|  |  |
| --- | --- |
| DocCoverBackground | CORE FLIGHT SYSTEM  FILE MANAGER APPLICATION  BUILD 2.5.2.0  FLIGHT SOFTWARE BUILD VERIFICATON  TEST REPORT  Flight Software Branch – Code 582  Version 1.0 |

Signatures

Submitted by:



Approved by:



Plan Update History

| Version | Date | Description | Affected Pages |
| --- | --- | --- | --- |
| 1.0 | 01/14/2015 | Initial release | All |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1 Introduction 1

1.1 Document Purpose 1

1.2 Applicable Documents 1

1.3 Document Organization 1

1.4 Definitions 2

2 OVERVIEW 3

2.1 Flight Data System Context 3

2.2 Test History 4

2.3 Testing Overview 4

2.4 Version Information 8

3 Build Verification Test Preparation 9

3.1 Scenerio Development 9

3.2 Procedure Development and Execution 9

3.3 Test Products 9

4 Build Verification Test Execution 10

4.1 Testbed Overview 10

4.2 Requirements Verification Matrix 11

4.3 Requirements Partially Tested 11

4.4 Requirements/Functionality Deferred 11

4.5 Requirements/Functionality Deferred For Mission Testing 11

5 Build Verficiaton Test Results 12

5.1 Overall Assessment 12

5.2 Procedure Description 12

5.3 Analysis Requirements Verification 15

5.4 Failed Requirements 16

5.5 DCRs 16

5.5.1 DCRs Verified 17

5.5.2 Outstanding DCRs 18

5.6 Notes 18

5.7 Follow-on 18

Appendix A - RTTM 19

Appendix B - Command, Telemetry, and Events Verification Matrix 20

# Introduction

## Document Purpose

This Test Report describes the test results from the core Flight System (cFS) File Manager (FM) Flight Software (FSW) Test Team builds 2.5.0.0, 2.5.1.0, and 2.5.2.0 verification testing. The FM 2.5.2.0 build contains the changes associated with builds 2.5.0.0, 2.5.1.0, and 2.5.2.0. The 2.5.1.0 and 2.5.2.0 are revisions that include bug fixes found during FM builds 2.5.0.0 and 2.5.1.0 Build Verification Testing (BVT). Build 2.5.2.0 is the final build of the cFS FM application following BVT of builds 2.5.0.0 and 2.5.1.0 of the cFS FM application.

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.

## 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
* 582-2004-001 FSB FSW Test Description Template
* 582-2004-002 FSB FSW Test Scenario Template
* 582-2004-003 FSB FSW Test Procedure *Template*
* 582-2004-004 FSB FSW Test Execution Summary Template
* 582-2004-005 FSB Test Product Peer Review Form
* 582-2000-002 FSB FSW Unit Test Standard

## 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

## Definitions

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

* Demonstration: 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, unles**s** (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

In each software test section in Section 5 there is a table of DCR’s. The state definitions are as follows:

* Opened: The DCR is currently being addressed
* Assigned: The DCR was accepted and the modification is being addressed
* InTest: The DCR was corrected and is currently in test
* Validated: The DCR was corrected and tested and have been validated, needs to have a CCB to close the DCR
* Closed: The DCR is closed and have been resolved and tested to satisfaction
* Closed with Defect: The DCR is closed and the defect is most likely assigned a differed DCR number associated with another subsystem.

# OVERVIEW

## Flight Data System Context

Figure 2-1 illustrates the cFS system context. The cFE interfaces to five external systems: an [Operating System](#Operating_System) (OS), a [Hardware Platform](#Hardware_Platform) (HP), an [Operational Interface](file:///C:\Users\wmoleski\Documents\Projects\cFE\CFS\File%20Manager\Testing\2.4.1.0\Operational_Interface) (OI), [Applications](#Application) (APP), and other cFE-based systems.

