MP3 Player dScope Quick Test Notes

By Ian Heaton, Application Support Engineer, Prism Media Products Ltd 27 March 2009

INTRODUCTION

The MP3 player script tests a range of parameters very rapidly, indicates Pass or Fail, and saves the results to a spreadsheet.

The basic principle is that the player is connected to the analogue inputs of the dScope and a test signal played from the MP3 player. The test signal consists of three sections:

-3dBFS 1kHz Sine on both channels
-60dBFS 1kHz Sine on both channels
Special 5 tone Multi-tone.

The script recognises the start of the audio and keeps in step with these stages. It makes a total of 18 measurements in typically about 6 seconds (depending on the signal itself and the speed of the computer running the dScope software).

HARDWARE REQUIREMENTS

Prism Sound dScope III
PC with Windows 98SE or later operating system and USB connection

SOFTWARE REQUIREMENTS

dScope version 1.21 or later
Microsoft Excel

INSTALLATION

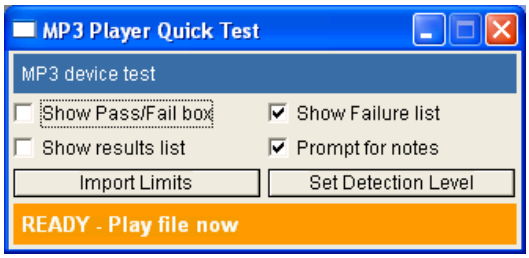
The files are supplied in a zip file with a script installer which creates the necessary folders and copies files to the required location. This script reads the dScope installation folder location from the registry, then installs the necessary files, creating new folders where necessary. To install the files, unzip the contents of the zip file to an empty folder (location is not important) and double click on the file “install test files.vbs”. This may get stopped by antivirus software as it uses the registry to get the path to the dScope folder. If so, you will need to tell the antivirus software to allow the script to run. Once it has run, it generates a text file called “report.txt” which lists the actions taken by the script, including where it has installed files. To un-install the scripts, you would need to look at this report and remove the files and folders it has copied and created. In this instance the installer may create one folder (depending on whether it exists already) and copies about 11 files. It makes no changes to the registry.

File Name	**Description**	**Location** (relative to dScope folder)
MP3 Player Quick Test.dss	Main dScope automation script	\scripts\automation\
MP3 Player Quick Test.dsc	Main dScope configuration	\configurations\
MP3 Player Quick Test Data.xls	Results and limits Excel spreadsheet	\Results\
Additional Files / Source files*:		
MP3 Quick Test xxxs.mp3	MP3 Test signals x 7	\User Wavetables\
1kHz 0dBFS Set-Up File.MP3	Utility test tone	\User Wavetables\

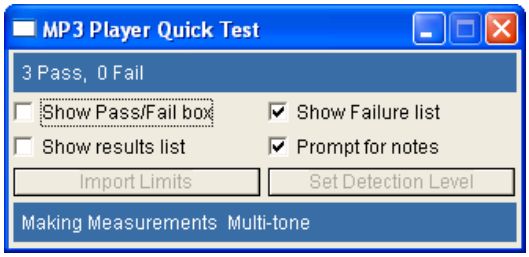
*The location of this file can be changed, but it will need to be changed in the automation script too. See notes later for how to do this.

QUICK START GUIDE

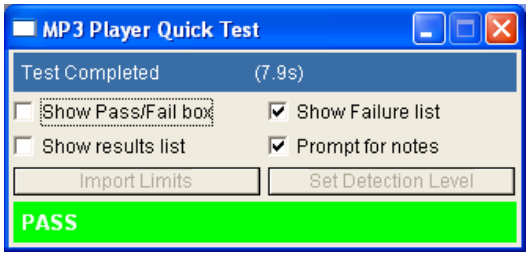
The Test should require very little configuration to get it running. Once installed you can simply run the script either clicking on the "run script" icon on the toolbar or selecting "run script" from the "Automation" menu and selecting the file "MP3 Player Quick Test.dss". Running this file will open the load the right dScope configuration, open the spreadsheet, import the limits, and then wait for you to play a test signal from your device under test with an interface that looks like the below.



