Code 582
Flight Software Branch

CORE FLIGHT SYSTEM
FILE MANAGER APPLICATION
BUILD 2.5.4.0

FLIGHT SOFTWARE BUILD VERIFICATON TEST REPORT

Flight Software Branch – Code 582

Version 1.0

SIGNATURES

Submitted by:

12/3/2020

X Walt Moleski

Walt Moleski/582
cFS Flight Software Tester
Signed by: WALTER MOLESKI

Approved by:

12/3/2020

X Elizabeth Timmons

Elizabeth Timmons/582
GSFC cFS Development Lead
Signed by: cards

PLAN UPDATE HISTORY

Version	Date	Description	Affected Pages
1.0		Initial release	All

TABLE OF CONTENTS

1 INTRODUCTION

1.1 DOCUMENT PURPOSE

This Test Report describes the test results from the core Flight System (cFS) File Manager (FM) Flight Software (FSW) Test Team builds 2.5.4.0 verification testing.

BVT is used to verify that the FM FSW has been tested in a manner that validates that it satisfies the functional and performance requirements defined within the cFS FM Requirements Document. This Test Report summarizes the FSW test history, the build verification process, the build test configuration, and the test execution and results.

1.2 APPLICABLE DOCUMENTS

Unless otherwise stated, these documents refer to the latest version.

Parent Documents (Mission and FSW)

- 582-2007-032 cFS File Manager Requirements Document, Version 1.5
- 582-2008-012 cFS Deployment Guide, Version 3.0

Reference Documents

All of the references below can be found on the Code 582 internal website at http://fsw.gsfc.nasa.gov/

- 582-2003-001 FSB FSW Test Plan Template
- 582-2004-002 FSB FSW Test Scenario Template
- 582-2004-003 FSB FSW Test Procedure Template
- 582-2004-005 FSB Test Product Peer Review Form

1.3 DOCUMENT ORGANIZATION

Section 1 of this document presents some introductory material.

Section 2 provides a flight software overview and context along with the test history and testing overview.

Section 3 describes the build verification process including procedure development and execution and test products produced.

Section 4 describes the build test configuration which includes an overview of the testbed and the requirements verification matrix.

Section 5 describes the test execution and results by subsystem.

Appendix A - provides the Requirements Traceability Matrix

Appendix B - provides the Command, Telemetry, and Events Verification Matrix

1.4 DEFINITIONS

There were 3 verifications methods used during build verification testing. They were:

- Test: Show compliance with system requirement by exhibiting the required capability (e.g. by demonstrating interactive capability, display capability, print capability, etc.
- Inspection: Show compliance with a system requirement by visual verification of the software (e.g. verifying preparation for delivery, proper interfacing)
- Analysis: Perform detailed analysis of code, generated data (both intermediate data and final output data), etc., to determine compliance with system requirements.

The fields in the Requirements Verification Matrix in Section 4.3 are defined as follows:

- Requirements Tested Passed: Requirement was fully tested in a build test procedure and passed all tests.
- Requirements Tested Failed: Requirement was fully tested in a build test procedure and failed one or more aspect of the testing.
- Requirements Tested Partially: Requirement was tested partially in a build test procedure. To be fully tested, the partially tested requirement is either tested additionally in one or more other test procedures within the same build and/or other aspects of the requirement must be tested in a later build, due to capabilities not present in the current build
- Total Tested: Total number of requirements fully tested in a build test procedure. Includes total passed and total failed, but does not include requirements tested partially, unless (included as a separate entry) testing in multiple procedures within the same build constitutes total testing of a particular requirement. Total Requirements Tested is computed this way in order to avoid multiple counting of individual requirements that are tested partially in more than one procedure.
- Deferred: Number of requirements that were planned to be tested in current build but were not tested due to some FSW capability or necessary system component not being present.
- Total: Total Requirements Tested + Number of Requirements Deferred

Section 5 contains a table of DCR's addressed by this testing effort.

2 OVERVIEW

2.1 FLIGHT DATA SYSTEM CONTEXT

Figure 2-1 illustrates the cFS system context. The cFE interfaces to five external systems: an Operating System (OS), a Hardware Platform (HP), an Operational Interface (OI), Applications (APP), and other cFE-based systems.

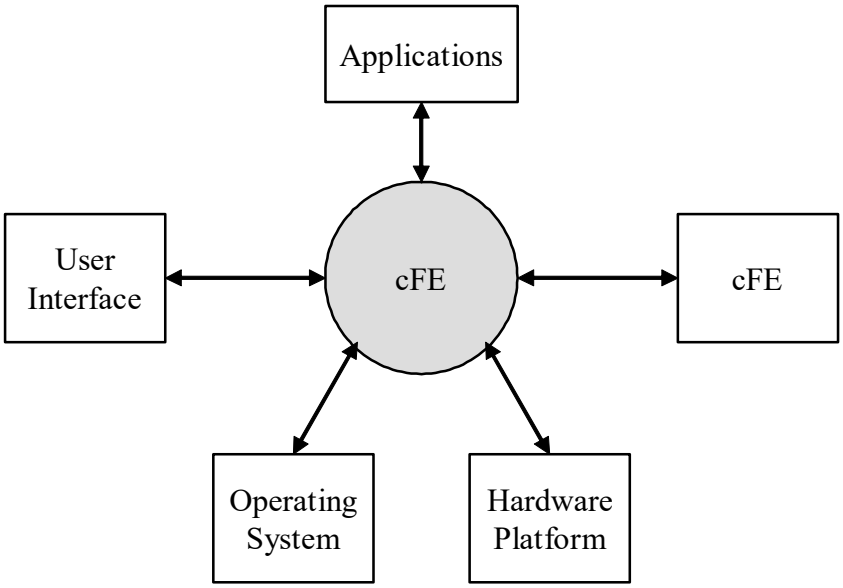

Figure 2-1 cFS System Context

The cFS File Manager (FM) application provides a ground interface for managing onboard file systems. The application file management services to the ground include copying files, moving or renaming files, deleting files, decompressing files, concatenating files, retrieving file and directory status information, creating directories, removing directories, and retrieving directory listings.

The cFS FM context shows use of a complete cFS, presenting interfaces with other cFS applications. SCH is the cFS scheduler application that submits periodic housekeeping requests to FM. Commands come from the cFS Command Ingest application (CI). Event messages and housekeeping packets are routed to the appropriate cFS output application, the Housekeeping (HK), Telemetry Output (TO), and/or Data Storage (DS) application. All accesses to the file system(s) are through the OS API layer of the cFE.

File systems can exist on RAM and EEPROM as well as custom devices such as a Solid State Recorder (SRR). The OSAL provides the interface to the file systems on any available devices. Custom devices such as SSRs will be handled outside of FM (potentially by another application).