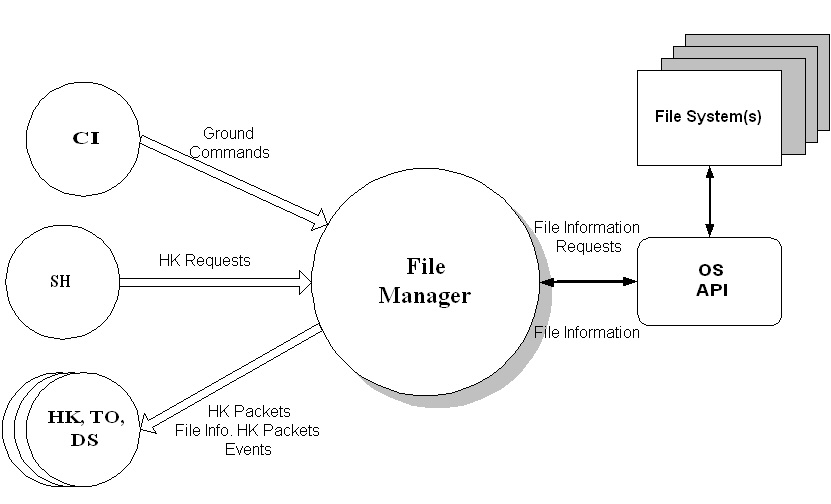


**Figure 2-1 cFS System Context**

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**

## 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

## Testing Overview

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

* 1 test application: tst\_fm
* 23 main test procedures (listed below)
* 7 test procedures that are called by the main procedures (listed below)
* Header files used in procedures: tst\_fm\_events.h, fm\_events.h, fm\_platform\_cfg.h, fm\_defs.h, ut\_statusdefs.h, cfe\_evs\_events.h, cfe\_es\_events.h
* RDLs used: template\_cmd\_FM\_CMD.rdl, template\_file\_FM\_DIRLISTFILE.rdl, template\_table\_FM\_FREESPACE\_TBL.rdl, template\_tlm\_FM\_DIRLIST\_TLM.rdl, template\_tlm\_FM\_FILESTAT\_TLM.rdl, template\_tlm\_FM\_FREESPACE\_TLM.rdl, template\_tlm\_FM\_HK\_TLM.rdl, template\_tlm\_FM\_OPENFILELIST\_TLM.rdl, template\_cmd\_TST\_FM\_CMD.rdl, template\_tlm\_TST\_FM\_HK\_TLM.rdl
* Header files used in RDLs: osconfig.h, fm\_platform\_cfg.h, cfe\_file\_header.rdl, ccsds\_header.rdl

The tst\_fm test application is used to send schedule requests for the output of FM’s housekeeping data. TST\_FM has 4 ground commands that are used by the FM test procedures:

* $sc\_$cpu\_TST\_FM\_NoOp
  + Input: none
  + Output: Command Accepted counter increased, info event message issued
* $sc\_$cpu\_TST\_FM\_ResetCtrs
  + Input: none
  + Output: Command Accepted and Rejected counters set to zero, debug event message issued
* $sc\_$cpu\_TST\_FM\_Open
  + Input: File name of the file that you wish to open
  + Output: Command Accepted counter increased, debug event message issued
* $sc\_$cpu\_TST\_FM\_Close
  + Input: File name of the file that you wish to close
  + Output: Command Accepted counter increased, debug event message issued
* $sc\_$cpu\_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 23 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. |
| 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. |
| 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\_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 cFS Deployment Guide contains the instruction for how to set up both the cFS Flight and Ground test environment. The testers use a cFS Test Account for each build test. This account runs ASIST and is setup to contain all the files needed to test the application. These files are extracted from MKS, the source repository tool. 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 |

## Version Information

|  |  |
| --- | --- |
| Item | Version |
| FM Requirements | 1.5 |
| FM Application | 2.5.2.0 |
| TST\_FM Application | 2.5.2.0 |
| CFE | 6.5.0.0 |
| ASIST | 20.2 |
| VxWorks | 6.9 |

# Build Verification Test Preparation

## Scenerio Development

No new scenarios were developed for build verification tests 2.5.0.0, 2.5.1.0, or 2.5.2.0. All scenarios are stored on the MKS server, in cFS-Repository FM test-and-ground directory within the test-review-packages subdirectory in the Scenarios folder. It should be noted that as FM requirement evolve these scenarios are not updated to reflect any changes made.

