PassMark CD/DVD test set as this provides pre-defined test data selected for the specific purpose of testing optical drives and that BurnInTest can better validate.

The following BurnInTest tests use loopback plugs or cables.
  o Serial Port test
  o Parallel Port test
  o Audio test
  o USB test. PassMark USB 2.0 Loopback plug: Used to test USB 1.x and 2.0 ports. Supports testing of FullSpeed (12Mb/s) and HighSpeed (480Mb/s). USB 2.0 ports will be tested at the faster HighSpeed.

It is recommended that you have at least one plug per port on the machine under test. For example, if the computer being tested has two COM ports, then having two serial loopback plugs will halve the amount of time you must spend testing.

To test Firewire or USB 3.0, it is recommended that the BurnInTest disk test be used with an external disk with the required interface.
Testing specialized hardware

If you have specialized hardware that BurnInTest does not test, you can write your own test and integrate it with BurnInTest (V5.0 and later).

Up to 5 Plugins can be specified. PassMark has developed Plugins for the following tests:

- Modem testing, using PassMark's ModemTest software;
- Keyboard testing, using PassMark's KeyboardTest software,
- Firewire port testing, using a PassMark developed Plugin and a "Kanguru FireFlash" drive.

Sample software ([http://www.passmark.com/ftp/BurnInTest_sample_plugin.zip](http://www.passmark.com/ftp/BurnInTest_sample_plugin.zip)) in C and C++ is also available to assist developing a Plugin for your specific hardware.

As well as demonstrating how to develop and integrate a test Plugin, these samples also provide examples of how to run pre-test applications and post test applications, and how to pass some information, such as pre-test configuration and post-test results, between BurnInTest and other applications.
Un-installer command line arguments

The BurnInTest uninstaller executable can be found in the BurnInTest installation directory. It is called, “unins000.exe”. Running this file will uninstall BurnInTest. There is also a command line option for the uninstaller.

/silent

When specified, the uninstaller will not ask the user any questions or display a message stating that uninstall is complete. Shared files that are no longer in use are deleted automatically without prompting. Any critical error messages will still be shown on the screen.
Feedback

We appreciate your feedback on this document, so if you find any mistakes or oversights please let us know.

Contacting PassMark Software

PassMark Software Pty Ltd
Suite 202, Level 2
35 Buckingham Street
Surry Hills, 2010
Sydney, Australia
Phone: +612 9690 0444
Fax: +612 9690 0445
http://www.passmark.com
Email Sales: sales@passmark.com
Email Support: help@passmark.com (send comments about this document to this address)
Appendix 1: Command line arguments

BurnInTest can be started from the DOS command line prompt or from a DOS batch file. (This can only be done from within Windows, you cannot boot to DOS and run BurnInTest as the software requires the Windows operating system to be running).

There are some command line parameters that when combined with a configuration file can help in automating the testing process.

Command line arguments can be used to specify certain BurnInTest options and can be particularly useful when automating BurnInTest.

Available command line arguments

The following are the BurnInTest command line parameters:

-A
BurnInTest will not collect System Information that requires elevated administrator privileges. This is useful for test automation under Windows Vista (and later) as Windows Vista would normally prompt with a security message before allowing the collection of the System information.

-B
BurnInTest will generate additional Serial port test information when activity trace level 2 logging is set. This can be useful to help debug specific serial port errors.

-C [configfilename]
 Loads the configuration file specified by [configfilename]

-D [minutes]
Sets the test duration to the value specified by minutes. Decimal values can be used.

-E [data]
Specifies the test data to use in the serial port test. [data] is a single byte in the range, 0..255. If this is not specified, the serial port test will use the default setting of random data. This can be useful to help debug specific serial port errors.

-F
If the disk self test is selected, then run the extended disk self test also (i.e. after the short disk self test).

-H
Set the screen resolution to 1024 x 768 with 32-bit color on startup. This is intended for use with BurnInTest when running on Microsoft WinPE.

-J
Cycle Disk test patterns between test files (when cyclic set). Note: Random seeking will be skipped in this case. This option has been added to allow multiple test patterns to be used across very large disks, without waiting for the disk to be completely tested with one pattern before moving on to the next disk test pattern.

-K
Keep disk test files. Specifies not to delete the disk drive test files when an error (e.g. Verification error) occurs. This is intended to assist investigating disk errors. It is recommended that this option is used in conjunction with the Auto Stop Tests on Error feature within Preferences. Once the test files have been investigated, they should be deleted manually.

-KA
Keep disk test files. Specifies not to delete the disk drive test files, in all cases. Once the test files have been investigated, they should be deleted manually.

-L [x,y,wt,ht]
Starts BurnInTest with the main window located at top-left co-ordinates x, y and with width wt and height ht. It is important that there are no white space characters in [x,y,wt,ht] specifications.
Automatically display the Machine ID Window when BurnInTest is started. This can be useful in a production line scenario to allow the tester to enter test specific information in a more automated fashion.

Force all files (e.g. configuration and log files) that would normally be saved or loaded from the User's Personal folder (ie. where the directory path has not been specified) to be saved or loaded from the BurnInTest directory. This can be useful when running BurnInTest from removable media, such as a USB drive, CD, DVD, Firewire drive etc.

Executes the tests immediately without needing to press the go button. It also skips the pre-test warning message.

On startup, BurnInTest will automatically run the script file specified by [scriptfilename]. [scriptfilename] can be an absolute or relative path to the script file, but if the path and/or filename contain any space characters, you should enclose the entire string in double quotes ("""). Learn more about Scripting here.

Often, -c <config file> is used with the -s command line parameter. This starts the script with a specific configuration, rather than the current configuration on the PC (in the LastUsed.bitcfg file). If the script contains REBOOT/REBOOTEND commands, the startup shortcut specifying BurnInTest (to start BurnInTest after reboot) should not include the -c <config file>, as in this case, the current configuration on the PC should be used for the continuation of a script after reboot.

Allows an external application to stop BurnInTest testing.

Force BurnInTest to set logging on at startup. Logging will be started with Activity trace 2 logging and a file name of Debug<_date/time>.trace.

Increase the error reporting detail for the standard RAM test.

Minimize the amount of System Information collected and displayed by BurnInTest. This can be useful for test automation as it can take some time to collect this information and slow test startup. It could also be used to simply reduce the amount of system information in reports. Note: That setting "-W" also sets "-A", i.e. you don't need to specify "-W -A", just "-W".

Skip the DirectX version checks at startup time. This can be useful for users that do not want to install the latest version of DirectX and do not want to use the DirectX tests (eg. 3D tests).

Start BurnInTest in Preferences editor mode.

Using command line arguments

There are a number of different way to specify command line arguments and this can depend on what you are trying to achieve. The most common options for specifying a command line argument are:

1) You can create a shortcut to bit.exe and then specify the command line in the shortcut target.

