PASSMARK™

SOFTWARE

White paper

Using BurnInTest in a production line environment

Abstract:

This white paper describes how BurnInTest 6.0 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.

Document Edition: 6.0

Date: 26 March 2009

Contents

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.

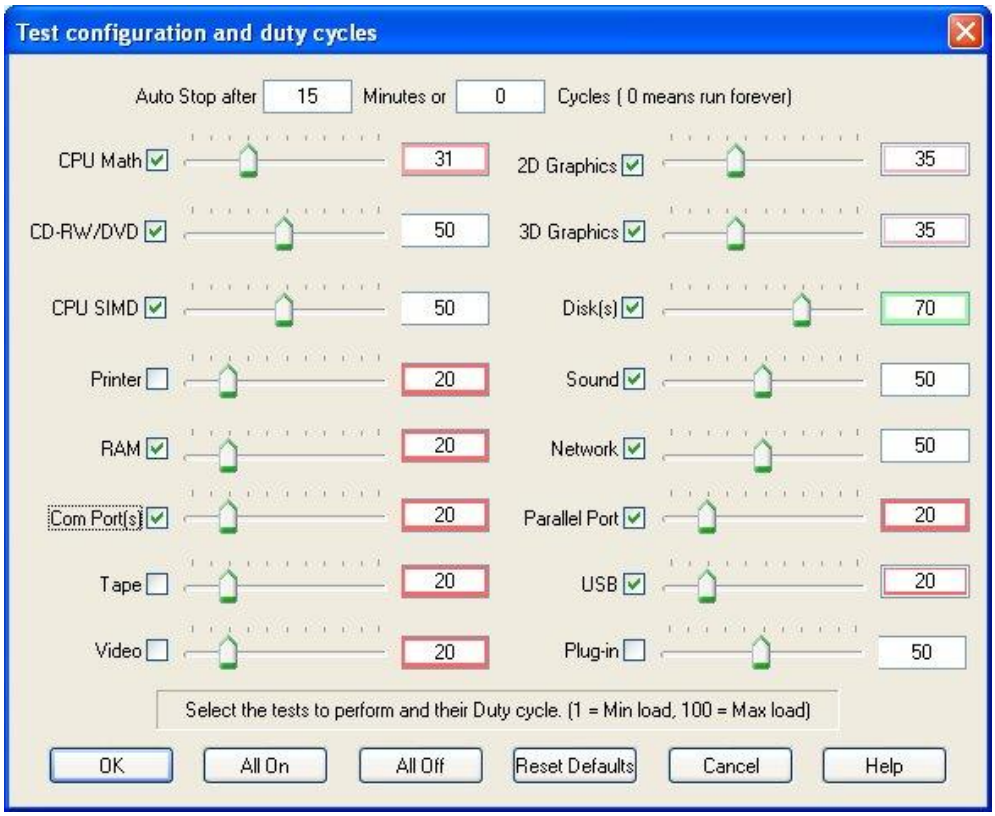

Figure 1 – Duty cycle window. These settings along with other preferences are saved in configuration files

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'.