## Procedure Development and Execution

This build test effort was completed by running 18 of the 23 test procedures. The 5 special character tests were not executed on FM 2.5.0.0, 2.5.1.0, or 2.5.2.0. Various procedures were modified as a result of the FM 2.5.0.0, 2.5.1.0, and 2.5.2.0 changes and are checked in to MKS 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>.

## Test Products

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

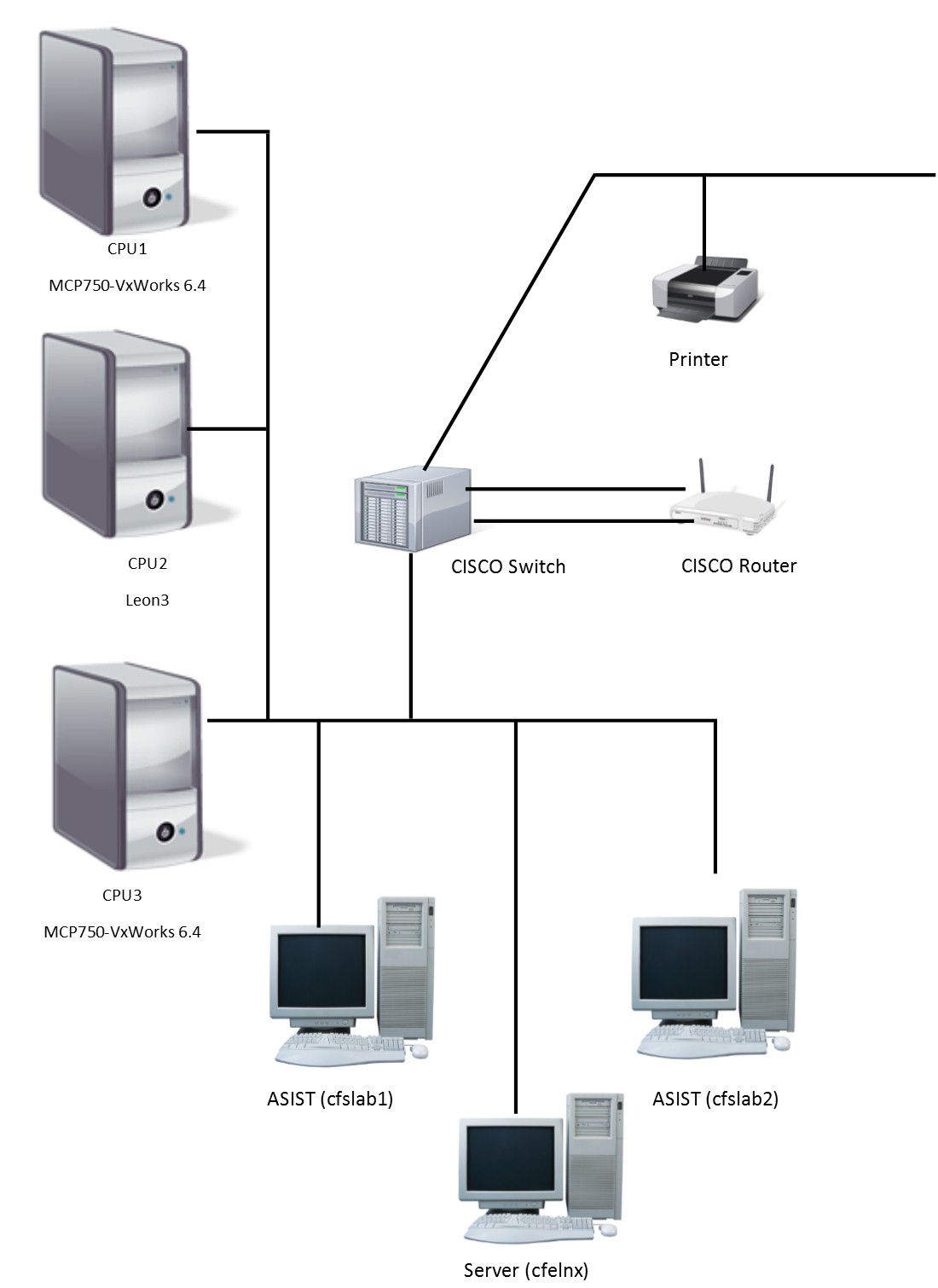
* Logs with the .loge extension list all events sent by the flight software
* Logs with the .logr extension list all requirements that passed validation by demonstration
* Logs with the .logp extension lists all prints that are generated by the test procedure
* Logs with the .logf extension lists everything 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 MKS for each procedure by the tester after build testing is completed. All test products are maintained on MKS in the cFS-Repository FM test-and-ground directory.