   Example 1: Start BurnInTest from a shortcut and run tests automatically.

   Right click on the bit.exe file and select Create Shortcut.

   Right click on the new shortcut file and select Properties and specify a Target with the command line argument.

   "C:\BurnInTest\bit.exe" -r
Now to run BurnInTest with these command line parameters, just double click the BurnInTest shortcut. You could also copy the shortcut to the Windows Startup folder to run BurnInTest automatically with the current default settings after Windows has booted.

2) You can open a command window, Start-R, “Cmd”. This will open a command window where you can type commands directly, such as shown below.

Example 2: Start BurnInTest from a command window and run a pre-defined script of tests

Select Windows Start-R, “Cmd”

Once the Windows command window is open, type in the commands:

```
cd C:\Program Files\BurnInTest
bit.exe -s MyScript.bits
```

3) You can write a batch file to execute BurnInTest with Command line arguments.

Example 3: Start BurnInTest from a batch file with a pre-defined configuration file

Create a file with an text editor (like Notepad) called Test.bat and include the lines in the file c:

```
cd "\program files\BurnInTest"
bit.exe –c "heavyload.bitcfg"
```

Example 4: Start BurnInTest on a USB flash drive from a batch file, run the tests automatically and use the bit.exe directory as the default directory (i.e. save log files to the USB thumb drive).

In the BurnInTest directory on the USB flash drive, you create a file, bit_p.bat, then edit this file and add the line:

```
bit.exe -r -p
```
Appendix 2: Scripting

Scripting tests

BurnInTest includes a simple scripting language that allows a series of tests with different configurations to be executed in a sequence.

A PASS/FAIL indication for all of the tests included in the script can be displayed at the end of the scripted test run. To achieve this you should select “Accumulate logs until manually cleared” from the Logging Preferences and results should be cleared before or at the start of the scripted test run.

Script files are ASCII text files that you can create with a text editor (e.g. notepad). The file name must end with the extension `.bits`. To start a script use the ‘Test / Execute script’ menu item.

Each script command must appear on its own line in the text file and the entire command must appear on a single line. (i.e. a single command cannot be split across multiple lines).

The command and its parameters must be separated by one or more spaces.

Comments can be included by starting the line with the ‘#’ character.

The main scripting commands are shown below. Scripting commands to configure test preferences are shown here.

EXECUTE COMMAND

Overview:
Executes an external file and continues processing the script.

Syntax:
```
EXECUTE <Filename> <Parameters>
```

Parameters:
- Filename: The name of the file to execute. The file name must be enclosed in double quotes ("""). The file must be an executable.
- Parameters: Any command line parameters which you wish to pass to your executable. If any of these parameters are filenames, you should enclose them in double quotes ("")

Examples:
- #Start up Notepad with some results before starting tests.
  - EXECUTE "c:\windows\system32\notepad.exe" "c:\MyResults\Results.txt"

RUN CONFIG

EXECUTEWAIT COMMAND

Overview:
Runs an executable file and waits for that process to finish before continuing to process the script.

Passing configuration information from an external program to BurnInTest.
The executable file may produce a file of scripting commands that are to be run by BurnInTest once the executable file has closed. As an example, this may be used to set the Machine type, serial number, test notes etc at the start of a script file.

The only script commands that will be processed by BurnInTest are the SET… commands. All other commands will be ignored. If this file, called a Sub-Script file, is created it must:

- be placed in the same directory as the EXECUTEWAIT executable file;
- must be called “bit-script-input.txt”; and
- must conform to the scripting file format.

After processing the script commands in this file, BurnInTest will delete the file.

See STARTLOGGING.

**Syntax:**

EXECUTE <Filename> <Parameters>

**Parameters:**

<table>
<thead>
<tr>
<th>Filename</th>
<th>The name of the file to execute. The file name must be enclosed in double quotes (“”). The file must be an executable.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>Any command line parameters which you wish to pass to your executable. If any of these parameters are filenames, you should enclose them in double quotes (“”)</td>
</tr>
</tbody>
</table>

**Examples:**

#Start PassMark’s Sleeper application with user defined parameters.

EXECUTEWAIT “sleeper.exe” -S1000 -R 30 -N 1 –E

#Start up Notepad with some results before starting tests.

EXECUTEWAIT “c:\winnt\system32\notepad.exe” “c:\MyResults\Results.txt”

RUN CONFIG

**EXIT COMMAND**

**Syntax:**

EXIT

**Parameters:**

- none

Note: BurnInTest will exit with a return code indicating whether the tests run during the script passed or failed. An exit code of 0 means SUCCESS, while and exit code of 1 means FAILURE.

**Examples:**

RUN CONFIG

#Exit BurnInTest and return an exit code to the batch file

EXIT
LOAD COMMAND

Syntax:

LOAD <File name>

Parameters:

<File name> The full path name to a configuration file. This configuration file must have been previously created from within BurnInTest. A partial path name can be used to load a configuration file from the current default directory (e.g. My Documents\<UserName>\PassMark\BurnInTest\), where <UserName> is the administrator account that BurnInTest is run under, or the bit.exe directory (e.g. C:\Program Files\BurnInTest\). A loaded configuration file will overwrite all preferences currently selected.

Examples:

#Load the low load configuration file that we created earlier
LOAD “C:\ConfigurationFiles\LowLoad.bitcfg”

LOG COMMAND

Overview:

Writes text to the detailed error and status log history.

Syntax:

LOG <Text>

Parameters:

<Text> The text to be added to the detailed error and status log history.

Examples:

#Start an S1 sleep and log the start and stop times
LOG "Sleep S1 Duration 60 seconds starting"
EXECUTEWAIT SLEEPER -S1 -D 60
LOG "Sleep S1 Duration 60 seconds complete"

LOOP

Syntax:

LOOP <number of iterations>

<Tasks to repeat>

Parameters:

<number of iterations> The number of times the Tasks within brackets will be repeated.

Examples:

LOG "Start"
SETDURATION 1
LOOP 3
MESSAGE COMMAND
Syntax:
    MESSAGE  <Message Text>
Parameters:
    < Message Text > A single line of text that will be displayed in a window with an OK button. The user must click on the OK button to continue with the script.
Examples:
    MESSAGE “Insert the test disc into the DVD drive then click on OK to proceed with the test”

SETCUSTOMER COMMAND
Syntax:
    SETCUSTOMER < Customer name >
Parameters:
    <Customer name> The customer name string. Must be enclosed in double quotes (“”).
Examples:
    #Set the customer name
    SETCUSTOMER "The big PC store"
NOTE: SETSERIAL, SETMACHINETYPE, SETNOTES, SETCUSTOMER, SETTECHNICIAN commands should be included as the first commands in a script so that the report settings are included in any log file System Information.