Connect the analogue outputs from the device to the dScope's analogue inputs, using the adaptors provided if appropriate, then play the test file. If the signal coming from the device is loud enough, the dScope will detect it and the tests will start automatically. If nothing happens, most likely this is because the level is too low. The default detection level is 0.2V. You can change it by clicking on the "Set Detection Level" button and entering a new value. Once the signal is detected, the test will proceed to make the measurements…



And a short time later, all being well, you will see a "Pass" screen as shown below.



After saving the results, the test will reset and is ready for another unit.

SETTING THE TEST LIMITS

The test limits are set in the results spreadsheet and are in the units indicated. Simply edit them as they appear on the page. There are some columns that do not have limits above them. These columns are typically the right channel readings and will share the limits used for the left channels. The limits are read by the script when the test is first run. Whether a unit passes or fails is then determined by the script based on these values. The pass/fail is then written back to the spreadsheet by the script. This means that the pass/fail in the spreadsheet refers to the limits in force at the time the test was conducted: Changing the limits will not change the pass/fail results for measurements already taken, but only for units tested after the changes are made.

If you change the limits in the results spreadsheet after the test has started, you will need to re-import them to the test before they will take effect. You can do this by clicking the "Import Limits" button. Note that the limits are imported when the script is first started, so this is only necessary if you subsequently change them.

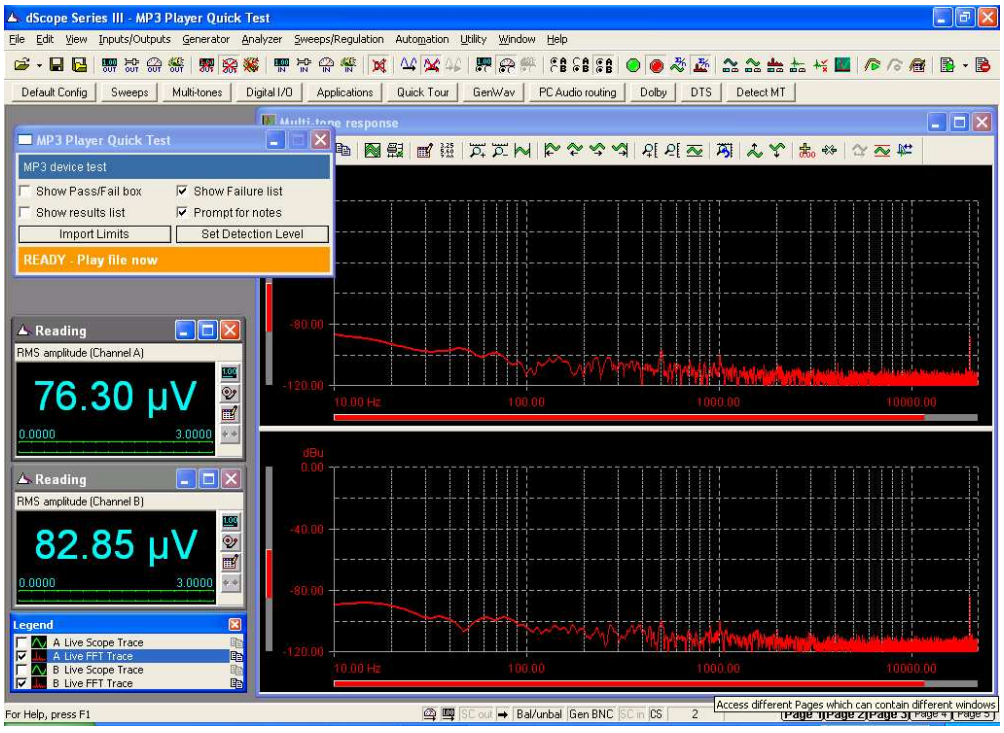
The spreadsheet has conditional formatting set up for the first 50 records. This is set so that any readings that fall outside the limits are formatted in red. This works dynamically so that it is possible to change the limits and see which parameters would have failed. Only the first 50 records (lines 8 to 58) are formatted this way as it makes the spreadsheet very large and greatly increases the processing time for the script.

When setting limits and levels, in order to get a good idea of what the limits should be, it may help to measure a number of known good units and look at the results in the spreadsheet to see what the average reading and the spread of measurements is. You can then adjust the limits to suit.

