

USB Mass Storage Device Test Guide

Version 2.0, April 30, 2010

This document is a guideline to basic USB Mass Storage device testing. The objective of this testing is to see if there are any failures when reading from or writing to the device. It carries out a regression test, which transfers data to the USB Mass Storage device, reads back the data, and compares for checking data integrity issues. The codes for windows based systems and Linux based systems are different. Examples of such codes are given here.

Windows based System Testing

- Test Setup

Create a directory structure for testing as follows:

- *C:\MSCtest* – Mass Storage Class device test folder
- *C:\MSCtest\log.txt* – empty text file, holds the log output from the test script
- *C:\MSCtest\src* – holds data files of various sizes depending on user preference
- *C:\MSCtest\dst* – empty folder, the script uses this for copy back and compare
- *C:\MSCtest\test.bat* – the Test script (example given below)

- Test code: (test.bat)

```
@ECHO OFF
set number=0
:start_loop
set /A number=%number%+1
echo running test number %number%
echo RUN %number% >> C:\testfiles\log.txt

XCOPY C:\MSCtest\src\*. * F:\ >> C:\MSCtest\log.txt
echo ----- >> C:\MSCtest\log.txt
fc /b F:\*. * C:\MSCtest\src\*. * | FIND "FC: no diff" >>
C:\MSCtest\log.txt
IF ERRORLEVEL 1 GOTO end
echo ----- >> C:\MSCtest\log.txt
XCOPY F:\*. * C:\MSCtest\dst\ >> C:\MSCtest\log.txt
echo ----- >> C:\MSCtest\log.txt
fc /b C:\MSCtest\dst\*. * C:\MSCtest\src\*. * | FIND "FC: no diff" >>
C:\MSCtest\log.txt
IF ERRORLEVEL 1 GOTO end
echo ----- >> C:\MSCtest\log.txt
del /Q F:\*. *
del /Q C:\MSCtest\dst\*. *

goto start_loop
:end
```

- In case there are errors, the script execution stops. "log.txt" will show where the execution stopped and why. This script is set to run till there are errors; therefore, in absence of errors it continues to run.

LINUX based system testing

- **Test Setup**

To run the test with a LINUX file system, for example EXT2, disable formatting of the NAND device before programming the NAND device. (toggle formatting on/off by ctrl+f)

Plug in the NX2LP DVK with the programmed, unformatted NAND device into a LINUX system

Format the NAND device and load EXT2 file system on it. Use the following steps as guidelines.

- Run terminal.
- Check which drive is the NX2LP device connected as by command 'dmesg'. For example, the device is connected as sdb.
- Format the disk by running command 'fdisk /dev/sdb'. If the NAND device is 4K-page device, use the command 'fdisk -b 1024 /dev/sdb'.
- Use options 'n' for new partition, 'p' for primary partition, '1' for the first partition, and rest default settings.
- When exiting 'fdisk', exit with save changes (option 'w')
- After successful format, write the ext2 file system by command 'mkfs -t ext2 /dev/sdb1'

Example directory structure as follows for test setup:

- */root/Desktop/test* – Mass Storage Class device test folder
- */root/Desktop/test/Test1* – Source folder for the test data (files+folders)
- */root/Desktop/test/Test2* – Mount directory for the NX2LP device
- */root/Desktop/test/log.txt* – empty text file, holds the log output from the test script
- */root/Desktop/test/TestScript* – the test script (example given below)

- **Test Code:**

```
#!/bin/bash
mount /dev/sdb1 /root/Desktop/test/Test2
cd /root/Desktop/test/Test2
rm -rf *
cd ..
counter=0
echo Start Test > /root/Desktop/test/log.txt
while [ $counter -lt 1000 ]; do

let counter+=1;
echo counter=$counter | tee -a /root/Desktop/test/log.txt

cd -r Test1/* Test2/
echo Files copied

if [ "$(diff -qr Test1 Test2 | tee -a /root/Desktop/test/log.txt)" = ""
];
then
echo No differences found | tee -a /root/Desktop/test/log.txt
else
echo Data integrity Issues | tee -a /root/Desktop/test/log.txt
break
fi
cd Test2
rm -rf *
```



```
echo All Files removed | tee -a /root/Desktop/test/log.txt  
cd ..
```

done

- This code runs until there is an error or 1000 iterations. On error the script exits and type of error can be found out from the log.txt file.