SETCYCLES COMMAND
Syntax:
    SETCYCLES <Number of test cycles>
Parameters:
    < Number of test cycles > Sets the number of test cycles that will lead to an automatic stopping of the test runs after all selected tests have reached or exceeded this number of test cycles. Using this command is the same as changing the auto-stop number of cycles from the preferences window.
Example 1:
    #Set the number of test cycles to 1
    SETCYCLES 1
Example : Run each test one cycle in series.
    SETCYCLES 1
    SETDURATION 0
    LOG “Run CPU Test”
NOTE: Automatic stopping after a set number of test cycles is only supported in the licensed version of BurnInTest.

**SETDURATION COMMAND**

*Syntax:*

```
SETDURATION  <Duration>
```

*Parameters:*

- `<Duration>` Sets the test duration in minutes. Using this command is the same as changing the auto-stop period from the preferences window.

*Examples:*

- `#Set the test duration to 90 seconds`
  - `SETDURATION  1.5`

**SETDUTYCYCLE COMMAND**

*Syntax:*

```
SETDUTYCYCLE  <Test Name>  <Duty setting>
```

*Parameters:*

- `<Test Name>` See below for a list of all test names.
- `<Duty setting>` Sets the duty cycle for the specified test to the value specified. Values must be between 1 and 100. Using this command is the same as changing the duty cycle value from the Test duty cycle window.

*Examples:*

- `#Set the disk test to maximum load`
  - `SETDUTYCYCLE  DISK  100`
- `#Set the CPU test to medium load`
  - `SETDUTYCYCLE  DISK  65`

**SETMACHINETYPE COMMAND**

*Syntax:*

```
SETMACHINETYPE < Machine Name>
```

*Parameters:*

- `<Machine Name>` The Machine Name. Must be enclosed in double quotes (" ").

*Examples:*

- `#Set Machine type`
  - `SETMACHINETYPE "Dell XPS800"`

**NOTE:** `SETSERIAL`, `SETMACHINETYPE`, `SETNOTES`, `SETCUSTOMER`, `SETTECHNICIAN` commands should be included as the first commands in a script so that the report settings are included in any log file System Information.
SETNOTES COMMAND
Syntax:
   SETNOTES <Notes>
Parameters:
   < Notes >       The Notes. Must be enclosed in double quotes (" ").
Examples:
   #Set Notes
   SETNOTES "Test notes defined by the external application."
NOTE: SETSERIAL, SETMACHINETYPE, SETNOTES, SETCUSTOMER, SETTECHNICIAN commands should be included as the first commands in a script so that the report settings are included in any log file System Information.

SETSERIAL COMMAND
Syntax:
   SETSERIAL < Serial Number>
Parameters:
   <Serial Number>  The serial number string. Must be enclosed in double quotes (" ").
Examples:
   #Set the serial number
   SETSERIAL "1234-shdflghs-GHGHG"
NOTE: SETSERIAL, SETMACHINETYPE, SETNOTES, SETCUSTOMER, SETTECHNICIAN commands should be included as the first commands in a script so that the report settings are included in any log file System Information.

SETTECHNICIAN COMMAND
Syntax:
   SETTECHNICIAN < Technician name>
Parameters:
   <Technician name>  The technician (or testers) name. Must be enclosed in double quotes (" ").
Examples:
   #Set the technician name
   SETTECHNICIAN "Bill Smith"
NOTE: SETSERIAL, SETMACHINETYPE, SETNOTES, SETCUSTOMER, SETTECHNICIAN commands should be included as the first commands in a script so that the report settings are included in any log file System Information.

SLEEP COMMAND
Syntax:
   SLEEP <Delay period>
Parameters:
< Delay period >  An integer that represents the number of milliseconds to pause before continuing with the next command in the script.

Examples:

#Pause 2 seconds
SLEEP 2000

**STARTLOGGING COMMAND**

To set report items like serial number in the report (log file) a script command, STARTLOGGING, is used to turn logging on and create a log file (start logging). The intent of this is to allow any test script reporting items to be configured before the log file is created.

Syntax:

STARTLOGGING

Parameters:

None.

Examples:

#Run an external application to set the report configuration subscript and pass this to BurnInTest via the bit-script-input.txt file
EXECUTEWAIT ".\bitsetup.exe"
#Start logging using current settings with the SETXXX commands in the resulting bit-script-input.txt file
STARTLOGGING
#Set first test duration time for 1 minute
SETDURATION 1
#Run the test
RUN CONFIG

**REBOOT and REBOOTEND COMMAND**

Note: These commands should only be used where multiple reboots within one script are required. The commands are designed in such a way that they ONLY make sense in the following context…

BurnInTest must be launched automatically at start up using a shortcut to bit.exe in the Start-Up directory. The shortcut must use the /s command line parameter to automatically run the script, which contains the REBOOT command. So if, for example the script file containing the REBOOT command was called ‘Reboot.bits’, then the command line ‘Target’ of the shortcut would look something like

“C:\Program Files\BurnInTest\bit.exe” /r /s Reboot.bits

These commands require that the Rebooter application is present in the BurnInTest application directory. Any reboots occurring as a result of these commands will use the current Rebooter settings. BurnInTest must have write access to the directory that the script is located in.

Overview:

REBOOT reboots the computer. After the computer boots up, and BurnInTest restarts, the script will continue to execute at the line following the REBOOT command.

REBOOTEND reboots the computer. After the computer boots up, BurnInTest will restart, but the script will no longer continue to execute.

Example:

MESSAGE “Run some 3D tests”
RUN 3D
MESSAGE “Reboot for the first time”
REBOOT
MESSAGE “Run some 2D tests”
RUN 2D
MESSAGE “Reboot for the second time”
REBOOT
MESSAGE “And now one final reboot”
REBOOTEND
MESSAGE “This message will never be displayed”

Note: It is recommended to use “Accumulate logs” when using REBOOT and REBOOTEND.

**RUN COMMAND**

*Syntax:*

RUN <Test Name>

*Parameters:*

<Test Name> See below for a list of all test names.

*Examples:*

#Run the CD test with the current settings
RUN CD

#Run all the tests in the current configuration simultaneously
RUN CONFIG

*<Test Name> Parameter*

The test name parameter can take the following values. The first value “CONFIG” is special because it does not refer to the name of an individual test. When used with the RUN command it causes all tests in the current configuration file to be started simultaneously.

CONFIG  
CPU (or MATHS)  
CD  
DISK  
MEMORY  
NETWORK  
PARALLEL  
PRINTER  
SERIAL  
SOUND  
TAPE
Example

#Load my preferred test configuration
LOAD "MyConfiguration1.bitcfg"
#Override the test duration for all tests
SETDURATION 60
MESSAGE "Click on OK to start test run"
RUN CPU
MESSAGE "Insert test discs into both the CD and DVD drive"
RUN CD
#Load my preferred test configuration for disk testing
LOAD "MyDiskConfig.bitcfg"
RUN CONFIG
Scripting test preferences
Starting with BurnInTest 7.0, most test preferences can be configured from the test script. This allows for convenient multiple test runs within a script without loading pre-saved configuration files before each test run.