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

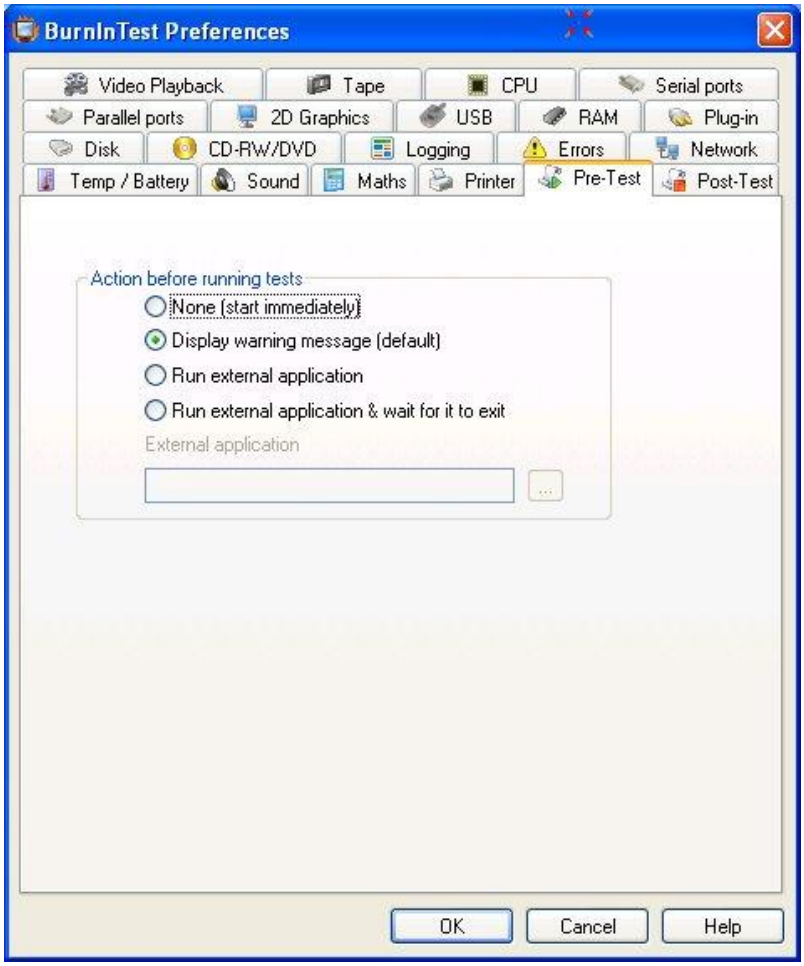

Figure 2– Pre-Test Preferences.

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.

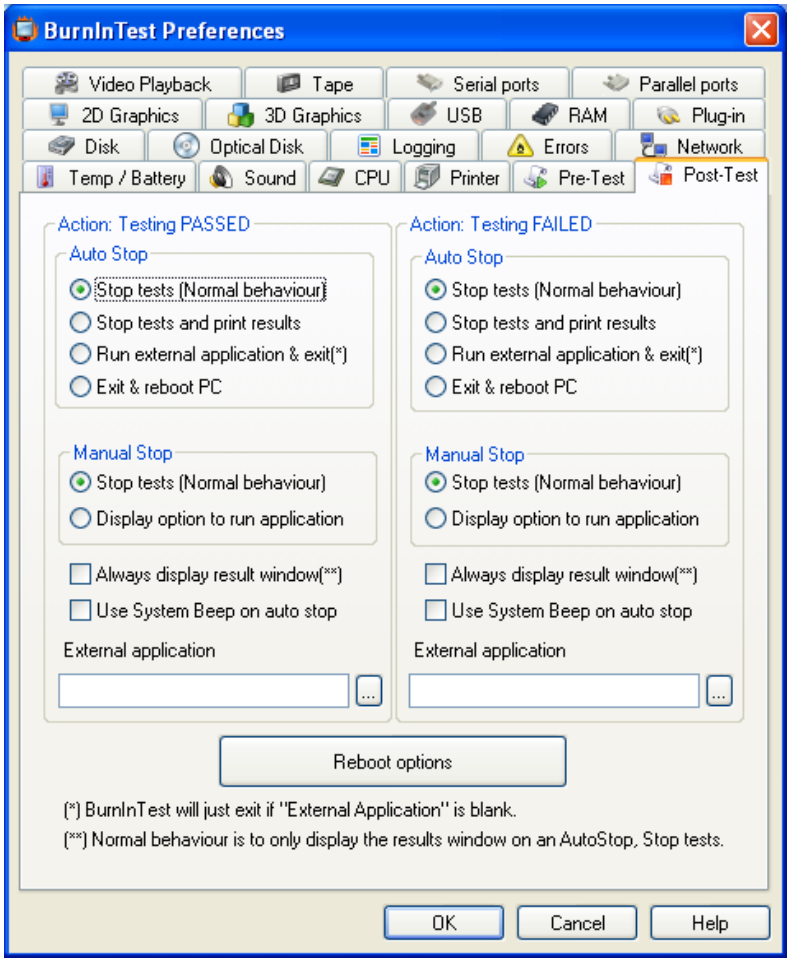

Figure 3– Timers tab in preferences window.

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.