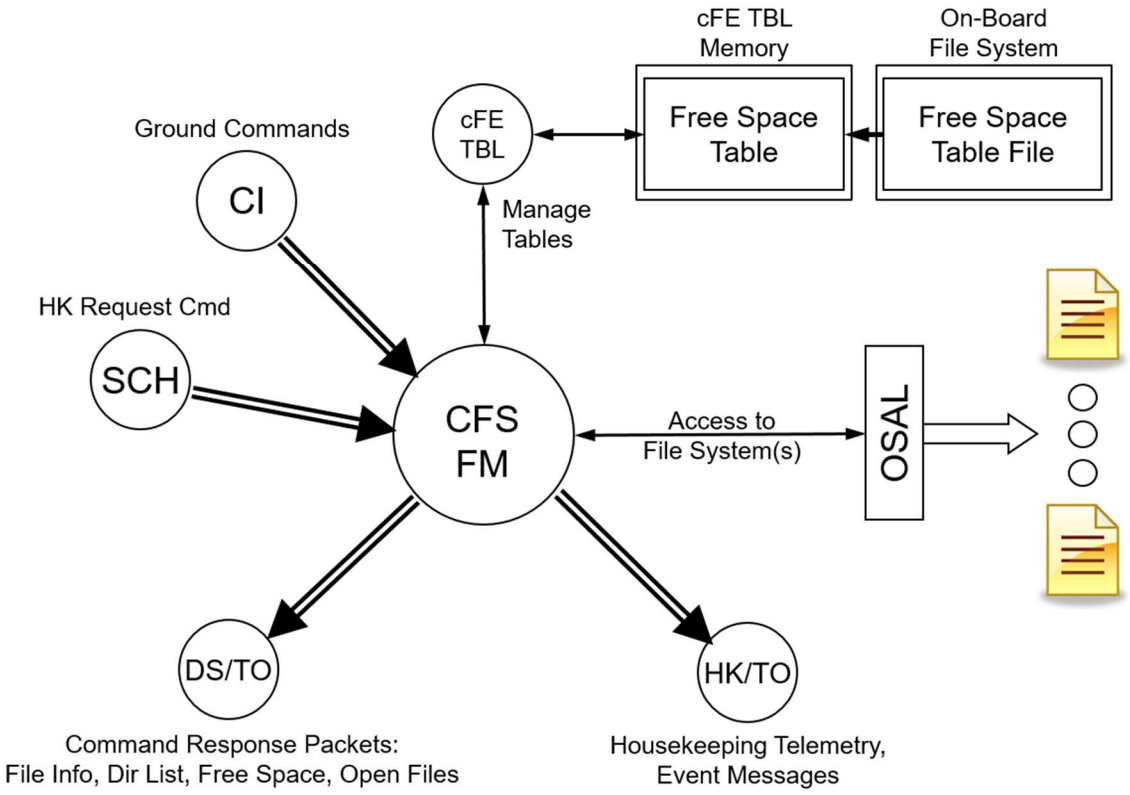

Figure 2.2 – cFS FM Context

2.2 TEST HISTORY

FM 1.0.0.0 – Build Verification Testing completed 12/16/2008 by Damon Stewart
FM 2.0.0.0 – Build Verification Testing completed 9/25/2009 by Walt Moleski
FM 2.1.0.0 – Build Verification Testing completed 5/17/2010 by Walt Moleski
FM 2.2.0.0 – Build Verification Testing completed 8/24/2010 by Walt Moleski
FM 2.3.1.0 – Build Verification Testing completed 5/9/2012 by Walt Moleski
FM 2.4.1.0 – Build Verification Testing completed 1/12/2015 by Walt Moleski
FM 2.4.2.0 – Build Verification Testing completed 1/22/2015 by Walt Moleski
FM 2.5.0.0 – Build Verification Testing completed 1/19/2017 by Walt Moleski
FM 2.5.1.0 – Build Verification Testing completed 1/25/2017 by Walt Moleski
FM 2.5.2.0 – Build Verification Testing completed 1/26/2017 by Walt Moleski
FM 2.5.3.0 – Build Verification Testing completed 4/2/2020 by Walt Moleski
FM 2.5.4.0 – Build Verification Testing completed 11/19/2020 by Walt Moleski

2.3 TESTING OVERVIEW

The FM application was tested during Build Verification testing using the following:

- 1 test application: tst_fm
- 24 main test procedures (listed below)
- 7 test procedures that are called by the main procedures (listed below)
- All tests require the ASIST Ground Station

The TST_FM test application is used to send schedule requests for the output of FM's housekeeping data. TST_FM has 5 ground commands that are used by the FM test procedures:

- TST_FM_NoOp

    - Input: none
    - Output: Command Accepted counter increased, info event message issued
- TST_FM_ResetCtrs
    - Input: none
    - Output: Command Accepted and Rejected counters set to zero, debug event message issued
- TST_FM_Open
    - Input: File name of the file that you wish to open
    - Output: Command Accepted counter increased, debug event message issued
- TST_FM_Close
    - Input: File name of the file that you wish to close
    - Output: Command Accepted counter increased, debug event message issued
- TST_FM_Delete
    - Input: File name of the file that you wish to delete using the FM internal delete command
    - Output: Command Accepted counter increased, info event message issued

These 24 main FM test procedures do the following:

Procedure	**Description**
fm_gencmds	The purpose of this test is to verify that the File Manager (FM) general commands function properly. The FM_NoOp and FM_Reset commands will be tested as well as invalid commands and an application reset to see if the FM application behaves appropriately.
fm_filerename_basic	The purpose of this test is to verify that the File Manager (FM) File Rename Command functions properly. The FM_FileRename, and FM_FileInfo commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DirCreate command is also used to facilitate the testing.
fm_filerename_stress	The purpose of this test is to stress the File Manager (FM) File Rename Command function. The FM_FileRename command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_FileInfo and FM_DirCreate commands are also used to facilitate the testing but are not stressed in this scenario.
fm_filemove_basic	The purpose of this test is to verify that the File Manager (FM) File Move Commands function properly. The FM_FileMove, and FM_FileInfo commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DirCreate command is used to facilitate the testing.
fm_filemove_stress	The purpose of this test is to stress the File Manager (FM) File Move Command function. The FM_FileMove command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_FileInfo and FM_DirCreate commands are also used to facilitate the testing but are not stressed in this scenario.
fm_fileinfo_basic	The purpose of this test is to verify that the File Manager (FM) File Info and File Close Commands functions properly. The FM_FileInfo and FM_Close commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DirCreate command is also used to facilitate the testing.