This section describes scripting many of the user settings available from the BurnInTest Preferences Windows.

Notes:
- All commands are case insensitive. They are written in upper case for illustration.
- Scripting of the 3D, tape and Printer preferences are not supported in the current version of BurnInTest.

SETDEFAULTPREFERENCES COMMAND

Overview:
Set the test preferences to the initial installation state.

Syntax:
SETDEFAULTPREFERENCES

Parameters:
This command does not require any parameters.

Example:
SETDEFAULTPREFERENCES

SETDISK COMMAND

Overview:
Set the disk test options. The first parameter for the command is always disk name except for ‘ALL’ parameter which tests all drives.

Syntax:
SETDISK DISK <parameter> MODE <parameter> SLOW <parameter> FILE <parameter> BLOCK <parameter> SMART <parameter> SEEK <parameter> BADSECTOR <parameter> THRESHOLD <parameter> DUTY <parameter>

SETDISK ALL <parameter>

Parameters:
DISK: Disk Name (volume letter)
This parameter must be specified first.

MODE (Test Mode):
1. CYCLIC (Cyclic)
2. SEQUENTIAL (Sequential data pattern)
3. SEEKING (Random data with random seeking)
4. HIGHLOW (High low frequency data overwrite)
5. BUTTERFLY (Butterfly seeking)
6. BINARY1 (Binary data pattern 1)
7. BINARY2 (Binary data pattern 2)
8. ZEROS (Zeros data pattern)
9. ONES (Ones data pattern)
10. RANDOM (Random data pattern)

SLOW (Slow drive threshold): Value in MB/sec
FILE (File Size): Value in % of disk space
BLOCK (Block size): Value in KB
SMART (Log SMART errors): [ YES | NO ]
SEEK (Seek count): Value
BADSECTOR (Log bad sector increase): [ YES | NO ]
THRESHOLD (Bad sector threshold): Value
DUTY (Duty cycle override): Value in %

The following parameters are global (i.e. not for a particular disk):

ALL (Select all hard drives): [ YES | NO ]

Examples:
SETDISK DISK c: MODE binary1 SLOW 4 FILE 2 BLOCK 512 SMART no
SETDISK DISK d: SEEK 10 BADSECTOR yes THRESHOLD 25 DUTY 5
SETDISK ALL yes

SETOPTICALDISK COMMAND

Overview:
Set the Optical disk test preferences. The first parameter for this command is always optical disk name except for ‘ALL’ parameter which tests all drives. The Burn CD and Burn DVD test modes cannot be scripted.

Syntax:

SETOPTICALDISK DRIVE <parameter> MODE <parameter> SEEK <parameter> SEEK_VAL <parameter> BLOCK_AUTO <parameter>

Parameters:

DRIVE: Drive Name
This parameter must be specified first.

MODE (Test mode):
1. NO_TEST (No test)
2. MUSIC (Music CD playback)
3. DATA (Data disk read and verify)
4. PASSMARK (Passmark test CD/DVD/BD)
5. NO_CD (No media in drive)

SEEK (Seek count): [ YES | NO ]

COUNT (Seek count value): Value of seek count

BLOCK_AUTO (Autoplay configuration): [ YES | NO ]

The following parameters are global (i.e. not for a particular disk):

ALL (Select all optical drives): [ YES | NO ]

Example:

SETOPTICALDISK DRIVE d: MODE passmark SEEK yes SEEK_VAL 5 BLOCK_AUTO yes
SETOPTICALDISK ALL yes

SETLOG COMMAND

Overview:
Set the Logging test options.

Syntax:

SETLOG LOG <parameter> NAME <parameter> PREFIX <parameter> TIME <parameter> REPORT <parameter> LOGLEVEL <parameter> TRACELEVEL <parameter> SUM <parameter> LINES <parameter> PERIODIC <parameter> FILE <parameter>

Parameters:

LOG (Turn automatic logging on): [ YES | NO ]

NAME (Log file name): File path within double quotes

PREFIX (Log file prefix): File prefix within double quotes

TIME (Time Stamped files): [ YES | NO ]

REPORT (Report type) (only one selectable):
1. TEXT (Text)
2. HTML (HTML)
3. CERT (Customer certificate)
LOGLEVEL (Log file detail level):

1. NO (result summary)
2. Yes (normal)

TRACELEVEL (Trace file detail level), the trace file format will be set to text:

1. No (no trace)
2. A1 (Activity trace 1)
3. A2 (Activity trace 2)

SUM (Summarize):  [ YES | NO ]

LINES (Max file size): Maximum number of lines

PERIODIC (Periodic result summary logging interval): Value in minutes

FILE (Pre-test logging Options):

1. NEW (Clear the test result and create a new log file)
2. APPEND (Clear the test result and append the next test to the existing log file)
3. ACCUMULATE (Accumulate test result across test runs and append to existing log

Examples:

SETLOG LOG yes NAME "file1.html" PREFIX "computername" TIME yes
SETLOG REPORT html REPORT cert LOGLEVEL yes TRACELEVEL a2 SUM yes LINES 10000
SETLOG PERIODIC 2 FILE append

Notes:

For convenience, the BurnInTest V6.0 syntax for SETLOG is still supported, however new scripts should not use the V6.0 syntax:

SETLOG COMMAND

Syntax:

SETLOG <Filename>

Parameters:

<Filename> The name of the log file. The file name must be enclosed in double quotes (""").

SETERRORS COMMAND

Overview:

Sets the errors test preferences

Syntax:
SETErrORS ACTION <parameter> WINDOWS <parameter>

Parameters:

ACTION (Action on Error):
1. CONTINUE
2. SOUND (Play sound and continue)
3. BEEP (System beep and continue)
4. STOP (Auto stop tests)

WINDOWS:
1. BLOCK (Block critical window errors)
2. ALLOW (Allow critical window errors)

Example:
SETErrORS ACTION  beep WINDOWS allow

SETNETWORK COMMAND

Overview:
Set the network preferences for test. The advanced network test options cannot be scripted.

Syntax:
SETNETWORK  ADD[1-12] <parameter>
SETNETWORK  ERROR <parameter> TIMEOUT <Parameter> RATIO <parameter>
SETNETWORK  ALL <parameter>
SETNETWORK  BLUETOOTH <parameter> NAME <parameter>

Parameters:
ADD1 – ADD12: Supply the network address

The following parameters are global (i.e. not for a network port test disk):

ERROR (Error setting for bad packet):
1. EVERYPACKET (Every bad packet generated an error)
2. HIGHRATIO (High bad packet ratio generated an error)

TIMEOUT (Timeout): value in msec.