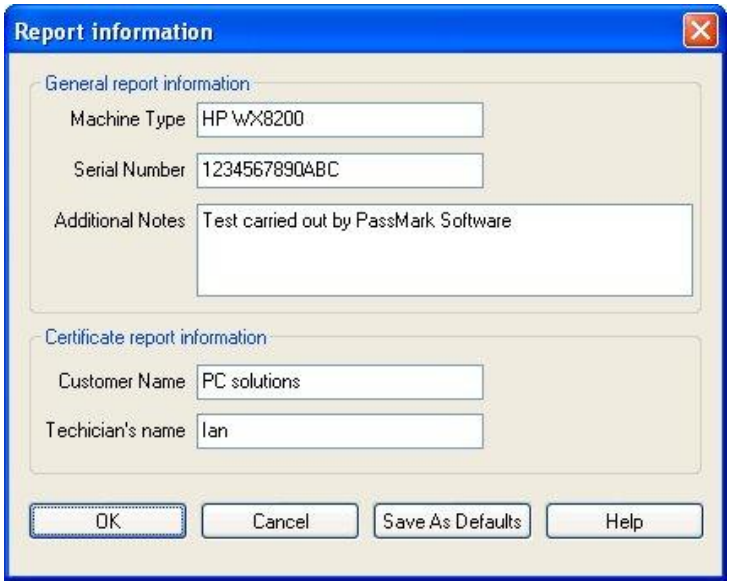

Figure 3 – Machine identity window.

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.

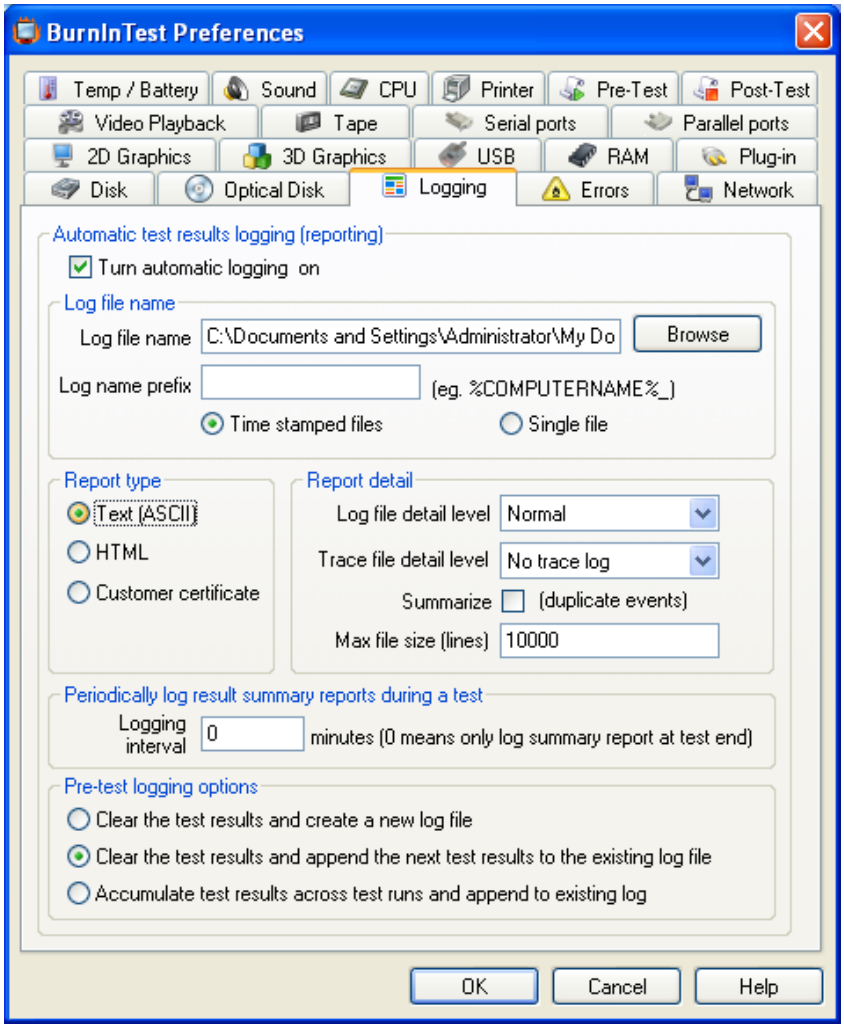

Figure 4 – Logging preferences window

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.