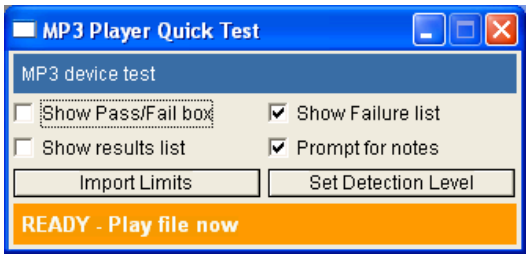
RUNNING THE TESTS

Once the files are installed the test would normally be run as follows:

1) Start the dScope software
2) Make sure the results spreadsheet is not open in any other application.
3) Run the script file: the dScope configuration will load and the screen below will appear:

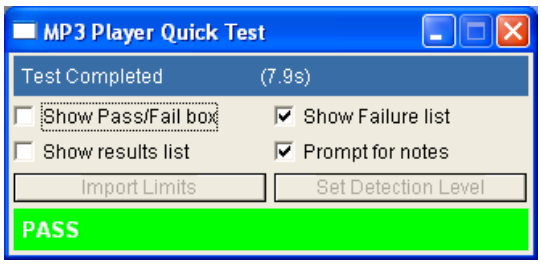


The Excel spreadsheet may also appear at this stage, and can be minimized. (Don't close it!). On the dScope screen are two level meters, one for each channel. To the right are the frequency spectra for the multi-tone tests. To the top left is the Test Status box as shown below.

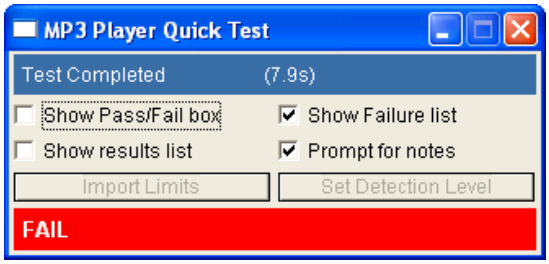


4) Load the MP3 file into the unit to be tested
5) Connect the unit to be tested to the analogue inputs of the dScope and set the volume to an appropriate level
6) Wait until prompted that the script is ready (orange prompt as above)
7) Play the file from the MP3 Player

The test should start and run automatically
A pass/fail screen will be shown:



or



8) At this point, what happens will depend on which boxes are checked in the interface.

- If the "Show Pass/Fail" box is checked, a separate box with Pass or Fail will appear. This has to be actively cleared by clicking on the "OK" button.
- If the "Show Failure List" box is checked, and the device has failed, a message box will appear stating which parameters have failed.
- If the "Show results list" box is checked, two message boxes will appear which show the results of all the tests.
- If the "Prompt for Notes" box is checked, an input box will appear to allow you to enter notes or the serial number of the unit that has just been tests. This will be added to the spreadsheet.

The results will now be saved to the spreadsheet and the script will reset ready for the next unit.

9) When all the units have been tested, simply close the user interface using the windows "x" in the top right corner. This asks you if you want to save the spreadsheet, and then if you want to close it. If you say yes to either question (the default), the script will do this for you.

SELECTING A TEST TRACK TO USE

The only difference between the test tracks is the length of the first two sections and this is indicated in the name of the track. The track MP3 Quick Test 1_5s.MP3 has the first two sections only lasting 1.5 seconds each. The complete test should run in about 5.5 seconds. The 2 second track should run in about 6.5seconds and the 2.5 second track in about 7.5 seconds.

The reason there is a selection of tracks is that it is impossible to know in advance how fast the computer that is running the test will be able to make the measurements. If it is a fast computer it will be able to measure with the 1.5 second test track. If a slow computer is used with this test track, there will be reading errors. The script is written in such a way that it checks that the readings are taken in the correct part of the test track. If the test track has moved on to the next stage before the dScope has made the measurements, the script will flag it up as a reading error and abort the test. The test will then have to be re-started. If this happens too often, it will be better to move to a longer test track to allow the computer longer to make the measurements. There is no way to know which to use other than to try them. It is probably best to start with the shortest track and, if errors occur, switch to a longer track or make changes to the PC to allow it to run faster.

SETTING THE DETECTION LEVEL

In order for the dScope to be able to detect the start of the test track and keep in time with it, it needs to know the level of the signal that it is to be detecting. By default the script is set to a level of 0.2 Volts RMS. The test signal has three stages, a -3dBFS 1kHz sine wave, a -60dBFS 1kHz sine wave, and a multi-tone with an RMS level approximately half the level of the sine wave. The detection level must be such that the level of background noise never triggers it when the track isn't playing, nor should the playing the -60dBFS signal. The level from the 1kHz tone and the level from the multi-tone should both be above the detection level. In practice with any device with a reasonable signal to noise ratio, there should be no problem finding a level that works reliably. Once you have found a detection level that works consistently, you can change the default value that is loaded when the script is first started. Look for the line that says.

```
dSigDetect        = 0.2
```

and set it to your new value.