Procedure	Description
fm_fileinfo_stress	The purpose of this test is to stress the File Manager (FM) File Info Command function. The FM_FileInfo command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_DirCreate command is also used to facilitate the testing but is not stressed in this scenario.
fm_filedecom_basic	The purpose of this test is to verify that the File Manager (FM) File Decompress Command functions properly. The FM_FileDecompress and FM_FileInfo commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DeleteAll, FM_DirCreate, FM_DirDelete, and FM_DirListTlm commands are used to facilitate the testing.
fm_filedecom_stress	The purpose of this test is to stress the File Manager (FM) File Decompress Command function. The FM_FileDecompress command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_FileInfo and FM_DirCreate commands are also used to facilitate the testing but are not stressed in this scenario.
fm_filecopy_basic	The purpose of this test is to verify that the File Manager (FM) File Copy Commands function properly. The FM_FileCopy and FM_FileInfo commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DirCreate command is also used to facilitate the testing.
fm_filecopy_stress	The purpose of this test is to stress the File Manager (FM) File Copy Command function. The FM_FileCopy command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_FileInfo and FM_DirCreate commands are also used to facilitate the testing but are not stressed in this scenario.
fm_filecat_basic	The purpose of this test is to verify that the File Manager (FM) File Concatenate Command functions properly. The FM_FileCat and FM_FileInfo commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DeleteAll, FM_DirCreate, FM_DirDelete, and FM_DirListTlm commands are used to facilitate the testing.
fm_filecat_stress	The purpose of this test is to stress the File Manager (FM) File Concatenate Command function. The FM_FileConcatenate command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_FileInfo and FM_DirCreate commands are also used to facilitate the testing but are not stressed in this scenario.
fm_dircmds_basic	The purpose of this test is to verify that the File Manager (FM) Directory Commands function properly. The FM_DirCreate, FM_DirDelete, FM_DirListFile, and FM_DirListTlm commands will be tested to see if the FM application handles these as desired by the requirements. The FM_delete command is also used to facilitate the testing.
fm_dircmds_stress	The purpose of this test is to stress the File Manager (FM) Directory Command functions. The FM_DirCreate, FM_DirDelete, FM_DirListFile, and FM_DirListTlm commands will be tested to see if the FM application handles error cases for these, both expected and unexpected.

Procedure	Description
fm_dirrename	The purpose of this test is to verify that the File Manager (FM) Application does not cause any anomalies when a directory is renamed.
fm_filedelete_basic	The purpose of this test is to verify the File Manager (FM) File Delete Command functions. The FM_Delete and FM_DeleteAll commands will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_DirCreate command is also used to facilitate the testing but is not tested in this scenario.
fm_filedelete_stress	The purpose of this test is to stress the File Manager (FM) File Delete Command functions. The FM_Delete and FM_DeleteAll commands will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_DirCreate command is also used to facilitate the testing but is not stressed in this scenario.
fm_openfiles	The purpose of this test is to verify the FM_ListOpenFiles command.
fm_specialchars1 fm_specialchars2 fm_specialchars3 fm_specialchars4 fm_specialchars5	The purpose of these tests are to stress the File Manager (FM) Command functions that have a directory or filename as an argument. The arguments are setup to contain special characters and the procedure documents which characters are valid and which ones are not. The five tests are as follows: 1. The special characters by themselves. 2. The special characters as the first character. 3. The special characters in the middle 4. The special characters as the last character 5. The special characters in the file extension for filename arguments only.

The 7 test procedures described in the table below are called by the main test procedures.

Procedure	Description
fm_startfmapps	Starts all needed applications and opens all FM Tlm pages
fm_clearallpages	Clears all Tlm Pages
fm_fileinfodisplay	Executes the File Info Command and displays the results for the log
fm_dirtlmdisplay	Executes the Directory Listing to Telemetry Command and displays the results for the log
fm_dirfiledisplay	Executes the Directory Listing to File Command and displays the file contents for the log
fm_tableloadfile	Builds the FM Freespace table used by the main test procedures.
fm_badtblloadfile	Creates a Freespace table image containing all possible validation errors in order to verify that these errors are properly detected.

The testers use the cFS Test Account for each build test. This account runs ASIST and is setup to contain all the files needed to test the application. Included in these files are test utilities. These utilities can be located in 2 places depending upon whether they are “local” or “global” utilities. The local utilities are extracted into the working prc directory ($WORK/prc). The global utilities are pointed to by ASIST in the global area defined on the test system. Additional tools utilized by the test procedures are located in the $TOOLS directory. It is assumed that test procedures and the ASIST telemetry database used for testing is built using procedure and database templates

The following utilities were used during testing:

Name	Description
cfe_startup	Directive combines the "start_data_center", "open_tlm", and "open cmd <cpu>" ASIST startup commands.
cfe_shutdown	Directive combines the "close_data_center" and "exit" ASIST shutdown commands.
FILE_TO_CVT	Directive that takes the contents of a file and associates it with a Current Value Table (CVT) for displaying in an ASIST Display page
ftp_file	To ftp a file to/from the FSW/GSW.
load_start_app	Procedure to load and start a user application from the $WORK/apps/cpux directory.
ut_runproc	Directive to formally run the procedure and capture the log file.
ut_sendcmd	Directive to send EVS commands Verifies command processed and command error counters.
ut_sendrawcmd	Send raw commands to the spacecraft. Verifies command processed and command error counters.
ut_setrequirements	A directive to set the status of the cFE requirements array.
ut_setupevents	Directive to look for multiple events and increment a value for each event to indicate receipt.
ut_tlmupdate	Procedure to wait for a specified telemetry point to update.
ut_tlmwait	Directive that waits for the specified telemetry condition to be met

2.4 VERSION INFORMATION

Item	Version
FM Requirements	1.5
FM Application	2.5.4.0
TST_FM Application	2.5.4.0
cFS Bundle	Bootes
CFE	6.8.0
ASIST	20.2
VxWorks	6.9

3 BUILD VERIFICATION TEST PREPARATION

3.1 SCENERIO DEVELOPMENT

No new scenarios were developed for FM 2.5.4.0 build verification tests. All scenarios are stored on the ETD GIT server, https://aetd-git.gsfc.nasa.gov/gsfc-cfs/cfs_fm in the test-and-ground/scenarios directory. It should be noted that as FM requirements evolve these scenarios are not updated to reflect any changes made.

3.2 PROCEDURE DEVELOPMENT AND EXECUTION

This build test effort was completed by running 11 of the 24 test procedures. The stress and special character tests were not executed on FM 2.5.4.0. A new test was developed named fm_deletefile_basic to test several requirements that were only contained in the fm_deletefile_stress procedure that was not executed. Various procedures were modified as a result of the FM 2.5.4.0 changes and are checked in to GIT at the conclusion of build testing. All test procedures were written using the STOL scripting language. The naming convention for files created by the test procedures was: scx_cpu<#>_<procedure name>_GMT.<ext>.

3.3 TEST PRODUCTS

Five log files were generated for every procedure that was run. They are defined as follows:

- Logs with the .loge extension list all events sent by the flight software
- Logs with the .logr extension list all requirements that passed validation by demonstration
- Logs with the .logp extension lists all prints that are generated by the test procedure
- Logs with the .logf extension lists everything from the other logs along with the steps in the test procedure
- Logs with the .logs extension lists the SFDU information (if applicable) contained in the full log.

A test summary report is developed in Jira by the tester after build testing is completed. All test products are maintained in Jira and/or GIT.

4 BUILD VERIFICATION TEST EXECUTION

4.1 TESTBED OVERVIEW

FM FSW testing took place in the cFS FSW Development and Test Facility. A high-level view of the cFS FSW Test Bed is shown in Figure 4-1. This facility is located in GSFC Building 23, Room W410N. This facility consists of two ASIST workstations running ASIST version 20.2 and three MPC750 CPU boards running VxWorks 6.9. CPU1 is primarily used for development testing while CPU2 and CPU3 are used for build verification testing.