RATIO (Bad Packet Ratio): Ratio of Bad packet to total packets

ALL (Test mode): [LOOPBACK | YES | ETHERNET | WIRELESS ]

BLUETOOTH (bluetooth_active): [ YES | NO ]
NAME (Remote device name): Name of the remote Bluetooth device

Examples:

SETNETWORK ADD1 "192.1.1.1"
SETNETWORK ADD2 "192.1.1.2"
SETNETWORK ADD12 "192.1.1.2"
SETNETWORK ALL yes
SETNETWORK ERROR everypacket TIMEOUT 3 RATIO 2
SETNETWORK BLUETOOTH yes NAME "My test device"

SET2DGRAPHICS COMMAND

Overview:
Sets the 2D graphics test options. The scrolling ‘H’ EMC test and multiple monitor testing option cannot be scripted.

Syntax:
SET2DGRAPHICS TYPE <parameter> IGNORE <parameter>

Parameters:

TYPE:
1. LOCAL (Local video memory only)
2. ALL (All available video memory)
3. LINES (Lines and Bitmap)

IGNORE (Ignore skipped frames): [ YES | NO ]

Example:
SET2DGRAPHICS TYPE all IGNORE yes

SETUSB COMMAND

Overview:
Set the USB test options. Script currently support change in maximum number of ports

Syntax:
SETUSB <number>

Parameters:

number: Value of ports to be tested

Example:
SETRAM COMMAND

Overview:
Set the RAM test options. Multi-Process Torture test and AWE tests options cannot be scripted.

Syntax:
SETRAM TYPE <parameter> PATTERN <parameter> PRETEST <parameter> LOG <parameter>

Parameters:
TYPE (Ram test mode):
1. STANDARD (Standard)
PATTERN: [CYCLIC | SEQUENCE | BINARY1 | BINARY2 | ZEROS | ONES | ADJACENCY ]
PRETEST (run the standard test as pre-test): [ YES | NO ]
LOG (Log memory applications): [ YES | NO ]

Example:
SETRAM TYPE standard PATTERN binary1 PRETEST no LOG yes

SETPLUGIN COMMAND

Overview:
Set the plug-in preferences for test.

Syntax:
SETPLUGIN PLUGIN[1-5] <parameter> PRETEST[1-3]<parameter>

Parameters:
PLUGIN1 - PLUGIN5 (specify plug-in application): specify the plug-in path within double quotes.
PRETEST1 - PRETEST3 (run as pre-test Plugin option available for Plugins 1-3): [ YES | NO ]

Examples:
SETPLUGIN PLUGIN1 "c:\plugin1.exe" PRETEST1 yes
SETPLUGIN PLUGIN2 "C:\plugin2.exe" PRETEST2 no

SETVIDEOPLAYBACK COMMAND

Overview:
Set the Video playback files for test. Multiple monitor testing options cannot be scripted.
Syntax:

SETVIDEOPLAYBACK VIDEO[1-4] <parameter>

Parameters:

VIDEO1 - VIDEO4: Video file path within double quotes

Example:

SETVIDEOPLAYBACK VIDEO1 "c:\video1.avi" VIDEO2 "c:\video2.mpg" VIDEO3 "c:\video3.wmv" VIDEO4 "c:\video4.mpeg"

SETSERIALPORTS COMMAND

Overview:

Set the serial test options.

Syntax:

SETSERIALPORTS COM<0-63> <parameter>

SETSERIALPORTS DISABLE <parameter> FLOWCONTROL <parameter> TIMEOUT <parameter> SPEED <parameter>

Parameters:

COM0 – COM63: Com port selection

1. DETECT_LOOP (Detect and loop test)
2. DETECT (Detect test only)
3. NONE (remove from comp port selected)

The following are global COM port test parameters (i.e. not specific to a COM port):

DISABLE (Disable RTS/CTS and DSR/DTR test phase): [ YES | NO ]

FLOWCONTROL (Use RTS/CTS flow control during loopback): [ YES | NO ]

TIMEOUT (Send and recv Timeout): Value in ms

PORT_SPEED (Kb/s): [ 110 | 300 | 600 | 1200 | 2400 | 4800 | 9600 | 14400 | 19200 | 38400 | 56000 | 57600 | 115200 | CYCLE_115K | 128000 | 256000 | CYCLE_256K | 921600 ]

Examples:

SETSERIALPORTS COM1 detect_loop COM2 detect_loop COM3 detect

SETSERIALPORTS DISABLE no TIMEOUT 20 SPEED 19200

SETPARALLELPORTS COMMAND

Overview:

Set the Parallel test options.
Syntax:

SETPARALLELPORTS PORT <parameter> TEST <parameter>

Parameters:

PORT (Parallel port number): Value of port number
TEST (Test type):
  1. DETECT_LOOP (Detection and loopback test)
  2. DETECT (Port detection only)

Note: Only 1 parallel port can be tested at a time.

Example:

SETPARALLELPORTS PORT lpt1 TEST detect

SETSCOUND COMMAND

Overview:

Set the Sound test options.

Syntax:

SETSCOUND TEST STANDARD WAV <parameter> MIDI<parameter> MP3 <parameter>
SETSCOUND TEST LOOPBACK DISTORTION <parameter>

Parameters:

TEST(Sound test type):
  1. STANDARD (Standard test)
  2. LOOPBACK (Loopback test)
WAV : File path
MIDI: File path
MP3 : File path
DISTORTION (Max Distortion) : value

Examples:

SETSCOUND TEST standard WAV "myfile.wav" MIDI "myfile.midi" MP3 "myfile.mp3"
SETSCOUND TEST loopback DISTORTION 7

SETCPU COMMAND

Overview:
Set the CPU test options. The CPU affinity options cannot be scripted.

Syntax:

SETCPU GP <parameter>  FPI <parameter>  PRIME <parameter>  MAX_HEAT <parameter>  MMX <parameter>  3DNOW <parameter>  SSE <parameter>  SSE2 <parameter>  SSE3 <parameter>  SSE4_1 <parameter>  SSE4_2 <parameter>  SSE4A <parameter>

Parameters:

GP (General Purpose Instructions): [ YES | NO ]
FPU (Floating Point unit Instructions): [ YES | NO ]
PRIME (Prime number test): [ YES | NO ]
MAX_HEAT: [ YES | NO ]
MMX: [ YES | NO ]
3DNOW: [ YES | NO ]
SSE: [ YES | NO ]
SSE2: [ YES | NO ]
SSE3: [ YES | NO ]
SSE4_1: [ YES | NO ]
SSE4_2: [ YES | NO ]
SSE4A: [ YES | NO ]

Examples:

SETCPU GP yes FPU yes PRIME yes MAX.HEAT yes MMX yes 3DNOW yes SSE yes SSE2 yes
SETCPU SSE3 yes SSE4_1 yes SSE4_2 yes SSE4A yes

SETPRETEST COMMAND

Overview:

Sets the test preferences for Pre-Test options

Syntax:

SETPRETEST ACTION <parameter> PATH <parameter>

Parameters:

ACTION (Action before running tests):

1. NONE (start immediately)
2. WARNING (display warning messages)
3. RUN (run external application)
4. WAIT (run external application and wait)

PATH (External application): Specify the application path within double quotes

Example:

SETPRETEST ACTION run PATH "c:\filename.exe"

SETPOSTTEST COMMAND

Overview:

Set the Post-Test options. The reboot options cannot be scripted.