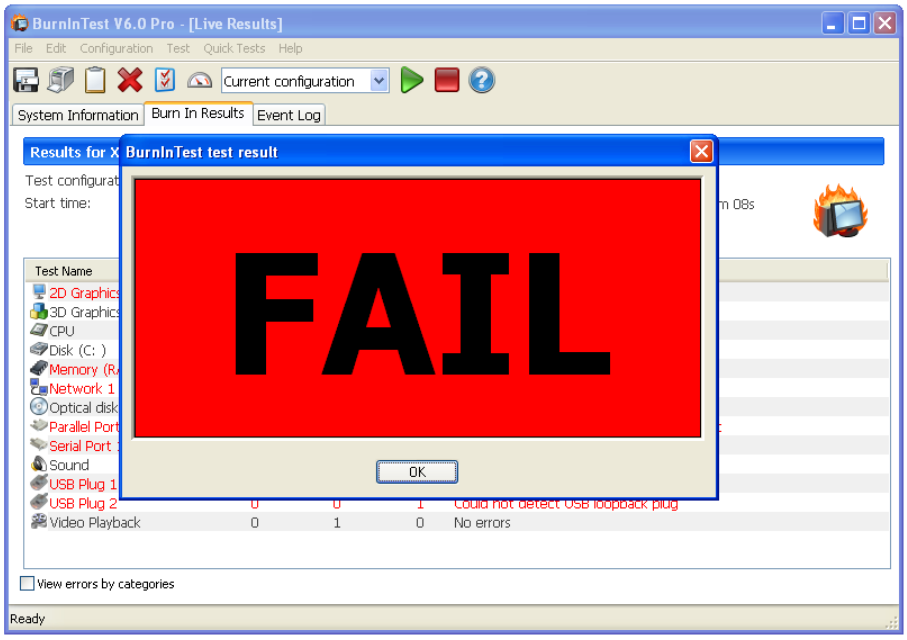

Figure 5 – Flashing Test failure window

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”

Running BurnInTest on a System without an operating system

A bootable CD can be created with Microsoft Windows XP and BurnInTest 6.0 Professional using Bart Pre-install Environment. 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 document (http://passmark.com/ftp/BIT_BartPE_v6.0.pdf) has been produced to assist people in setting up an environment that allows PassMark BurnInTest to be used in these situations.

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.

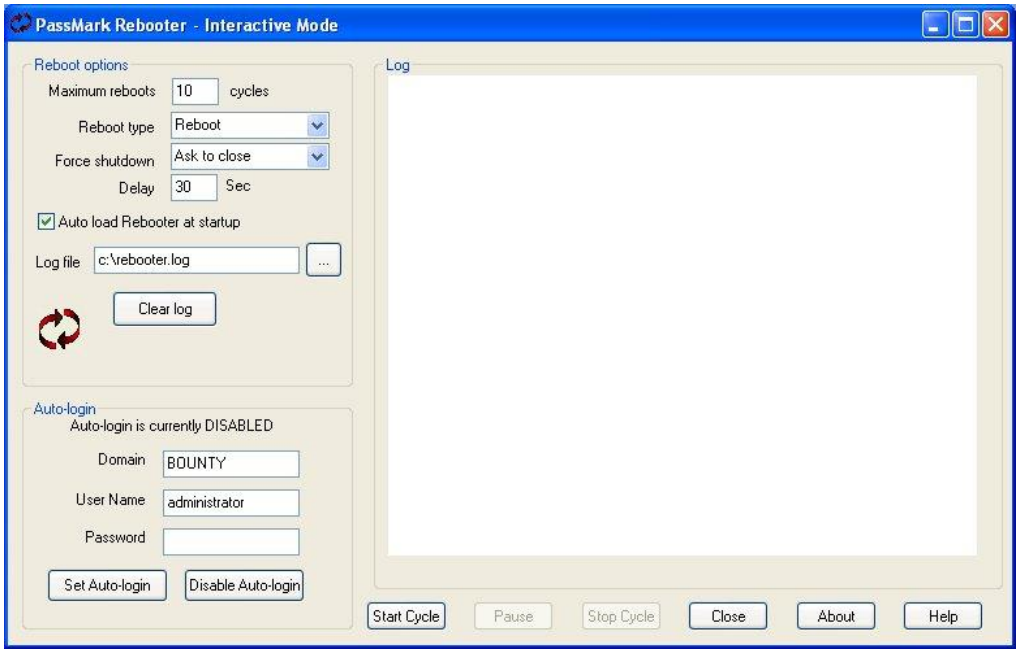

Figure 6 – Rebooter

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).

Sleeper

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.