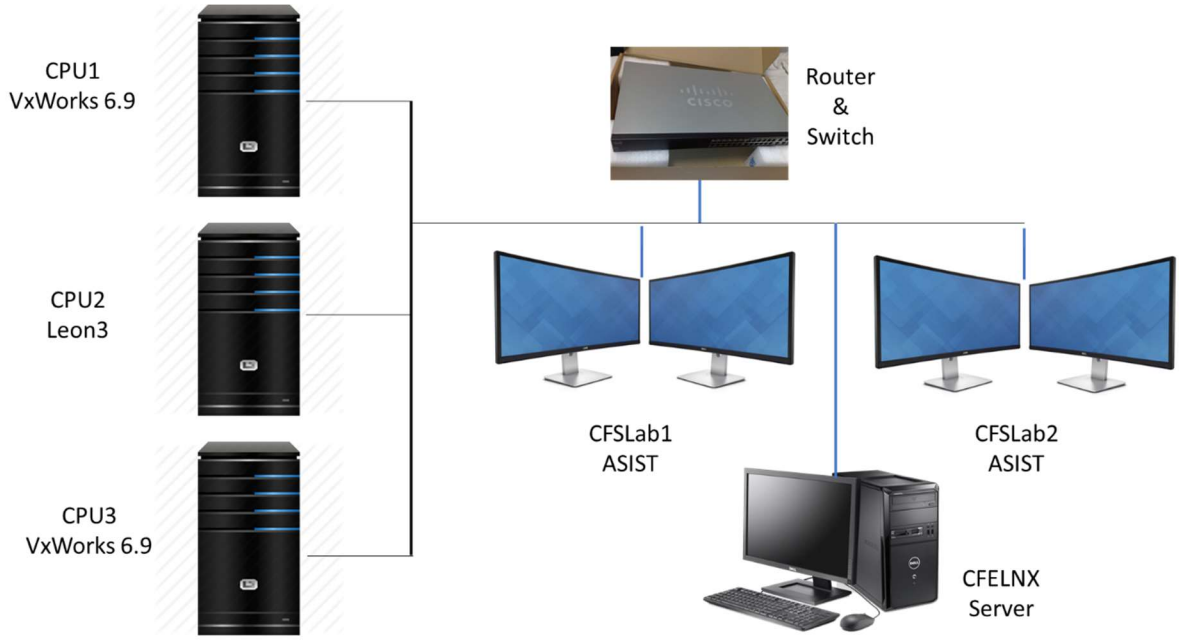

Figure 4-1 cFS FSW Development and Testing Facility

4.2 REQUIREMENTS VERIFICATION MATRIX

	File Manager (FM)
Requirements Tested Passed	45
Requirements Tested Failed	0
Requirements Tested Partially	0
Total Tested	45
Deferred/Untested	1
Total	46

4.3 REQUIREMENTS PARTIALLY TESTED

No requirements were partially tested.

4.4 REQUIREMENTS/FUNCTIONALITY DEFERRED/UNTESTED

1 requirement was not tested with FM 2.5.4:

Requirement	Requirement Text	Reason for not Testing
FM1008	The CFS FM FSW shall utilize full path specifications having a maximum length of <PLATFORM_DEFINED> characters for all command input arguments requiring file or pathname.	cFE Bootes enforces a stricter convention for the length of file names. The CFE_MISSION_MAX_FILE_LEN is now used with a default value of 20.

4.5 REQUIREMENTS/FUNCTIONALITY DEFERRED FOR MISSION TESTING

The following functionality was deferred to mission testing:

- RAM was the only physical memory type tested fully. Compact Flash was also tested in a few procedures that attempt to move or copy across memory types. EEPROM and SSR were not tested. EEPROM testing was done by simulating EEPROM in RAM.

5 BUILD VERFICATON TEST RESULTS

5.1 OVERALL ASSESSMENT

During this build test of the FM Application the software behaved as expected. Below is a summary of the results:

- 45 requirements passed via Test
- 0 requirements Failed
- 1 requirement was Deferred (See Section 4.4 above).
- 9 DCRs were validated

5.2 PROCEDURE DESCRIPTION

Procedure	Description	Requirements Tested
fm_gencmds	The purpose of this test is to verify that the File Manager (FM) general commands function properly. The FM_NoOp and FM_Reset commands will be tested as well as invalid commands and an application reset to see if the FM application behaves appropriately.	FM1000, FM1001, FM1002, FM1003, FM1004, FM1009, FM1009.1, FM1009.2, FM1009.3, FM1009.4, FM4000, FM4001, FM5000
fm_filerename_basic	The purpose of this test is to verify that the File Manager (FM) File Rename Command functions properly. The FM_FileRename, and FM_FileInfo commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DirCreate command is also used to facilitate the testing.	FM1003, FM1004, FM2005, FM2005.1, FM2011, FM3000, FM4000, FM5000
fm_filerename_stress	The purpose of this test is to stress the File Manager (FM) File Rename Command function. The FM_FileRename command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_FileInfo and FM_DirCreate commands are also used to facilitate the testing but are not stressed in this scenario.	FM1002, FM1003, FM1004, FM1005, FM1008, FM2005, FM2005.1, FM2011, FM3000, FM4000, FM5000
fm_filemove_basic	The purpose of this test is to verify that the File Manager (FM) File Move Commands function properly. The FM_FileMove, and FM_FileInfo commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DirCreate command is used to facilitate the testing.	FM1003, FM1004, FM1005, FM2004, FM2004.1, FM2011, FM3000, FM4000, FM5000

Procedure	Description	Requirements Tested
fm_filemove_stress	The purpose of this test is to stress the File Manager (FM) File Move Command function. The FM_FileMove command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_FileInfo and FM_DirCreate commands are also used to facilitate the testing but are not stressed in this scenario.	FM1002, FM1003, FM1004, FM1005, FM1008, FM2004, FM3000, FM4000, FM5000
fm_fileinfo_basic	The purpose of this test is to verify that the File Manager (FM) File Info and File Close Commands functions properly. The FM_FileInfo and FM_Close commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DirCreate command is also used to facilitate the testing.	FM1003, FM1004, FM2007, FM2007.1, FM2007.1.1, FM2008, FM2008.1, FM2011, FM2013, FM3000, FM4000, FM5000
fm_fileinfo_stress	The purpose of this test is to stress the File Manager (FM) File Info Command function. The FM_FileInfo command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_DirCreate command is also used to facilitate the testing but is not stressed in this scenario.	FM1002, FM1003, FM1004, FM1005, FM1008, FM2011, FM3000, FM4000, FM5000
fm_filedecom_basic	The purpose of this test is to verify that the File Manager (FM) File Decompress Command functions properly. The FM_FileDecompress and FM_FileInfo commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DeleteAll, FM_DirCreate, FM_DirDelete, and FM_DirListTlm commands are used to facilitate the testing.	FM1003, FM1004, FM2009, FM2009.1, FM2011, FM4000, FM5000
fm_filedecom_stress	The purpose of this test is to stress the File Manager (FM) File Decompress Command function. The FM_FileDecompress command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_FileInfo and FM_DirCreate commands are also used to facilitate the testing but are not stressed in this scenario.	FM1002, FM1003, FM1004, FM1005, FM1008, FM2009, FM2009.1, FM2011, FM3000, FM4000, FM5000
fm_filecopy_basic	The purpose of this test is to verify that the File Manager (FM) File Copy Commands function properly. The FM_FileCopy and FM_FileInfo commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DirCreate command is also used to facilitate the testing.	FM1003, FM1004, FM1005, FM2002, FM2002.1, FM2011, FM3000, FM4000, FM5000