CHANGING THE SPREADSHEET NAME AND PATH

The default path and name of the results spreadsheet can be changed in the script. Make sure the spreadsheet exists in the location desired then edit the script "MP3 Player Quick Test.dss" to show the new location. Look for the lines:

```
'Default Excel spreadsheet file location – relative to the dScope installation
StrXLfile = "Results\MP3 Player Quick Test data.xls"
```

To change the path, simply edit the name to the new location. If you change this to a full path (not relative to the dScope installation), there are a couple of other places in the script where this is treated as relative to the dScope installation and these will need to be changed.

> NOTE: Be careful not to change the FORMAT of the spreadsheet by adding columns or rows. The script references the cells directly and changes in the spreadsheet format will mean that the script will not function correctly unless it is edited to keep the references in line.

OPTIONS WITHIN THE SCRIPT

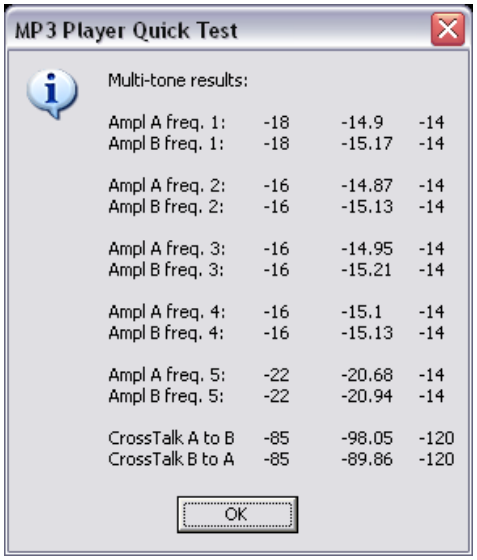
The script can be configured with various options in addition to the ones on the user interface. These are set in the section of the main automation script “MP3 Player Quick Test.dss” in the section “User options”. Each option is set by a variable which can be set true or false. The options are as follows:

Show Option Checkboxes on the user interface: this is controlled by the variable “bShowCheckboxes”. If set to true, there will be four checkboxes as shown in the screenshots previously. If set to false, the various options set by the checkboxes will be configured by the next four options.

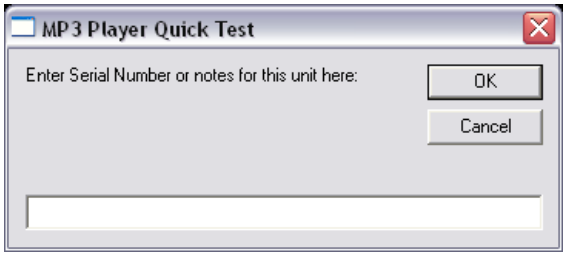
Show Pass/Fail box: This is controlled by the variable “bPassFailbox” and brings up an additional larger Pass / Fail indicator which must be cleared by clicking the “OK” button. If user interface checkboxes are enabled, this sets the default setting of the “Show Pass/Fail box” checkbox when the script is started. If checkboxes are not enabled, this sets the behaviour of the script. The Pass/Fail boxes created in this way are shown below.

Show Failure List. In the event that a device fails the test, it is possible to have the script show a list of the parameters that have failed. This is controlled by the variable “bShowFailures” If user interface checkboxes are enabled, this sets the default setting of the “Show Failures” checkbox when the script is started. If checkboxes are not enabled, this sets the behaviour of the script. An example of the failure list is shown below.

Show Results list. It is possible to get the script to show you all the measurements and their respective limits in a couple of message boxes after each test. This is controlled by the variable "bShowData" If user interface checkboxes are enabled, this sets the default setting of the "show results list" checkbox when the script is started. If checkboxes are not enabled, this sets the behaviour of the script. The second part of the results list showing the multi-tone results is shown below.