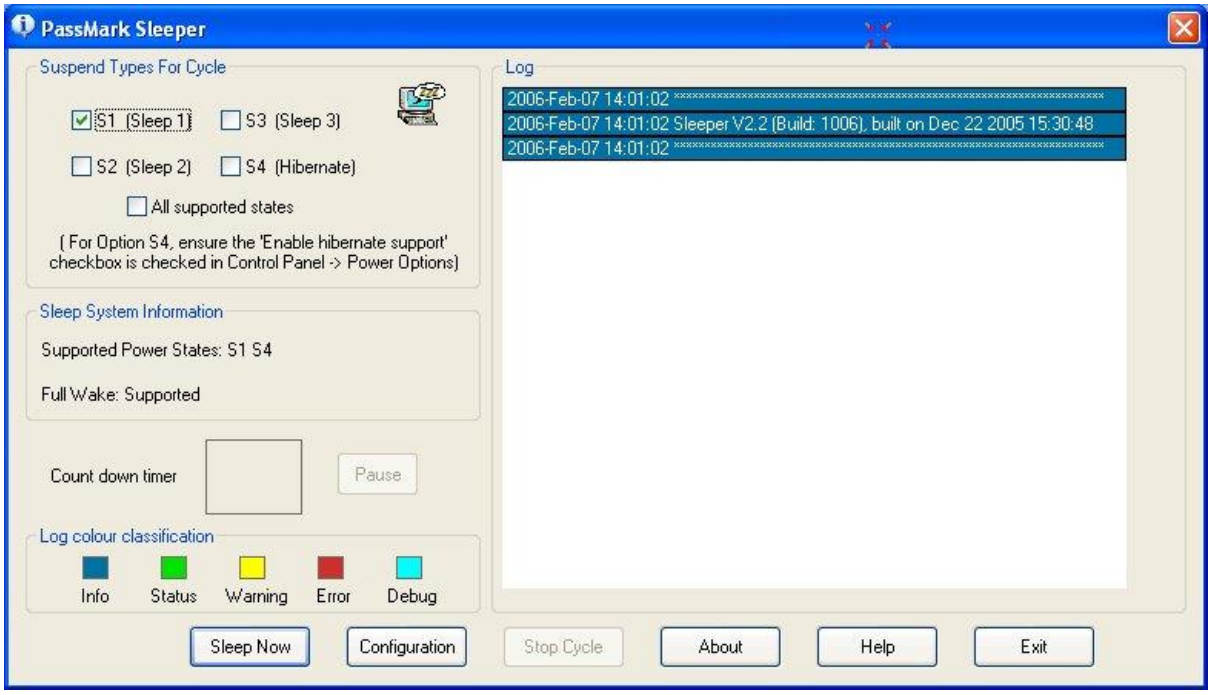

Figure 7 – Sleeper

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.

Sleeper is available from http://www.passmark.com/ftp/sleeper.zip

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.

- Serial Port test
- Parallel Port test
- Audio test
- 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.

When testing BurnInTest with the BurnInTest Firewire Plugin, a 3rd part Firewire plug is recommended (Kangaru FireFlash).

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) 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 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.

-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.

-M
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.

-P
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.

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

-S [scriptfilename]
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.

-U
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.

-V
Increase the error reporting detail for the standard RAM test.

-W
Minimize the amount of System Information collected and displayed by BurnInTest. This can be useful for test automation as is 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".

-X
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).

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 thumb 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 thumbdrive, you create a file, bit_p.bat, then edit this file and add the line:

bit.exe -r -p

Appendix 2: Scripting

BurnInTest includes a simple scripting language that allows tests 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 following commands are available in the current version of the software

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:\winnt\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 is 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.

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 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\) or the bit.exe directory (e.g. C:\Program Files\BurnInTest\). A loaded configuration file will over write 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 60seconds 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

  LOG "CPU Test"
  RUN CPU
}
```

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”

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”

RUN CPU

Etc…

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
```

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-shdfgdhs-GHGHG"
```

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"
```