Procedure	Description	Requirements Tested
fm_filecopy_stress	The purpose of this test is to stress the File Manager (FM) File Copy Command function. The FM_FileCopy command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_FileInfo and FM_DirCreate commands are also used to facilitate the testing but are not stressed in this scenario.	FM1002, FM1003, FM1004, FM1005, FM1008, FM2002, FM2002.1, FM2008, FM3000, FM4000, FM5000
fm_filecat_basic	The purpose of this test is to verify that the File Manager (FM) File Concatenate Command functions properly. The FM_FileCat and FM_FileInfo commands will be tested to see if the FM application handles these as desired by the requirements. The FM_DeleteAll, FM_DirCreate, FM_DirDelete, and FM_DirListTlm commands are used to facilitate the testing.	FM1003, FM1004, FM2010, FM2010.1, FM2011, FM4000, FM5000
fm_filecat_stress	The purpose of this test is to stress the File Manager (FM) File Concatenate Command function. The FM_FileConcatenate command will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_FileInfo and FM_DirCreate commands are also used to facilitate the testing but are not stressed in this scenario.	FM1002, FM1003, FM1004, FM1005, FM1008, FM2010, FM2011, FM3000, FM4000, FM5000
fm_dircmds_basic	The purpose of this test is to verify that the File Manager (FM) Directory Commands function properly. The FM_DirCreate, FM_DirDelete, FM_DirListFile, and FM_DirListTlm commands will be tested to see if the FM application handles these as desired by the requirements. The FM_delete command is also used to facilitate the testing.	FM1003, FM1004, FM1006, FM2008, FM3000, FM3001, F3001.1, FM3002, FM3002.1, FM3002.2, FM3002.3, FM3003, FM3003.1, FM4000, FM5000
fm_dircmds_stress	The purpose of this test is to stress the File Manager (FM) Directory Command functions. The FM_DirCreate, FM_DirDelete, FM_DirListFile, and FM_DirListTlm commands will be tested to see if the FM application handles error cases for these, both expected and unexpected.	FM1002, FM1003, FM1004, FM1005, FM1006, FM1008, FM3000, FM3001, FM3001.1, FM3002, FM3002.1, FM3002.2, FM3003, FM4000, FM5000
fm_dirrename	The purpose of this test is to verify that the File Manager (FM) application does not cause any erroneous things to happen when a directory is renamed.	No requirements are tested by this procedure.

Procedure	Description	Requirements Tested
fm_filedelete_basic	The purpose of this test is to verify the File Manager (FM) File Delete Command functions. The FM_Delete and FM_DeleteAll commands will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_DirCreate command is also used to facilitate the testing but is not tested in this scenario.	FM1002; FM1003; FM1004; FM1005; FM1006; FM2007; FM2007.2; FM2007.2.1; FM2008; FM2008.2; FM2011; FM3000; FM4000; FM5000
fm_filedelete_stress	The purpose of this test is to stress the File Manager (FM) File Delete Command functions. The FM_Delete and FM_DeleteAll commands will be tested to see if the FM application handles error cases for these, both expected and unexpected. The FM_DirCreate command is also used to facilitate the testing but is not stressed in this scenario.	FM1002, FM1003, FM1004, FM1005, FM1006, FM1008, FM2007, FM2007.2, FM2007.2.1, FM2008, FM2008.2, FM2011, FM3000, FM4000, FM5000
fm_openfiles	The purpose of this test is to verify the FM_ListOpenFiles command.	FM1002, FM1003, FM1004, FM2007, FM2007.1, FM2007.1.1, FM2008, FM2008.1, FM2012, FM3000, FM4000, FM4001, FM5000
fm_specialchars1 fm_specialchars2 fm_specialchars3 fm_specialchars4 fm_specialchars5	The purpose of these tests is to stress the File Manager (FM) Command functions that have a directory or filename as an argument. The arguments are setup to contain special characters and the procedure documents which characters are valid and which ones are not. The five tests are as follows: 1. The special characters by themselves. 2. The special characters as the first character. 3. The special characters in the middle 4. The special characters as the last character 5. The special characters in the file extension for filename arguments only.	FM1003, FM3002.1, FM4000, FM5000

5.3 FAILED REQUIREMENTS

No requirements failed during the final FM build 2.5.4.0 testing.

5.4 DCRS

No new DCRs were generated during FM 2.5.4.0 testing.

5.4.1 DCRs Verified

The following DCRs were verified during testing:

DCR	Description	Test Method	Test Approach
GSFCCFS-1030	Remove magic numbers in FM	Inspection	The hard-coded numbers are no longer contained in the fsw delivered.
GSFCCFS-1031	FM should check all function arguments for NULL	Inspection	The changes described in this issue were found in the fsw delivered.
GSFCCFS-1134	Update FM to handle Decompress removed from cFE	Test	The fm_filedecom_basic test verifies this functionality.
GSFCCFS-1139	Update FM to handle CFE_TIME_FS2Seconds Deprecation	Inspection	The specified items that were marked for deprecation were not found in the fsw delivered.
GSFCCFS-1140	FM relies on deprecated items	Inspection	The specified items that were marked for deprecation were not found in the fsw delivered.
GSFCCFS-1142	FM uses OS_FS* Error Codes (soon deprecated)	Inspection	The specified items that were marked for deprecation were not found in the fsw delivered.
GSFCCFS-1235	FM may have alignment problems on some platforms	Inspection	The test of the VxWorks platform did not contain any alignment issues.
GSFCCFS-1244	FM has command and telemetry alignment issues	Test	The Copy and Move commands were executed successfully and the Housekeeping packet was received properly by ASIST.
GSFCCFS-1245	FM Version number is not correct for Release Candidate 2.5.4	Test	The version was properly reported when the FM application was started during testing.

5.5 NOTES

None.

5.6 FOLLOW-ON

None

an be found on the ETD GIT server, https://aetd-git.gsfc.nasa.gov/gsfc-cfs/cfs_fm in the test-and-ground/results folder.

APPENDIX B - COMMAND, TELEMETRY, AND EVENTS VERIFICATION MATRIX

Command	Test Procedure(s)	Notes/Comments
FM_NoOp	gencmds	
FM_ResetCtrs	gencmds	
FM_FileCopy	filecopy_basic, filecopy_stress, specialchars1 - 5	
FM_FileMove	filemove_basic, filemove_stress, specialchars1 - 5	
FM_FileRename	Dirrename, filerename_basic, filerename_stress, specialchars1 - 5	
FM_Delete	dircmds_basic, filecopy_stress, filedelete_basic,filedelete_stress, fileinfo_basic, openfiles, specialchars1 - 5	
FM_DeleteAll	dircmds_stress, filedecom_stress, filedelete_basic, filedelete_stress, fileinfo_basic, openfiles, specialchars1 - 4	
FM_Decompress	filedecom_basic, filedecom_stress, specialchars1 - 5	
FM_FileCat	filecat_basic, filecat_stress, specialchars1 - 5	
FM_FileInfo	dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_basic, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress, filerename_basic, filerename_stress, specialchars1 - 5	
FM_ListOpenFiles	dirrename, openfiles	
FM_DirCreate	dircmds_basic, dircmds_stress, dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_basic, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress, filerename_basic, filerename_stress, openfiles, specialchars1 – 5	
FM_DirDelete	dircmds_basic, dircmds_stress, specialchars1 - 4	