Syntax:

SETPOSTTEST AUTO_STOP_P <parameter> MAN_STOP_P <parameter> RESULT_P <parameter> SYSTEM_BEEP_P <parameter> RUN_P <parameter>

SETPOSTTEST AUTO_STOP_F <parameter> MAN_STOP_F <parameter> RESULT_F <parameter> SYSTEM_BEEP_P <parameter> RUN_F <parameter>

Parameters:

AUTO_STOP_P /AUTO_STOP_F (auto stop options):

1. NORMAL (Stop tests)
2. PRINT (Stop test and print result)
3. RUNAPP (Run extern app and exit)
4. REBOOT (Exit and reboot PC)

MAN_STOP_P/MAN_STOP_F (Manual stop options)

1. NORMAL (Stop tests)
2. RUNEXTERNAL (Display option to run application)

RESULT_P/RESULT_F (Always display result window): [YES | NO]

SYSTEM_BEEP_P/SYSTEM_BEEP_F (Use System Beep on autostop): [YES | NO]

RUN_P/RUN_F (run external application): Specify the application path within double quotes.

Examples:

SETPOSTTEST AUTO_STOP_P runapp MAN_STOP_P normal RESULT_P no RUN_P "pass_app.exe"

SETPOSTTEST AUTO_STOP_F runapp MAN_STOP_F normal RESULT_F no RUN_F "fail_app.exe"
Appendix 3 – Example reports

Text report

PassMark BurnInTest Log file - http://www.passmark.com

Date: 05/18/11 17:06:21

BurnInTest V7.0 Pro 0000
(64-bit)
System summary:
Windows 7 Ultimate Edition build 7600 (64-bit),
1 x Intel(R) Core(TM) i5 CPU 750 @ 2.67GHz,
4.0GB RAM,
ATI Radeon HD 3450,
466GB HDD,
CD-RW/DVDRW/BD/HD DVD-ROM,

General:
System Name: COREI5
System Model: GBTUACPI
BIOS Manufacturer: GBT - 42302e31
BIOS Release Date: 2009/08/01

CPU:
CPU manufacturer: GenuineIntel
CPU Type: Intel(R) Core(TM) i5 CPU 750 @ 2.67GHz
Codename: Lynnfield
CPUID: Family 6, Model 1E, Stepping 5, Revision B1
Socket: LGA1156
Lithography: 45nm
Physical CPU's: 1
Cores per CPU: 4
Hyperthreading: Disabled
CPU features: MMX SSE SSE2 SSE3 SSSE3 SSE4.1 SSE4.2 DEP PAE Intel64 VMX SMX Turbo

Clock frequencies:
- Measured CPU speed: 2665.2 MHz [Turbo: 3198.2MHz]
- Multiplier: x20.0 [Turbo: x24.0]
- Base Clock: 133.3 MHz
Multiplier range: Min: x9, Max non turbo: x20 [Turbo: 4C: x21, 3C: x21, 2C: x24, 1C: x24]
Cache per CPU package:
- L1 Instruction Cache: 4 x 32 KB
- L1 data cache: 4 x 32 KB
- L2 cache: 4 x 256 KB
- L3 cache: 8 MB
TDP Limit: 95 Watts
TDC limit: 89 Amps

Memory
Total Physical Memory: 4091MB
Available Physical Memory: 2871MB
Memory devices:
DIMM 3:
- 2GB DDR3 SDRAM PC3-8500
- Corsair CM3X2G1600C9DHX
  - 1.5V, Clk: 533.3MHz, Timings 7-7-7-20 (@ Max. freq.)
DIMM 4:
- 2GB DDR3 SDRAM PC3-8500
- Corsair CM3X2G1600C9DHX
  - 1.5V, Clk: 533.3MHz, Timings 7-7-7-20 (@ Max. freq.)
Virtual memory: C:\\pagefile.sys (allocated base size 4091MB)

Graphics
ATI Radeon HD 3450
- Chip Type: ATI display adapter (0x95C5)
DAC Type: Internal DAC(400MHz)
Memory: 256MB
BIOS: 113-AAAAAXXX-XXX
Driver provider: ATI Technologies Inc.
Driver version: 8.821.0.0
Driver date: 1-26-2011
Monitor 1: 1920x1200x32 59Hz (Primary monitor)

Disk volumes
A: Floppy
C: Local drive, NTFS, (465.76GB total, 390.29GB free)
D: Optical drive, PASSMARK_DVD, UDF
E: Removable

Disk drives
Disk drive: Model WDC WD5000AADS-00L4B1 ATA Device (Size: 465.76GB)
Disk drive: Model Corsair VoyagerGT USB Device (Size: 15.12GB)

Optical drives
D: HL-DT-ST BDDVDRW GGC-H20L (CD-RW/DVDRW/BD/HD DVD-ROM)

Network
Realtek PCIe GBE Family Controller (Speed: 1Gb/s)

Ports
Communications Port: COM1 - RS232 Serial Port (max Baud rate: 115200)
Parallel port: LPT1

USB
Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B3B
Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B3E
Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B3F
Intel(R) 5 Series/3400 Series Chipset Family USB Enhanced Host Controller - 3B3C
Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B3E
Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B3F
Intel(R) 5 Series/3400 Series Chipset Family USB Enhanced Host Controller - 3B3C
Microsoft 3-Button Mouse with IntelliEye(TM)
Intel(R) 5 Series/3400 Series Chipset Family USB Enhanced Host Controller - 3B3E
Corsair Voyager GT (SN: A804012700016209)

RESULT SUMMARY
Test Start time: Wed May 18 17:06:21 2011
Test Stop time: Wed May 18 17:07:27 2011
Test Duration: 000h 01m 06s
Temperature CPU 1 average (Min/Current/Max): 34.3C / 49.8C / 52.3C
Temperature CPU 1 core 1 (Min/Current/Max): 37.0C / 52.0C / 55.0C
Temperature CPU 1 core 2 (Min/Current/Max): 32.0C / 50.0C / 51.0C
Temperature CPU 1 core 3 (Min/Current/Max): 36.0C / 51.0C / 53.0C
Temperature CPU 1 core 4 (Min/Current/Max): 32.0C / 46.0C / 50.0C
Temperature HDD 0 (WDC WD5000AADS-00L4B1 ATA Device) (Min/Current/Max): 34.0C / 34.0C / 34.0C