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."
```

SETLOG COMMAND

Syntax:

SETLOG <Filename>

Parameters:

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

Examples:

#Set log file with full path

SETLOG "C:\Program Files\Plugin\plugin_log"

#Set log file with path relative to BurnInTest drive

SETLOG "\Program Files\Plugin\plugin_log"

#Set log file with path relative to BurnInTest application path

SETLOG "plugin_log"

Notes:

Loading a saved configuration file will overwrite this setting.

SETPLUGIN COMMAND

Sets up the *first* external test plug-in application executable file name.

Syntax:

SETPLUGIN <Filename>

Parameters:

<Filename> The name of the plugin file. The file name must be enclosed in double quotes (“”). The file must be an executable, and the “.exe” must be included.

Examples:

#Set plug-in test application file with full path

SETPLUGIN "C:\Program Files\Plugin\plugin.exe”

#Set plug-in test application file with path relative to BurnInTest drive

SETPLUGIN "\Program Files\Plugin\plugin.exe”

#Set plug-in test application file with path relative to BurnInTest application path

SETPLUGIN "plugin.exe”

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-shdfgdhs-GHGHG"

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"

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

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” /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.

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 takes 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

MATHS

CPU (or MATHS)

CD

DISK

MEMORY

NETWORK

PARALLEL

PRINTER

SERIAL

SOUND

TAPE

USB

2D

3D

VIDEO

PLUGIN

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
```

Appendix 3 – Example reports

Text report

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

Date: 12/15/08 14:52:05

BurnInTest V6.0 Pro 1000
Customer: ian
Technician: tom
Logging detail level: Normal

**************
SYSTEM SUMMARY
**************
Windows XP Professional Service Pack 3 build 2600 (32-bit),
1 x Genuine Intel(R) CPU @ 2.40GHz,
2.0GB RAM,
NVIDIA Quadro FX 1400, ATI Radeon HD 3450,
149GB HDD, 75GB HDD,
CD/DVDRW,

GENERAL
System Name: XW6400XP32
System Model: HP xw6400 Workstation
Motherboard Manufacturer: Hewlett-Packard
Motherboard Name: 0A04h
Motherboard Serial Number: SAS1044ze2
BIOS Manufacturer: Hewlett-Packard
BIOS Version: 786D4 v02.26
BIOS Release Date: 09/21/2007

CPU
CPU manufacturer: GenuineIntel
CPU Type: Genuine Intel(R) CPU @ 2.40GHz
Physical CPU's: 1
Cores per CPU: 4
Hyperthreading: Not capable
CPU features: MMX SSE SSE2 SSE3 SSSE3 DEP PAE Intel64 VMX
Clock frequencies:
   Measured CPU speed: 2353.8 MHz
   Multiplier: x9.0
   Scalable Bus Speed: 266.0 MHz
   Front Side Bus: 1066.0 MHz
Cache per CPU package:
   L1 Instruction Cache: 4 x 32 KB
   L1 Data Cache: 4 x 32 KB
   L2 Cache: 2 x 4096 KB

MEMORY
Total Physical Memory: 2047MB
Available Physical Memory: 1232MB
Memory devices:
   DIMM01:
   - DDR2 FB-DIMM, 2048MB, 667MHz,
   - Serial number: F29F0812
   DIMM02:
   - Not populated
   DIMM03:
   - Not populated
   DIMM04:
   - Not populated
   SYSTEM ROM:
   - Flash, 33MB,
Virtual memory: C:\pagefile.sys (allocated base size 1000MB)
```

```
GRAPHICS
NVIDIA Quadro FX 1400
   Chip Type: Quadro FX 1400
   DAC Type: Integrated RAMDAC
   Memory: 128MB
   BIOS: Version 5.41.02.17.24
   Driver provider: NVIDIA
   Driver version: 6.14.11.6996
   Driver date: 5-26-2008
   Monitor 1: 1280x1024x32 60Hz (Primary monitor)
ATI Radeon HD 3450
   Chip Type: ATI display adapter (0x95C5)
   DAC Type: Internal DAC(400MHz)
   Memory: 512MB
   BIOS: 113-AAXXXXX-XXX
   Driver provider: ATI Technologies Inc.
   Driver version: 8.451.0.0
   Driver date: 12-20-2007
   Monitor 2: 1280x1024x32 60Hz

DISK VOLUMES
C:  Local drive, NTFS, (74.53GB total, 7.71GB free)
E:  Optical drive, TEST_DVD, CD/DVDW SH-W162Z, UDF
F:  Local drive, Backup, NTFS, (139.28GB total, 2.24GB free)
K:  Local drive, test, NTFS, (0.49GB total, 0.48GB free)
L:  Local drive, New Volume, NTFS, (1.95GB total, 1.84GB free)
M:  Local drive, New Volume, NTFS, (1.95GB total, 1.94GB free)