Command	Test Procedure(s)	Notes/Comments
FM_DirListFile	dircmds_basic, dircmds_stress; filedelete_basic, filedelete_stress, specialchars1 - 5	
FM_DirListTlm	dircmds_basic, dircmds_stress, openfiles, specialchars1 - 5	
FM_GetFreeSpace	Gencmds, openfiles	
FM_SetTblState	gencmds	

Telemetry	Test Procedure(s)	Notes/Comments
FM_CMDPC	dircmds_basic, dircmds_stress, dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_basic, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress, filerename_basic, filerename_stress, gencmds, openfiles, specialchars1 – 5	
FM_CMDEC	dircmds_basic, dircmds_stress, dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_basic, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress, filerename_basic, filerename_stress, gencmds, openfiles, specialchars1 – 5	
FM_NumOpen	filecat_basic, filecopy_basic, filedecom_basic, fileinfo_basic, filemove_basic, filerename_basic, openfiles	
FM_ChildCMDPC	dircmds_basic, dircmds_stress, filecopy_stress, filedecom_stress, filedelete_stress, filemove_stress, specialchars1 - 5	
FM_ChildCMDEC	dircmds_basic, dircmds_stress, filecat_stress, filecopy_stress, filedecom_stress, fileinfo_basic, filemove_basic, filemove_stress, filerename_stress	
FM_ChildWarnCtr		
FM_ChildQueCnt		
FM_ChildCurrCC		
FM_ChildPrevCC		
FM_TotalOpenFiles	openfiles	
FM_OpenFileList[].FileName	openfiles	
FM_OpenFileList[].AppName		
FM_DirName	dircmds_basic, dircmds_stress, openfiles, specialchars1 – 5	
FM_TotalFiles	dircmds_basic, dircmds_stress, openfiles, specialchars1 - 5	
FM_PktFiles	dircmds_basic, dircmds_stress, openfiles, specialchars1 - 5	
FM_DirOffset	dircmds_basic, dircmds_stress, openfiles, specialchars1 - 5	
FM_DirList[].Name	dircmds_basic, dircmds_stress, openfiles, specialchars1 - 5	
FM_DirList[].FileSize	dircmds_basic, dircmds_stress, openfiles, specialchars1 - 5	
FM_DirList[].LastModTime	dircmds_basic, dircmds_stress, openfiles, specialchars1 - 5	

Telemetry	Test Procedure(s)	Notes/Comments
FM_DirList[].FilePerms	dircmds_basic, dircmds_stress; filedelete_stress, openfiles, specialchars1 - 5	
FM_FileStatus	dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress filerename_basic, filerename_stress, specialchars1 - 5	
FM_ComputeCRC	dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress filerename_basic, filerename_stress, specialchars1 - 5	
FM_CRC	dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress filerename_basic, filerename_stress, specialchars1 - 5	
FM_InfoFileSize	dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress filerename_basic, filerename_stress, specialchars1 - 5	
FM_ModTime	dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress filerename_basic, filerename_stress, specialchars1 - 5	
FM_Perms		
FM_InfoFileName[]	dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress filerename_basic, filerename_stress, specialchars1 - 5	
FM_FreeSpacePkt[].Upper32	gencmds, openfiles	
FM_FreeSpacePkt[].Lower32	gencmds, openfiles	

Telemetry	Test Procedure(s)	Notes/Comments
FM_FreeSpacePkt[].Name	gencmds, openfiles	

File and Table Telemetry	Test Procedure(s)	Notes/Comments
FM_DirNameInFile	dircmds_basic, dircmds_stress, filedelete_stress, specialchars1 – 5	
FM_TotalFilesInDir	dircmds_basic, dircmds_stress, filedelete_stress, specialchars1 – 5	
FM_NumFilesWritten	dircmds_basic, dircmds_stress, filedelete_stress, specialchars1 – 5	
FM_FileListEntry[].Name	dircmds_basic, dircmds_stress; filedelete_stress, openfiles, specialchars1 - 5	
FM_FileListEntry[].FileSize	dircmds_basic, dircmds_stress; filedelete_stress, openfiles, specialchars1 - 5	
FM_FileListEntry[].LastModTime	dircmds_basic, dircmds_stress; filedelete_stress, openfiles, specialchars1 - 5	
FM_FileListEntry[].FilePerms	dircmds_basic, dircmds_stress; filedelete_stress, openfiles, specialchars1 - 5	
FM_FreeSpaceTBL[].State	badtblloadfile, gencmds, openfiles, tableloadfile	
FM_FreeSpaceTBL[].Name	badtblloadfile, gencmds, openfiles, tableloadfile	

Id	Event Message	Test Procedure(s)	Notes/Comments
1	FM_STARTUP_EID	dircmds_basic, dircmds_stress, dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_basic,filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress, filerename_basic, filerename_stress, gencmds, openfiles	
2	FM_STARTUP_EVENTS_ERR_EID		
3	FM_STARTUP_CREAT_PIPE_ERR_EID		
4	FM_STARTUP_SUBSCRIB_HK_ERR_EID		
5	FM_STARTUP_SUBSCRIB_GCMD_ERR_EID		

Id	Event Message	Test Procedure(s)	Notes/Comments
6	FM_STARTUP_TABLE_INIT_ERR_EID		
7	FM_SB_RECEIVE_ERR_EID		
8	FM_EXIT_ERR_EID	gencmds	
9	FM_MID_ERR_EID		
10	FM_CC_ERR_EID	gencmds	
11	FM_HK_REQ_ERR_EID		
12	FM_NOOP_CMD_EID	gencmds	
13	FM_NOOP_PKT_ERR_EID	gencmds	
14	FM_RESET_CMD_EID	gencmds	
15	FM_RESET_PKT_ERR_EID	gencmds	
16	FM_COPY_CMD_EID	dircmds_stress, filecopy_basic, filecopy_stress	
17	FM_COPY_PKT_ERR_EID	filecopy_stress	
18	FM_COPY_OVR_ERR_EID		
19	FM_COPY_OS_ERR_EID	dircmds_stress, filecopy_stress	
20	FM_MOVE_CMD_EID	filemove_basic, filemove_stress	
21	FM_MOVE_PKT_ERR_EID	filemove_stress	
22	FM_MOVE_OVR_ERR_EID		
23	FM_MOVE_OS_ERR_EID	filemove_stress	
24	FM_RENAME_CMD_EID	filerename_basic, filerename_stress	
25	FM_RENAME_PKT_ERR_EID	filerename_stress	
26	FM_RENAME_OS_ERR_EID	filerename_stress	
27	FM_DELETE_CMD_EID	dircmds_basic, filecopy_stress, filedelete_stress, filemove_stress	
28	FM_DELETE_PKT_ERR_EID	filedelete_stress	
29	FM_DELETE_OS_ERR_EID		
30	FM_DELETE_ALL_CMD_EID	dircmds_stress, filedecom_stress, filedelete_stress, fileinfo_basic, openfiles	
31	FM_DELETE_ALL_FILES_ND_WARNING_EID	filedelete_stress, fileinfo_basic, openfiles	
32	FM_DELETE_ALL_SKIP_WARNING_EID	filedelete_stress	
33	FM_DELETE_ALL_PKT_ERR_EID	filedelete_stress	
34	FM_DELETE_ALL_OS_ERR_EID		
35	FM_DECOM_CMD_EID	filedecom_basic, filedecom_stress	
36	FM_DECOM_PKT_ERR_EID	filedecom_stress	
37	FM_DECOM_CFE_ERR_EID	filedecom_stress	
38	FM_CONCAT_CMD_EID	filecat_basic, filecat_stress	
39	FM_CONCAT_PKT_ERR_EID	filecat_stress	
40	FM_CONCAT_OSCPY_ERR_EID	filecat_stress	
41	FM_CONCAT_OPEN_SRC2_ERR_EID		
42	FM_CONCAT_OPEN_TGT_ERR_EID		
43	FM_CONCAT_OSRD_ERR_EID		
44	FM_CONCAT_OSWR_ERR_EID		

