

# PassMark® White Paper

## Using BurnInTest sample applications



**Edition:** 3.0  
**Date:** 4 May 2015  
**BurnInTest Version:** 8.0

BurnInTest is a trademark of PassMark software

## Contents

|   |   |
|---|---|
| Overview .....  | 3 |
| The sample applications .....                           | 4 |
| BurnInTest setup.....                                   | 4 |
| Pre-test Plugin test with user interaction (a GUI)..... | 4 |
| Plugin test .....                                       | 4 |
| Post test.....  | 4 |
| Demonstration script.....                               | 4 |
| Setting up the sample demonstration.....                | 5 |
| Running the sample demonstration.....                   | 5 |
| Soucecode .....   | 5 |

# Overview

---

Software can be written to set up the BurnInTest environment prior to testing, to run customer hardware specific tests (with and without user interactions) and to do some post processing of results.

This document provides a summary of how to use the BurnInTest sample applications for these purposes. These applications demonstrate:

1. A BurnInTest setup application used to set BurnInTest parameters such as the Serial number;
2. A pre-test Plugin test with user interaction (a GUI) for customer specific hardware;
3. A Plugin test for customer specific hardware;
4. A post-test application to extract and display the results from BurnInTest.

These are demonstrated via a script.

Sample applications require BurnInTest Professional V7.0 or above (for version 4 of the Plug-in interface).

# The sample applications

---

## *BurnInTest setup*

Creates a subscript file, bit-script-input.txt. This is a subset of script commands that will be run before a test. The sample sets the following:

```
SETSERIAL "1234-shdfgdhs-GHGHG"  
SETMACHINETYPE "Dell XPS800"  
SETNOTES "Test notes defined by the external application."  
SETLOG "\\Program Files\Plugin\plugin_log\"
```

## *Pre-test Plugin test with user interaction (a GUI)*

Shows an example of a pre-test plugin to test customer specific hardware where manual interaction is required. For example, pressing a button or visual confirmation of an action.

## *Plugin test*

Shows an example of an automated plugin to test customer specific hardware where no manual interaction is required. For example, testing non-standard radio hardware.

## *Post test*

Shows an example of an application that is run when the test fails and some test parameters are passed to the application. In the sample, the parameters passed to the application are just displayed for some time.

## *Demonstration script*

# Set up the test environment that is not known to the script, but to an external application.

```
EXECUTEWAIT "\\Program Files\BurnInTest\bitsetup.exe"
```

# Configure that at the end of the test, if it fails, run the post test application and pass the results to this application

```
SETPOSTTEST AUTO_STOP_F runapp RUN_F "\\program files\burnintest\posttest.exe $RESULT $SERIAL $MACHINETYPE $NOTES"
```

# Run an a pre-test plugin with user interaction, then an automated test.

```
SETPLUGIN PLUGIN1 "\\program files\burnintest\PluginGUI.exe" PRETEST1 yes
```

```
SETDURATION 1
```

```
SETPLUGIN PLUGIN2 "\\program files\burnintest\Plugin.exe" PRETEST2 no
```

```
LOG "Run plugin demonstration"
```

```
RUN PLUGIN
```

## Setting up the sample demonstration

---

Copy the sample files to the BurnInTest directory, e.g. C:\Program Files\BurnInTest. This includes:

- plugintest.bits (a script to demonstrate the use of all of the samples)
- bitsetup.exe (the BurnInTest setup application)
- PluginGUI.exe (the Pre-test Plugin with user interaction (a GUI))
- Plugin.exe (the plugin test)
- posttest.exe (the post test application)

## Running the sample demonstration

---

Within BurnInTest, execute the plugintest.bits script by selecting the menu item "Test", "Execute script" and browsing to \Program Files\burnintest\plugintest.bits.

## Soucecode

---

The source code for bitsetup.exe, PluginGUI.exe, plugin.exe and posttest.exe is provided and is written in C++. Each of the projects has been built using Microsoft Visual Studio 2013.