DISK DRIVES
Disk drive: Model ST3160827AS (Size: 149.05GB)
Disk drive: Model WDC WD800JD-60LSA0 (Size: 74.53GB)

OPTICAL DRIVES
E: TSSTcorp CD/DVDW SH-W162Z (CD/DVDRW)

NETWORK
Broadcom NetXtreme Gigabit Ethernet

PORTS
Communications Port: COM1 - RS232 Serial Port (max Baud rate: 115200)
Parallel port: LPT1
Keyboard Port: PS/2  connector
Mouse Port: PS/2  connector
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2688
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2689
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268A
   - Microsoft Natural® Ergonomic Keyboard 4000
   - Kensington USB/PS2 Wheel Mouse
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268B
Intel(R) 631xESB/6321ESB USB2 Enhanced Host Controller - 268C
******************
DETAILED EVENT LOG
******************
LOG NOTE: 2008-12-15 14:52:05, Status, PassMark BurnInTest V6.0 Pro 1000
LOG NOTE: 2008-12-15 14:52:06, Status, Main Tests started
LOG NOTE: 2008-12-15 14:52:06, CPU, 3DNow! SSE4.1 SSE4.2 SSE4a test(s) selected, but CPU feature not
detected
LOG NOTE: 2008-12-15 14:53:08, Status, Test run stopped

**************
RESULT SUMMARY
**************
Test Start time: Mon Dec 15 14:52:05 2008
Test Stop time: Mon Dec 15 14:53:07 2008
Test Duration: 000h 01m 02s

Test Name                   Cycles   Operations      Result Errors   Last Error
              2D Graphics   0        275             PASS   0        No errors
                      CPU   11       36.701 Billion  PASS   0        No errors
               Disk (C: )   0        563 Million     PASS   0        No errors
             Memory (RAM)   5        5.362 Billion   PASS   0        No errors
                Network 1   1        8640            PASS   0        No errors
        Optical disk (E:)   1        25.285 Million  PASS   0        No errors
                    Sound   0        970200          PASS   0        No errors
TEST RUN PASSED
```

```
****************************************
SERIOUS ERROR SUMMARY FOR THE LAST TEST RUN
****************************************