Prompt for Notes or Serial Number. After each test, the script can bring up an input box to enter notes or serial number information about the test that has just been conducted. This is controlled by the variable "bNotesBox" If user interface checkboxes are enabled, this sets the default setting of the "Prompt for notes" checkbox when the script is started. If checkboxes are not enabled, this sets the behaviour of the script. The input box is shown below.



Error Logging: In the event that the script doesn't run correctly (i.e., fails to detect the changes in level etc.) the script can write the errors to a text file error log. This is controlled by the variable "bLogErrors". It is recommended to leave this enabled as it is a valuable source of debugging information and takes almost no time to execute. The path to the log file can also be set by the variable "strLogFile". By default this is set to save a file called "MP3 Quick Test Error Log.txt" in the dScope installation folder. It can be changed to anything.

Run with Excel visible: By default, this script opens Excel visible and available for editing. This is set by the variable "bShowExcel" – set it to false to run with Excel in the background.

Show Session Statistics. At the end of each test session, the script can give the statistics of how many devices were tested, how many passed and how long the average test took. This is enabled by the variable "bShowSessionStats". By default this is off.

TROUBLESHOOTING

The two main problems likely to be encountered are the audio levels and the speed of the process. Since the MP3 file will play back at a precise speed, it is essential that the computer running the dScope is fast enough to make the measurements in the time allowed. Every effort has been made to make sure that the test will not give a false pass on a bad unit or a false fail on a good one. The side effect of this is that it can be surprisingly difficult to get the test to run at all! If the playback levels are too low the script will never start as it relies on detecting levels to run. If it runs too slowly, the computer will still be trying to measure amplitude while the test track has moved on to play the signal to noise section and the script will abort the test.

Spreadsheet Problems

If the Excel spreadsheet used to save results is open when the script tries to open it, the script will open a read-only copy. It will warn you of this as it starts up. The script can still write to the spreadsheet, but it can not save the results. If you want to save them, you will have to leave the spreadsheet open when the script closes and save it manually with a new name.

Anti-virus software and scripting

If a test fails to run correctly and if the error log is enabled, the script will attempt to make a text log file in the location c:\ MP3 Quick Test error log.txt. This is done by the script using the Windows File System Object to open a text file to write to. Accessing the File System Object using a script is a process also used by viruses to attack computers and for this reason it is possible that anti-virus software will stop this process and flag it up as a possible attack. You should satisfy yourself that this is what is happening (have a look at the main script file for the process that is being used and read the anti-virus message carefully) and then, when you are happy, allow the script to run. If you prefer not to allow the script to make a log file, this can be disabled in the script. See previously under “Options within the Script”

Technical Reference
The parameters measured by this test are:

Amplitude

Source: -3dBFS 1kHz stage of test signal
Units: Volts
Filters: None
Instrument: Signal Analyzer RMS amplitude meter

THD+N

Source: -3dBFS 1kHz stage of test signal
Units: Percent, self relative
Filters: 1/3rd Octave band reject, 22Hz high pass, 22kHz Low Pass
Instrument: Continuous Time Detector

Signal To Noise Ratio

Source: -60dBFS 1kHz stage of test signal
Units: dB self-relative adjusted to assumed full scale
Filters: 1/3rd Octave band reject, 22Hz high pass, 22kHz Low Pass
Instrument: Continuous Time Detector
Note: this is done by measuring THD+N at –60dBFS and adding 60dB to the result

Frequency Response

amplitudes 20Hz, 250Hz, 1kHz, 4kHz (L), 8kHz (R) , 20kHz
Source: Multi-tone section of test signal
Units: Volts RMS
Filters: window notch band pass,
Instrument: FFT detectors

Cross-talk

Source: Multi-tone section of test signal
Units: dB relative to opposite channel
Filters: window notch band pass,
Instrument: FFT detectors

Note on cross-talk: This measurement is done in both directions simultaneously by having the cross-talk from A to B measured at 4kHz and from B to A at 8kHz. The multi-tone is encoded with a 4kHz tone on the left channel only and an 8kHz tone on the right channel only. Because of the MPEG encoding and the presence of noise skirts around the tones, these signals have had to be separated by a significant frequency margin. Normally with a multi-tone and a synchronous measurement system (which this isn't) it is possible to measure cross-talk at nearly the same frequency in both directions simultaneously. In this case, to measure at the same frequency would require two additional measurement signals and would approximately double the test measurement time. Because of the frequency dependency of most cross-talk problems, we can not expect the cross-talk to be the same in both directions.