# Build Verification Test Execution

## 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 N410. This facility consists of two ASIST workstations running ASIST version 20.2 and three MPC750 CPU boards running VxWorks 6.4. 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**

## Requirements Verification Matrix

|  |  |
| --- | --- |
|  | File Manager (FM) |
| Requirements Tested Passed | 46 |
| Requirements Tested Failed | 0 |
| Requirements Tested Partially | 0 |
| Total Tested | 46 |
| Deferred | 0 |
| Total | 46 |

## Requirements Partially Tested

No requirements were partially tested.

## Requirements/Functionality Deferred

No requirements were deferred for later build testing

## 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.

# Build Verficaton Test Results

## Overall Assessment

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

* 46 requirements passed demonstration
* 0 requirements were validated by analysis
* 0 requirement failed
* 0 requirements were deferred for testing later
* 14 existing DCRs were validated

## 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 |
| 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 |
| 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. |
| 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 |

## Analysis Requirements Verification

No requirements were verified using analysis.

## Failed Requirements

No requirements failed during the final FM build 2.5.2.0 testing.

## DCRs

Three new DCRs were generated during FM 2.5.0.0 testing:

1. DCR 145999
2. DCR 146000
3. DCR 146006

See section 5.5.1, [DCRs Verified](#_DCRs_Verified), for DCR details.

Three new DCRs were generated during FM 2.5.1.0 testing:

1. DCR 146008
2. DCR 146011
3. DCR 146018

See section 5.5.1, [DCRs Verified](#_DCRs_Verified), for DCR details.

No new DCRs were generated during FM 2.5.2.0 testing.

### DCRs Verified

The following DCRs were verified during testing:

| **DCR** | **Description** | **Test Method** | **Test Approach** |
| --- | --- | --- | --- |
| 4014 | FM – Consider removing Child Task Priority and Pipe Depth Checks from verify.h | Inspection | The verify.h file no longer contains a check for Pipe Depth and the priority check was changed to be less restrictive. |
| 4023 | GPM-IVV-1340 – FM requirements do not capture overwrite capability | Inspection | Requirements FM2002.1 and FM2004.1 have been updated to capture the overwrite capability. |
| 4178 | Verify result from OS\_MutSemCreate() before use of FM\_GlobalData.ChildQueCountSem | Inspection | The result from OS\_MutSemCreate() is now verified in the code. |
| 7558 | High CPU seen when using the directory listing to file command GC\_FM\_DIRLISTFILE | Test Procedure | An option was added to the directory list commands as well as measures to prevent FM hogging the CPU. Testing the commands did not result in any CPU Hogging. |
| 145875 | FM – Add Command to Set/Change File Permissions | Test Procedure | The command was added and tested successfully to the extent possible since the underlying OSAL does not support the command used to actually set the file permissions at the time of FM testing. |
| 145907 | FM – Integrate and Implement Babelfish Ticket Fixes | Analysis | No compiler warnings or errors were generated when FM 2.5.2.0 was compiled. |
| 145918 | FM – CFE\_EVS\_SendEvent Format Warnings | Analysis | No compiler warnings or errors were generated when FM 2.5.2.0 was compiled. |
| 145983 | FM – Add requirement to Set/Change File Permissions | Inspection | The new FM2013 requirement exists. |
| 145986 | FM – Update Requirements Based on DCR 7558 Updates | Inspection | The updates stated were found in the FM requirements. |
| 145999 | FM – Issues Found During BVT | Analysis | The issues were fixed and tested successfully. |
| 146000 | Fix Typo in FM Requirement FM2011 | Inspection | The typo was fixed. |
| 146008 | FM: Directory List to File command generates a file that contains incorrect information when Flag is set to FALSE | Test Procedure | The file contents are now set to 0 when the flag is set to FALSE, |
| 146011 | FM: The Set/Change File Permissions command does not increment the Child task error count on OSAL error | Test Procedure | The child error counter now increments on OSAL error. |
| 146018 | New File Permissions Requirements are Ambiguous | Inspection | Requirements FM3002.3 and FM 3003.1 were modified to explicitly state what the fields are to be set to when the flag is set to FALSE. |

### 

### Outstanding DCRs

|  |  |  |
| --- | --- | --- |
| **DCR** | **Description** | **State** |
| 4114 | FM - Add Trick Simulation Support (JSC Request) | Submitted |
| 4136 | The child task command processing in the FM application is hard coded. During the Flight Software Systems Branch Internal CDR of the CFS FM application questions came up as to whether or not child task command processing should be configurable, if all FM commands should be processed by the child, or if the decision should be made in real time (RFA #4). | Submitted |
| 4108 | FM CRC Tool Displays Output in Hexidecimal Only | Submitted |
| 4064 | The FM application currently will only provide file system disk free space information via command. Disk space is often monitored by applications such as LC for appropriate action to be taken if a file system/disk free space falls below a certain amount. For future FM releases consider adding platform configuration option for having free space sent in telemetry (or not). | Submitted |
| 4009 | On GPM, when given a directory argument "/", the FM\_DirListTlm command lists the files in the /ram directory, but without file size and timestamp information (as is done when the command argument is "/ram"). The FM\_DirListFile command should also be checked. | Submitted |

## Notes

None.

## Follow-on

None

1. RTTM

The FM Build 2.5.2.0 RTTM can be found on the MKS server, in cFS-Repository FM test-and-ground directory results folder.

1. Command, Telemetry, and Events Verification Matrix

| **Command** | **Test Procedure(s)** | **Notes/Comments** |
| --- | --- | --- |
| FM\_NoOp | gencmds |  |
| FM\_ResetCtrs | gencmds |  |
| FM\_FileCopy | filecopy\_basic, filecopy\_stress |  |
| FM\_FileMove | filemove\_basic, filemove\_stress |  |
| FM\_FileRename | filerename\_basic, filerename\_stress |  |
| FM\_Delete | dircmds\_basic, filecopy\_stress, filedelete\_stress, fileinfo\_basic,  openfiles |  |
| FM\_DeleteAll | filedelete\_stress, fileinfo\_basic,  openfiles |  |
| FM\_Decompress | filedecom\_basic, filedecom\_stress |  |
| FM\_FileCat | filecat\_basic, filecat\_stress |  |
| FM\_FileInfo | 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\_ListOpenFiles | openfiles |  |
| FM\_DirCreate | dircmds\_basic, dircmds\_stress, 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, specialchars1 – 5 |  |
| FM\_DirDelete | dircmds\_basic, dircmds\_stress,  specialchars1 - 4 |  |
| FM\_DirListFile | dircmds\_basic, dircmds\_stress; filedelete\_stress, specialchars1 - 5 |  |
| FM\_DirListTlm | dircmds\_basic, dircmds\_stress,  openfiles, specialchars1 - 5 |  |
| FM\_GetFreeSpace | gencmds |  |
| 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\_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\_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 |  |
| FM\_ChildCMDPC | dirfiledisplay, filecopy\_stress, filedecom\_stress |  |
| FM\_ChildCMDEC | dircmds\_basic, dircmds\_stress, filecat\_stress, filecopy\_stress, filedecom\_stress, 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 |  |
| 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\_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 |  |
| FM\_FreeSpacePkt[].Lower32 | gencmds |  |
| FM\_FreeSpacePkt[].Name | gencmds |  |