Test Name                   Cycles   Operations      Result Errors   Last Error
------------------------------------------------------------------------------------------------------------
CPU                        3           40.559 Billion  PASS 0           No errors
Memory (RAM)               0           4.493 Billion   PASS 0           No errors
2D Graphics                0           400            PASS 0           No errors
Sound                      0           1.125 Million   PASS 0           No errors
Optical disk (D:)          1           251 Million     PASS 0           No errors
Disk (C:)                  0           3.150 Billion   PASS 0           No errors
Network 1                  1           9760           PASS 0           No errors

TEST RUN PASSED
SERIOUS ERROR SUMMARY FOR THE LAST TEST RUN
BurnInTest results

<table>
<thead>
<tr>
<th>BurnInTest Version</th>
<th>V7.0 Pro 0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>05/18/11 17:06:21</td>
</tr>
<tr>
<td>Logging level</td>
<td>Normal</td>
</tr>
</tbody>
</table>

System summary

System summary

Windows 7 Ultimate Edition build 7600 (64-bit),
1 x Intel(R) Core(TM) i5 CPU 750 @ 2.67GHz,
4.0GB RAM,
ATI Radeon HD 3450,
486GB HDD,
CD-RW/DVDRW/BD/HD DVD-ROM

General

<table>
<thead>
<tr>
<th>System Name:</th>
<th>COREI5</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Model:</td>
<td>GBTJACPI</td>
</tr>
<tr>
<td>BIOS Manufacturer:</td>
<td>GBT - 42302e31</td>
</tr>
<tr>
<td>BIOS Release Date:</td>
<td>2009/06/01</td>
</tr>
</tbody>
</table>
### CPU

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU manufacturer:</td>
<td>Genuine Intel</td>
</tr>
<tr>
<td>CPU Type:</td>
<td>Intel® Core™ i5 CPU 750 @ 2.67GHz</td>
</tr>
<tr>
<td>Codename:</td>
<td>Lynnfield</td>
</tr>
<tr>
<td>CPUID</td>
<td>Family 6, Model 1F, Stepping S, Revision B1</td>
</tr>
<tr>
<td>Socket:</td>
<td>LGA1156</td>
</tr>
<tr>
<td>Lithography:</td>
<td>45nm</td>
</tr>
<tr>
<td>Physical CPU(s):</td>
<td>1</td>
</tr>
<tr>
<td>Cores per CPU:</td>
<td>4</td>
</tr>
<tr>
<td>Hyperthreading:</td>
<td>Disabled</td>
</tr>
<tr>
<td>CPU features:</td>
<td>MMX SSE SSE2 SSE3 SSE4 SSE4.1 SSE4.2 DEP PAE Intel64 VMX SMX Turbo</td>
</tr>
<tr>
<td>Clock frequencies:</td>
<td></td>
</tr>
<tr>
<td>- Measured CPU speed:</td>
<td>2665.2 MHz [Turbo: 3198.2 MHz]</td>
</tr>
<tr>
<td>- Multiplier:</td>
<td>x20.0 [Turbo: x24.0]</td>
</tr>
<tr>
<td>- Base Clock:</td>
<td>133.3 MHz</td>
</tr>
<tr>
<td>Multiplier range:</td>
<td>Min: x9, Max non turbo: x20 [Turbo: 4C: x21, 3C: x21, 2C: x24, 1C: x24]</td>
</tr>
<tr>
<td>Cache per CPU package:</td>
<td></td>
</tr>
<tr>
<td>- L1 Instruction Cache:</td>
<td>4 x 32 KB</td>
</tr>
<tr>
<td>- L1 data cache:</td>
<td>4 x 32 KB</td>
</tr>
<tr>
<td>- L2 cache:</td>
<td>4 x 256 KB</td>
</tr>
<tr>
<td>- L3 cache:</td>
<td>8 MB</td>
</tr>
<tr>
<td>TDP Limit:</td>
<td>95 Watts</td>
</tr>
<tr>
<td>TDC limit:</td>
<td>89 Amps</td>
</tr>
</tbody>
</table>
## Memory

<table>
<thead>
<tr>
<th>Total Physical Memory:</th>
<th>4091MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Physical Memory:</td>
<td>2871MB</td>
</tr>
<tr>
<td>Memory devices:</td>
<td></td>
</tr>
<tr>
<td><strong>DIMM 3:</strong></td>
<td>2GB DDR3 SDRAM PC3-8500</td>
</tr>
<tr>
<td></td>
<td>Corsair CM3X2G1600C9DHX</td>
</tr>
<tr>
<td></td>
<td>1.5V, Clk: 533.3MHz, Timings 7-7-7-20 (@ Max. freq.)</td>
</tr>
<tr>
<td><strong>DIMM 4:</strong></td>
<td>2GB DDR3 SDRAM PC3-8500</td>
</tr>
<tr>
<td></td>
<td>Corsair CM3X2G1600C9DHX</td>
</tr>
<tr>
<td></td>
<td>1.5V, Clk: 533.3MHz, Timings 7-7-7-20 (@ Max. freq.)</td>
</tr>
<tr>
<td>Virtual memory:</td>
<td>C:\pagefile.sys (allocated base size 4091MB)</td>
</tr>
</tbody>
</table>

## Graphics

<table>
<thead>
<tr>
<th>ATI Radeon HD 3450</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Chip Type:</td>
<td>ATI display adapter (0x95C8)</td>
</tr>
<tr>
<td>DAC Type:</td>
<td>Internal DAC(400MHz)</td>
</tr>
<tr>
<td>Memory:</td>
<td>256MB</td>
</tr>
<tr>
<td>BIOS:</td>
<td>113-AAXXXXXX-XXX</td>
</tr>
<tr>
<td>Driver provider:</td>
<td>ATI Technologies Inc.</td>
</tr>
<tr>
<td>Driver version:</td>
<td>8.321.0.0</td>
</tr>
<tr>
<td>Driver date:</td>
<td>1-26-2011</td>
</tr>
<tr>
<td>Monitor 1:</td>
<td>1920x1200x32 59Hz (Primary monitor)</td>
</tr>
</tbody>
</table>
## Disk volumes

- A: Floppy
- C: Local drive, NTFS. (465.76GB total, 390.25GB free)
- D: Optical drive, PASSMARK_DVD, UDF
- E: Removable

## Disk drives

- Disk drive: Model WD WD5000AAAE-00L4B1 ATA Device (Size: 465.76GB)
- Disk drive: Model Corsair Voyager GT USB Device (Size: 15.12GB)

## Optical drives


## Network

- Realtek PCIe GBE Family Controller (Speed: 1Gb/s)

## Ports

- Communications Port: COM1 - RS232 Serial Port (max Baud rate: 115200)
- Parallel port: LPT1
### USB