Id	Event Message	Test Procedure(s)	Notes/Comments
45	FM_GET_FILE_INFO_CMD_EID	dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress, filerename_basic, filerename_stress	
46	FM_GET_FILE_INFO_STATE_WARNING_EID		
47	FM_GET_FILE_INFO_TYPE_WARNING_EID	fileinfo_stress	
48	FM_GET_FILE_INFO_OPEN_ERR_EID		
49	FM_GET_FILE_INFO_READ_WARNING_EID		
50	FM_GET_FILE_INFO_PKT_ERR_EID	fileinfo_stress	
51	FM_GET_FILE_INFO_SRC_ERR_EID	fileinfo_stress	
52	FM_GET_OPEN_FILES_CMD_EID	dirrename, openfiles	
53	FM_GET_OPEN_FILES_PKT_ERR_EID	openfiles	
54	FM_CREATE_DIR_CMD_EID	dircmds_basic, dircmds_stress, dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress, filerename_basic, filerename_stress, openfiles	
55	FM_CREATE_DIR_PKT_ERR_EID	dircmds_stress	
56	FM_CREATE_DIR_OS_ERR_EID	dircmds_stress	
57	FM_DELETE_DIR_CMD_EID	dircmds_basic, dircmds_stress	
58	FM_DELETE_DIR_PKT_ERR_EID	dircmds_stress	
59	FM_DELETE_DIR_EMPTY_ERR_EID	dircmds_basic, dircmds_stress	
60	FM_DELETE_OPENDIR_OS_ERR_EID		
61	FM_DELETE_RMDIR_OS_ERR_EID		
62	FM_GET_DIR_FILE_CMD_EID	dircmds_basic, dircmds_stress, filedelete_stress	
63	FM_GET_DIR_FILE_PKT_ERR_EID	dircmds_stress	
64	FM_GET_DIR_FILE_WARNING_EID	dircmds_stress, filedelete_stress	
65	FM_GET_DIR_FILE_OSOPENDIR_ERR_EID		
66	FM_GET_DIR_FILE_WRBLANK_ERR_EID		
67	FM_GET_DIR_FILE_WRHDR_ERR_EID		
68	FM_GET_DIR_FILE_OSCREAT_ERR_EID	dircmds_stress	
69	FM_GET_DIR_FILE_WRENTRY_ERR_EID	dircmds_stress	
70	FM_GET_DIR_FILE_UPSTATS_ERR_EID		
71	FM_GET_DIR_PKT_CMD_EID	dircmds_basic, dircmds_stress, openfiles	
72	FM_GET_DIR_PKT_WARNING_EID	dircmds_stress	
73	FM_GET_DIR_PKT_PKT_ERR_EID	dircmds_stress	
74	FM_GET_DIR_PKT_OS_ERR_EID		
75	FM_GET_FREE_SPACE_CMD_EID	gencmds, openfiles	
76	FM_GET_FREE_SPACE_PKT_ERR_EID	gencmds	
77	FM_GET_FREE_SPACE_TBL_ERR_EID		
78	FM_SET_TABLE_STATE_CMD_EID	gencmds	

Id	Event Message	Test Procedure(s)	Notes/Comments
79	FM_SET_TABLE_STATE_PKT_ERR_EID	gencmds	
80	FM_SET_TABLE_STATE_TBL_ERR_EID	gencmds	
81	FM_SET_TABLE_STATE_ARG_IDX_ERR_EID	gencmds	
82	FM_SET_TABLE_STATE_ARG_STATE_ERR_EID	gencmds	
83	FM_SET_TABLE_STATE_UNUSED_ERR_EID	gencmds	
84	FM_TABLE_VERIFY_EMPTY_ERR_EID	gencmds	
85	FM_TABLE_VERIFY_TOOLONG_ERR_EID		
86	FM_TABLE_VERIFY_INVALID_ERR_EID		
87	FM_TABLE_VERIFY_BAD_STATE_ERR_EID		
88	FM_CHILD_INIT_EID	dircmds_basic, dircmds_stress, dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress, filerename_basic, filerename_stress, gencmds, openfiles	
89	FM_CHILD_INIT_SEM_ERR_EID		
90	FM_CHILD_INIT_QSEM_ERR_EID		
91	FM_CHILD_INIT_CREATE_ERR_EID		
92	FM_CHILD_TERM_EMPTYQ_ERR_EID		
93	FM_CHILD_TERM_QIDX_ERR_EID		
94	FM_CHILD_TERM_SEM_ERR_EID		
95	FM_CHILD_EXE_ERR_EID		
96	FM_TABLE_VERIFY_EID	dircmds_basic, dircmds_stress, dirrename, filecat_basic, filecat_stress, filecopy_basic, filecopy_stress, filedecom_basic, filedecom_stress, filedelete_stress, fileinfo_basic, fileinfo_stress, filemove_basic, filemove_stress, filerename_basic, filerename_stress, gencmds, openfiles	
97	FM_SET_PERM_ERR_EID		
98	FM_SET_PERM_CMD_EID		
99	FM_SET_PERM_OS_ERR_EID	fileinfo_basic	
100	FM_COPY_SRC_INVALID_ERR_EID	filecopy_stress	
101	FM_COPY_SRC_DNE_ERR_EID	filecopy_stress	
102	FM_COPY_SRC_ISDIR_ERR_EID	filecopy_stress	
105	FM_COPY_SRC_UNKNOWN_ERR_EID		
106	FM_COPY_TGT_INVALID_ERR_EID	filecopy_stress	
107	FM_COPY_TGT_EXIST_ERR_EID	filecopy_stress	
108	FM_COPY_TGT_ISDIR_ERR_EID	filecopy_stress	
111	FM_COPY_TGT_UNKNOWN_ERR_EID		
112	FM_COPY_CHILD_DISABLED_ERR_EID		
113	FM_COPY_CHILD_FULL_ERR_EID		