| **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, specialchars1 – 5 |  |
| FM\_FileListEntry[].FileSize | dircmds\_basic, dircmds\_stress, filedelete\_stress, specialchars1 – 5 |  |
| FM\_FileListEntry[].LastModTime | dircmds\_basic, dircmds\_stress, filedelete\_stress, specialchars1 – 5 |  |
| FM\_FreeSpaceTBL[].State | badtblloadfile, gencmds, tableloadfile |  |
| FM\_FreeSpaceTBL[].Name | badtblloadfile, gencmds, 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\_stress, fileinfo\_basic, fileinfo\_stress, filemove\_basic, filemove\_stress, filerename\_basic, filerename\_stress, gencmds, openfiles, specialchars1 - 5 |  |
| **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 |  |  |
| **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, specialchars1 - 5 |  |
| **17** | FM\_COPY\_PKT\_ERR\_EID | filecopy\_stress |  |
| **18** | FM\_COPY\_OVR\_ERR\_EID |  |  |
| **19** | FM\_COPY\_SRC\_ERR\_EID | filecopy\_stress, specialchars1 - 5 |  |
| **20** | FM\_COPY\_TGT\_ERR\_EID | filecopy\_basic, filecopy\_stress, specialchars1 - 5 |  |
| **21** | FM\_COPY\_CHILD\_ERR\_EID |  |  |
| **22** | FM\_COPY\_OS\_ERR\_EID | dircmds\_stress, filecopy\_stress, specialchars1 - 5 |  |
| **23** | FM\_MOVE\_CMD\_EID | filemove\_basic, filemove\_stress, specialchars1 - 5 |  |
| **24** | FM\_MOVE\_PKT\_ERR\_EID | filemove\_stress |  |
| **25** | FM\_MOVE\_OVR\_ERR\_EID |  |  |
| **26** | FM\_MOVE\_SRC\_ERR\_EID | filemove\_basic, filemove\_stress, specialchars1 - 5 |  |
| **27** | FM\_MOVE\_TGT\_ERR\_EID | filemove\_basic, filemove\_stress, specialchars1 - 5 |  |
| **28** | FM\_MOVE\_CHILD\_ERR\_EID |  |  |
| **29** | FM\_MOVE\_OS\_ERR\_EID | filemove\_stress |  |
| **30** | FM\_RENAME\_CMD\_EID | filerename\_basic, filerename\_stress,  specialchars1 - 5 |  |
| **31** | FM\_RENAME\_PKT\_ERR\_EID | filerename\_stress |  |
| **32** | FM\_RENAME\_SRC\_ERR\_EID | dirrename, filerename\_stress,  specialchars1 - 5 |  |
| **33** | FM\_RENAME\_TGT\_ERR\_EID | filerename\_basic, filerename\_stress,  specialchars1 - 5 |  |
| **34** | FM\_RENAME\_CHILD\_ERR\_EID |  |  |
| **35** | FM\_RENAME\_OS\_ERR\_EID | filerename\_stress,  specialchars1 - 5 |  |
| **36** | FM\_DELETE\_CMD\_EID | dircmds\_basic, filecopy\_stress, filedelete\_stress, openfiles, specialchars1 - 5 |  |
| **37** | FM\_DELETE\_PKT\_ERR\_EID | filedelete\_stress |  |
| **38** | FM\_DELETE\_SRC\_ERR\_EID | filedelete\_stress, fileinfo\_basic, openfiles, specialchars1 - 5 |  |