<table>
<thead>
<tr>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B3B</td>
</tr>
<tr>
<td>Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B3E</td>
</tr>
<tr>
<td>Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B3F</td>
</tr>
<tr>
<td>Intel(R) 5 Series/3400 Series Chipset Family USB Enhanced Host Controller - 3B3C</td>
</tr>
<tr>
<td>Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B36</td>
</tr>
<tr>
<td>Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B37</td>
</tr>
<tr>
<td>Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B30</td>
</tr>
<tr>
<td>Intel(R) 5 Series/3400 Series Chipset Family USB Universal Host Controller - 3B39</td>
</tr>
<tr>
<td>Microsoft Microsoft 3-Button Mouse with IntelliEye(TM)</td>
</tr>
<tr>
<td>Intel(R) 5 Series/3400 Series Chipset Family USB Enhanced Host Controller - 3B34</td>
</tr>
<tr>
<td>Corsair Voyager GT (SN: A804012700016209)</td>
</tr>
</tbody>
</table>

### Detailed event log

<table>
<thead>
<tr>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG NOTE: 2011-05-18 17:05:21, Status, PassMark BurnInTest V7.0 Pro 0000 BETA</td>
</tr>
<tr>
<td>LOG NOTE: 2011-05-19 17:06:22, Status, Main Tests started</td>
</tr>
<tr>
<td>LOG NOTE: 2011-05-18 17:06:22, CPU, SSE4a test(s) selected, but CPU feature not detected</td>
</tr>
<tr>
<td>LOG NOTE: 2011-05-19 17:07:27, Status, Test run stopped</td>
</tr>
</tbody>
</table>
# Result summary

<table>
<thead>
<tr>
<th>Test Start time</th>
<th>Wed May 18 17:06:21 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Stop time</td>
<td>Wed May 18 17:07:27 2011</td>
</tr>
<tr>
<td>Test Duration</td>
<td>00h 01m 06s</td>
</tr>
<tr>
<td>Temperature CPU 1 average (Min/Current/Max)</td>
<td>34.0°C / 49.0°C / 52.0°C</td>
</tr>
<tr>
<td>Temperature CPU 1 core 1 (Min/Current/Max)</td>
<td>37.0°C / 52.0°C / 55.0°C</td>
</tr>
<tr>
<td>Temperature CPU 1 core 2 (Min/Current/Max)</td>
<td>32.0°C / 50.0°C / 51.0°C</td>
</tr>
<tr>
<td>Temperature CPU 1 core 3 (Min/Current/Max)</td>
<td>36.0°C / 51.0°C / 53.0°C</td>
</tr>
<tr>
<td>Temperature CPU 1 core 4 (Min/Current/Max)</td>
<td>32.0°C / 46.0°C / 50.0°C</td>
</tr>
<tr>
<td>Temperature HDD 0 (WDC WD5000AAKS-00L4B1 ATA Device) (Min/Current/Max)</td>
<td>34.0°C / 34.0°C / 34.0°C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test</th>
<th>Cycles</th>
<th>Operations</th>
<th>Result</th>
<th>Errors</th>
<th>Last Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>3</td>
<td>40.559 Billion</td>
<td>PASS</td>
<td>0</td>
<td>No errors</td>
</tr>
<tr>
<td>Memory (RAM)</td>
<td>0</td>
<td>4.493 Billion</td>
<td>PASS</td>
<td>0</td>
<td>No errors</td>
</tr>
<tr>
<td>2D Graphics</td>
<td>0</td>
<td>400</td>
<td>PASS</td>
<td>0</td>
<td>No errors</td>
</tr>
<tr>
<td>Temperature</td>
<td>-</td>
<td>-</td>
<td>PASS</td>
<td>0</td>
<td>No errors</td>
</tr>
<tr>
<td>Optical disk (Dr:)</td>
<td>1</td>
<td>251 Million</td>
<td>PASS</td>
<td>0</td>
<td>No errors</td>
</tr>
<tr>
<td>Sound</td>
<td>0</td>
<td>1.125 Million</td>
<td>PASS</td>
<td>0</td>
<td>No errors</td>
</tr>
<tr>
<td>Disk (Ci)</td>
<td>0</td>
<td>3.150 Billion</td>
<td>PASS</td>
<td>0</td>
<td>No errors</td>
</tr>
<tr>
<td>Network 1</td>
<td>1</td>
<td>9750</td>
<td>PASS</td>
<td>0</td>
<td>No errors</td>
</tr>
</tbody>
</table>

**TEST RUN PASSED**
BurnInTest Certificate

Report Date: 12/15/08
Customer: Ian
Technician: Tom
Generated by: BurnInTest Version V6.0 Pro

System summary

<table>
<thead>
<tr>
<th>System component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Name</td>
<td>XY6400XP32</td>
</tr>
<tr>
<td>Operating system</td>
<td>Windows XP Professional Service Pack 3 build 2600 (32-bit)</td>
</tr>
<tr>
<td>CPU type</td>
<td>Intel(R) Core(TM)2 Quad CPU Q6600 (2.394.0 MHz)</td>
</tr>
<tr>
<td>RAM</td>
<td>2047 MB</td>
</tr>
<tr>
<td>Video card</td>
<td>NVIDIA Quadro FX 1400 (128MB)</td>
</tr>
<tr>
<td>Video card</td>
<td>ATI Radeon HD 2450 (512MB)</td>
</tr>
<tr>
<td>Disk drive</td>
<td>Model ST3150027AS (Size: 149.0GB)</td>
</tr>
<tr>
<td>Disk drive</td>
<td>Model WDC WD6000JD-60LSA0 (Size: 74.5GB)</td>
</tr>
<tr>
<td>Optical drive</td>
<td>TSSCorp CD/DVDW SH-W1622</td>
</tr>
</tbody>
</table>

Result summary

| Test Start time | Mon Dec 15 18:27:41 2008 |
| Test Stop time  | Mon Dec 15 18:28:43 2008 |
| Test Duration   | 00h 01m 02s |

<table>
<thead>
<tr>
<th>Test</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>PASS</td>
</tr>
<tr>
<td>Memory (RAM)</td>
<td>PASS</td>
</tr>
<tr>
<td>2D Graphics</td>
<td>PASS</td>
</tr>
<tr>
<td>Disk (C:)</td>
<td>PASS</td>
</tr>
<tr>
<td>Optical disk (F:)</td>
<td>PASS</td>
</tr>
<tr>
<td>Sound</td>
<td>PASS</td>
</tr>
<tr>
<td>Network 1</td>
<td>PASS</td>
</tr>
</tbody>
</table>

TEST RUN PASSED

Notes

Certification

This document certifies that the Tests described above have been carried out by a suitably qualified technician on the System described above.

Signed

Suite 10, Level 1, 38 Waterloo St., Surry Hills, 2010, Sydney, Australia
Phone +61 2 9690 0444 Fax +61 2 9690 0445
E-mail: info@passmark.com