Id	Event Message	Test Procedure(s)	Notes/Comments
114	FM_COPY_CHILD_BROKEN_ERR_EID		
115	FM_MOVE_SRC_INVALID_ERR_EID	filemove_stress	
116	FM_MOVE_SRC_DNE_ERR_EID	filemove_stress	
117	FM_MOVE_SRC_ISDIR_ERR_EID	filemove_stress	
121	FM_MOVE_TGT_INVALID_ERR_EID	filemove_stress	
122	FM_MOVE_TGT_DNE_ERR_EID	filemove_stress	
123	FM_MOVE_TGT_ISDIR_ERR_EID	filemove_stress	
127	FM_MOVE_CHILD_DISABLED_ERR_EID		
128	FM_MOVE_CHILD_FULL_ERR_EID		
129	FM_MOVE_CHILD_BROKEN_ERR_EID		
130	FM_RENAME_SRC_INVALID_ERR_EID	filerename_stress	
131	FM_RENAME_SRC_DNE_ERR_EID	filerename_stress	
132	FM_RENAME_SRC_ISDIR_ERR_EID	filerename_stress	
136	FM_RENAME_TGT_INVALID_ERR_EID	filerename_stress	
137	FM_RENAME_TGT_DNE_ERR_EID	filerename_stress	
138	FM_RENAME_TGT_ISDIR_ERR_EID	filerename_stress	
142	FM_RENAME_CHILD_DISABLED_ERR_EID		
143	FM_RENAME_CHILD_FULL_ERR_EID		
144	FM_RENAME_CHILD_BROKEN_ERR_EID		
145	FM_DELETE_SRC_INVALID_ERR_EID	filedelete_stress	
146	FM_DELETE_SRC_DNE_ERR_EID	filedelete_stress	
147	FM_DELETE_SRC_ISDIR_ERR_EID	filedelete_stress	
148	FM_DELETE_SRC_OPEN_ERR_EID	fileinfo_basic, openfiles	
151	FM_DELETE_CHILD_DISABLED_ERR_EID		
152	FM_DELETE_CHILD_FULL_ERR_EID		
153	FM_DELETE_CHILD_BROKEN_ERR_EID		
154	FM_DELETE_ALL_SRC_INVALID_ERR_EID	filedelete_stress	
155	FM_DELETE_ALL_SRC_DNE_ERR_EID	filedelete_stress	
156	FM_DELETE_ALL_SRC_FILE_ERR_EID		
160	FM_DELETE_ALL_CHILD_DISABLED_ERR_EID		
161	FM_DELETE_ALL_CHILD_FULL_ERR_EID		
162	FM_DELETE_ALL_CHILD_BROKEN_ERR_EID		
163	FM_DECOM_SRC_INVALID_ERR_EID	filedecom_stress	
164	FM_DECOM_SRC_DNE_ERR_EID	filedecom_stress	
165	FM_DECOM_SRC_ISDIR_ERR_EID	filedecom_stress	
166	FM_DECOM_SRC_OPEN_ERR_EID	filedecom_basic	
169	FM_DECOM_TGT_INVALID_ERR_EID	filedecom_stress	
170	FM_DECOM_TGT_EXIST_ERR_EID	filedecom_stress	
171	FM_DECOM_TGT_ISDIR_ERR_EID	filedecom_stress	
175	FM_DECOM_CHILD_DISABLED_ERR_EID		
176	FM_DECOM_CHILD_FULL_ERR_EID		
177	FM_DECOM_CHILD_BROKEN_ERR_EID		
178	FM_CONCAT_SRC1_INVALID_ERR_EID	filecat_stress	
179	FM_CONCAT_SRC1_DNE_ERR_EID	filecat_stress	
180	FM_CONCAT_SRC1_ISDIR_ERR_EID	filecat_stress	
181	FM_CONCAT_SRC1_OPEN_ERR_EID	filecat_basic	
184	FM_CONCAT_SRC2_INVALID_ERR_EID	filecat_stress	
185	FM_CONCAT_SRC2_DNE_ERR_EID	filecat_stress	
186	FM_CONCAT_SRC2_ISDIR_ERR_EID	filecat_stress	
187	FM_CONCAT_SRC2_OPEN_ERR_EID	filecat_basic	
190	FM_CONCAT_TGT_INVALID_ERR_EID	filecat_stress	

Id	Event Message	Test Procedure(s)	Notes/Comments
191	FM_CONCAT_TGT_EXIST_ERR_EID	filecat_basic, filecat_stress	
192	FM_CONCAT_TGT_ISDIR_ERR_EID	filecat_stress	
196	FM_CONCAT_CHILD_DISABLED_ERR_EID		
197	FM_CONCAT_CHILD_FULL_ERR_EID		
198	FM_CONCAT_CHILD_BROKEN_ERR_EID		
199	FM_FILE_INFO_CHILD_DISABLED_ERR_EID		
200	FM_FILE_INFO_CHILD_FULL_ERR_EID		
201	FM_FILE_INFO_CHILD_BROKEN_ERR_EID		
202	FM_CREATE_DIR_SRC_INVALID_ERR_EID	dircmds_stress	
203	FM_CREATE_DIR_SRC_DNE_ERR_EID	dircmds_stress	
204	FM_CREATE_DIR_SRC_ISDIR_ERR_EID	dircmds_stress	
208	FM_CREATE_DIR_CHILD_DISABLED_ERR_EID		
209	FM_CREATE_DIR_CHILD_FULL_ERR_EID		
210	FM_CREATE_DIR_CHILD_BROKEN_ERR_EID		
211	FM_DELETE_DIR_SRC_INVALID_ERR_EID	dircmds_stress	
212	FM_DELETE_DIR_SRC_DNE_ERR_EID	dircmds_stress	
213	FM_DELETE_DIR_SRC_ISDIR_ERR_EID	dircmds_stress	
217	FM_DELETE_DIR_CHILD_DISABLED_ERR_EID		
218	FM_DELETE_DIR_CHILD_FULL_ERR_EID		
219	FM_DELETE_DIR_CHILD_BROKEN_ERR_EID		
220	FM_GET_DIR_FILE_SRC_INVALID_ERR_EID	dircmds_stress	
221	FM_GET_DIR_FILE_SRC_DNE_ERR_EID	dircmds_basic, dircmds_stress	
222	FM_GET_DIR_FILE_SRC_ISDIR_ERR_EID	dircmds_stress	
226	FM_GET_DIR_FILE_TGT_INVALID_ERR_EID		
227	FM_GET_DIR_FILE_TGT_DNE_ERR_EID		
228	FM_GET_DIR_FILE_TGT_ISDIR_ERR_EID	dircmds_stress	
232	FM_GET_DIR_FILE_CHILD_DISABLED_ERR_EID		
233	FM_GET_DIR_FILE_CHILD_FULL_ERR_EID		
234	FM_GET_DIR_FILE_CHILD_BROKEN_ERR_EID		
235	FM_GET_DIR_PKT_SRC_INVALID_ERR_EID	dircmds_stress	
236	FM_GET_DIR_PKT_SRC_DNE_ERR_EID	dircmds_basic, dircmds_stress	
237	FM_GET_DIR_PKT_SRC_ISDIR_ERR_EID	dircmds_stress	
241	FM_GET_DIR_PKT_CHILD_DISABLED_ERR_EID		
242	FM_GET_DIR_PKT_CHILD_FULL_ERR_EID		
243	FM_GET_DIR_PKT_CHILD_BROKEN_ERR_EID		