| **39** | FM\_DELETE\_CHILD\_ERR\_EID |  |  |
| **40** | FM\_DELETE\_OS\_ERR\_EID |  |  |
| **41** | FM\_DELETE\_ALL\_CMD\_EID | dircmds\_stress, filedecom\_stress, filedelete\_stress, fileinfo\_basic, openfiles, specialchars1 - 4 |  |
| **42** | FM\_DELETE\_ALL\_WARNING\_EID | filedelete\_stress, fileinfo\_basic, openfiles |  |
| **43** | FM\_DELETE\_ALL\_PKT\_ERR\_EID | filedelete\_stress |  |
| **44** | FM\_DELETE\_ALL\_SRC\_ERR\_EID | filedelete\_stress,  specialchars1 - 4 |  |
| **45** | FM\_DELETE\_ALL\_CHILD\_ERR\_EID |  |  |
| **46** | FM\_DELETE\_ALL\_OS\_ERR\_EID |  |  |
| **47** | FM\_DECOM\_CMD\_EID | filedecom\_basic, filedecom\_stress,  specialchars1 - 5 |  |
| **48** | FM\_DECOM\_PKT\_ERR\_EID | filedecom\_stress |  |
| **49** | FM\_DECOM\_SRC\_ERR\_EID | filedecom\_basic, filedecom\_stress,  specialchars1 - 5 |  |
| **50** | FM\_DECOM\_TGT\_ERR\_EID | filedecom\_basic, filedecom\_stress,  specialchars1 - 5 |  |
| **51** | FM\_DECOM\_CHILD\_ERR\_EID |  |  |
| **52** | FM\_DECOM\_CFE\_ERR\_EID | filedecom\_stress |  |
| **53** | FM\_CONCAT\_CMD\_EID | filecat\_basic, filecat\_stress, specialchars1 - 5 |  |
| **54** | FM\_CONCAT\_PKT\_ERR\_EID | filecat\_stress |  |
| **55** | FM\_CONCAT\_SRC1\_ERR\_EID | filecat\_basic, filecat\_stress, specialchars1 - 5 |  |
| **56** | FM\_CONCAT\_SRC2\_ERR\_EID | filecat\_basic, filecat\_stress, specialchars1 - 5 |  |
| **57** | FM\_CONCAT\_TGT\_ERR\_EID | filecat\_basic, filecat\_stress, specialchars1 - 5 |  |
| **58** | FM\_CONCAT\_CHILD\_ERR\_EID |  |  |
| **59** | FM\_CONCAT\_OS\_ERR\_EID | filecat\_stress |  |
| **60** | 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,  specialchars1 - 5 |  |
| **61** | FM\_GET\_FILE\_INFO\_PKT\_ERR\_EID | fileinfo\_stress |  |
| **62** | FM\_GET\_FILE\_INFO\_SRC\_ERR\_EID | fileinfo\_stress, specialchars1 - 5 |  |
| **63** | FM\_GET\_FILE\_INFO\_CHILD\_ERR\_EID |  |  |
| **64** | FM\_GET\_FILE\_INFO\_WARNING\_EID | fileinfo\_stress |  |
| **65** | FM\_GET\_OPEN\_FILES\_CMD\_EID | dirrename, openfiles |  |
| **66** | FM\_GET\_OPEN\_FILES\_PKT\_ERR\_EID | openfiles |  |
| **67** | 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, specialchars1 - 5 |  |
| **68** | FM\_CREATE\_DIR\_PKT\_ERR\_EID | dircmds\_stress |  |
| **69** | FM\_CREATE\_DIR\_SRC\_ERR\_EID | dircmds\_stress, specialchars1 - 5 |  |
| **70** | FM\_CREATE\_DIR\_CHILD\_ERR\_EID |  |  |