-----------------------------------------------------------------------------------------------
```

HTML report

BurnInTest results

BurnInTest Version	V6.0 Pro 1000
Date	12/15/08 14:59:17
Customer:	ian
Technician:	tom
Logging level	Normal

System summary

System summary

Windows XP Professional Service Pack 3 build 2600 (32-bit),
1 x Genuine Intel(R) CPU @ 2.40GHz,
2.0GB RAM,
NVIDIA Quadro FX 1400, ATI Radeon HD 3450,
149GB HDD, 75GB HDD,
CD/DVDRW,

General

System Name	XW6400XP32
System Model	HP xw6400 Workstation
Motherboard Manufacturer	Hewlett-Packard
Motherboard Name	0A04h
Motherboard Serial Number	SAS1044ze2
BIOS Manufacturer	Hewlett-Packard
BIOS Version	786D4 v02.26
BIOS Release Date	09/21/2007

CPU

CPU manufacturer	GenuineIntel
CPU Type	Genuine Intel(R) CPU @ 2.40GHz
Physical CPU's	1
Cores per CPU	4
Hyperthreading	Not capable
CPU features	MMX SSE SSE2 SSE3 SSSE3 DEP PAE Intel64 VMX
Clock frequencies	
- Measured CPU speed	2353.8 MHz
- Multiplier	x9.0
- Scalable Bus Speed	266.0 MHz
- Front Side Bus	1066.0 MHz
Cache per CPU package	
- L1 Instruction Cache:	4 x 32 KB
- L1 Data Cache	4 x 32 KB
- L2 Cache	2 x 4096 KB

Memory

Total Physical Memory	2047MB
Available Physical Memory	1222MB
Memory devices:	
DIMM01:	DDR2 FB-DIMM, 2048MB, 667MHz,
	Serial number: F29F0812
DIMM02:	Not populated
DIMM03:	Not populated
DIMM04:	Not populated
SYSTEM ROM:	Flash, 33MB,
Virtual memory	C:\pagefile.sys (allocated base size 1000MB)

Graphics

NVIDIA Quadro FX 1400	
Chip Type:	Quadro FX 1400
DAC Type:	Integrated RAMDAC
Memory:	128MB
BIOS:	Version 5.41.02.17.24
Driver provider:	NVIDIA
Driver version:	6.14.11.6996
Driver date:	5-26-2008
Monitor 1:	1280x1024x32 60Hz (Primary monitor)
ATI Radeon HD 3450	
Chip Type:	ATI display adapter (0x95C5)
DAC Type:	Internal DAC(400MHz)
Memory:	512MB
BIOS:	113-AAXXXXX-XXX
Driver provider:	ATI Technologies Inc.
Driver version:	8.451.0.0
Driver date:	12-20-2007
Monitor 2:	1280x1024x32 60Hz

Disk volumes

C: Local drive, NTFS, (74.53GB total, 7.71GB free)
E: Optical drive, TRANSFORMERS_PAL, CD/DVDW SH-W162Z, UDF
F: Local drive, Backup, NTFS, (139.28GB total, 2.24GB free)
K: Local drive, test, NTFS, (0.49GB total, 0.48GB free)
L: Local drive, New Volume, NTFS, (1.95GB total, 1.84GB free)
M: Local drive, New Volume, NTFS, (1.95GB total, 1.94GB free)

Disk drives

Disk drive: Model ST3160827AS (Size: 149.05GB)
Disk drive: Model WDC WD800JD-60LSA0 (Size: 74.53GB)

Optical drives

E: TSSTcorp CD/DVDW SH-W162Z (CD/DVDRW)

Network

Broadcom NetXtreme Gigabit Ethernet

Ports

Communications Port: COM1 - RS232 Serial Port (max Baud rate: 115200)
Parallel port: LPT1
Keyboard Port: PS/2 connector
Mouse Port: PS/2 connector

USB Host Controllers and connected devices

Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2688
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2689
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268A
- Microsoft Natural® Ergonomic Keyboard 4000
- Kensington USB/PS2 Wheel Mouse
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268B
Intel(R) 631xESB/6321ESB USB2 Enhanced Host Controller - 268C

Detailed event log

Event
LOG NOTE: 2008-12-15 14:59:17, Status, PassMark BurnInTest V6.0 Pro 1000 Beta 15
LOG NOTE: 2008-12-15 14:59:17, Status, Main Tests started
LOG NOTE: 2008-12-15 14:59:18, CPU, 3DNow! SSE4.1 SSE4.2 SSE4a test(s) selected, but CPU feature not detected
LOG NOTE: 2008-12-15 15:00:19, Status, Test run stopped

Result summary

Test Start time	Mon Dec 15 14:59:17 2008
Test Stop time	Mon Dec 15 15:00:19 2008
Test Duration	000h 01m 02s

Test	Cycles	Operations	Result	Errors	Last Error
CPU	11	36.157 Billion	PASS	0	No errors
Memory (RAM)	5	5.429 Billion	PASS	0	No errors
2D Graphics	0	286	PASS	0	No errors
Disk (C:)	0	568 Million	PASS	0	No errors
Optical disk (E:)	1	26.153 Million	PASS	0	No errors
Sound	0	1.003 Million	PASS	0	No errors
Network 1	1	8480	PASS	0	No errors

TEST RUN PASSED

Notes	

Serious error summary for the last test run

Serious and critical errors
No serious or critical errors

BurnInTest Certificate

Report Date	12/15/08
Customer	ian
Technician	tom
Generated by	BurnInTest Version V6.0 Pro

System summary

System component	Description
Computer Name	XW6400XP32
Operating system	Windows XP Professional Service Pack 3 build 2600 (32-bit)
CPU type	Intel(R) Core(TM)2 Quad CPU Q6600 (2394.0 MHz)
RAM	2047 MB
Video card	NVIDIA Quadro FX 1400 (128MB)
Video card	ATI Radeon HD 3450 (512MB)
Disk drive	Model ST3160827AS (Size: 149.0GB)
Disk drive	Model WDC WD800JD-60LSA0 (Size: 74.5GB)
Optical drive	TSSTcorp CD/DVDW SH-W162Z

Result summary

Test Start time	Mon Dec 15 18:27:41 2008
Test Stop time	Mon Dec 15 18:28:43 2008
Test Duration	000h 01m 02s

Test	Result
CPU	PASS
Memory (RAM)	PASS
2D Graphics	PASS
Disk (C:)	PASS
Optical disk (E:)	PASS
Sound	PASS
Network 1	PASS

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