| **71** | FM\_CREATE\_DIR\_OS\_ERR\_EID | dircmds\_stress |  |
| **72** | FM\_DELETE\_DIR\_CMD\_EID | dircmds\_basic, dircmds\_stress, specialchars1 - 4 |  |
| **73** | FM\_DELETE\_DIR\_PKT\_ERR\_EID | dircmds\_stress |  |
| **74** | FM\_DELETE\_DIR\_SRC\_ERR\_EID | dircmds\_stress, specialchars1 - 4 |  |
| **75** | FM\_DELETE\_DIR\_CHILD\_ERR\_EID |  |  |
| **76** | FM\_DELETE\_DIR\_EMPTY\_ERR\_EID | dircmds\_basic, dircmds\_stress |  |
| **77** | FM\_DELETE\_DIR\_OS\_ERR\_EID |  |  |
| **78** | FM\_GET\_DIR\_FILE\_CMD\_EID | dircmds\_basic, dircmds\_stress, filedelete\_stress,  specialchars1 - 5 |  |
| **79** | FM\_GET\_DIR\_FILE\_PKT\_ERR\_EID | dircmds\_stress |  |
| **80** | FM\_GET\_DIR\_FILE\_SRC\_ERR\_EID | dircmds\_basic, dircmds\_stress, filedelete\_stress,  specialchars1 - 4 |  |
| **81** | FM\_GET\_DIR\_FILE\_TGT\_ERR\_EID | dircmds\_stress, specialchars1 - 5 |  |
| **82** | FM\_GET\_DIR\_FILE\_WARNING\_EID | dircmds\_stress, filedelete\_stress |  |
| **83** | FM\_GET\_DIR\_FILE\_CHILD\_ERR\_EID |  |  |
| **84** | FM\_GET\_DIR\_FILE\_OS\_ERR\_EID | dircmds\_stress |  |
| **85** | FM\_GET\_DIR\_PKT\_CMD\_EID | dircmds\_basic, dircmds\_stress, openfiles, specialchars1 - 4 |  |
| **86** | FM\_GET\_DIR\_PKT\_WARNING\_EID | dircmds\_stress |  |
| **87** | FM\_GET\_DIR\_PKT\_PKT\_ERR\_EID | dircmds\_stress |  |
| **88** | FM\_GET\_DIR\_PKT\_SRC\_ERR\_EID | dircmds\_basic, dircmds\_stress, specialchars1 - 4 |  |
| **89** | FM\_GET\_DIR\_PKT\_CHILD\_ERR\_EID |  |  |
| **90** | FM\_GET\_DIR\_PKT\_OS\_ERR\_EID |  |  |
| **91** | FM\_GET\_FREE\_SPACE\_CMD\_EID | gencmds, openfiles |  |
| **92** | FM\_GET\_FREE\_SPACE\_PKT\_ERR\_EID | gencmds |  |
| **93** | FM\_GET\_FREE\_SPACE\_TBL\_ERR\_EID |  |  |
| **94** | FM\_SET\_TABLE\_STATE\_CMD\_EID | gencmds |  |
| **95** | FM\_SET\_TABLE\_STATE\_PKT\_ERR\_EID | gencmds |  |
| **96** | FM\_SET\_TABLE\_STATE\_TBL\_ERR\_EID | gencmds |  |
| **97** | FM\_SET\_TABLE\_STATE\_ARG\_ERR\_EID | gencmds |  |
| **98** | FM\_SET\_TABLE\_STATE\_UNUSED\_ERR\_EID | gencmds |  |
| **99** | FM\_TABLE\_VERIFY\_ERR\_EID | gencmds |  |
| **100** | 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, specialchars1 - 5 |  |
| **101** | FM\_CHILD\_INIT\_ERR\_EID |  |  |
| **102** | FM\_CHILD\_TERM\_ERR\_EID |  |  |
| **103** | FM\_CHILD\_EXE\_ERR\_EID |  |  |
| **104** | 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, specialchars1 - 5 |  |