

Code 582
Flight Software Branch

**CORE FLIGHT SYSTEM
FILE MANAGER APPLICATION
BUILD 2.5.2.0**

**FLIGHT SOFTWARE BUILD VERIFICATION
TEST REPORT**

Flight Software Branch – Code 582

Version 1.0

SIGNATURES

Submitted by:

X

Walt Moleski/582
cFS Flight Software Tester

Approved by:

X

Susanne Strege/582
cFS Flight Software Product Development Lead

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 VERIFICIATON 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

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 2.5.0.0, 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.

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
- 582-2004-001 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

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:

- 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, 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

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.

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.

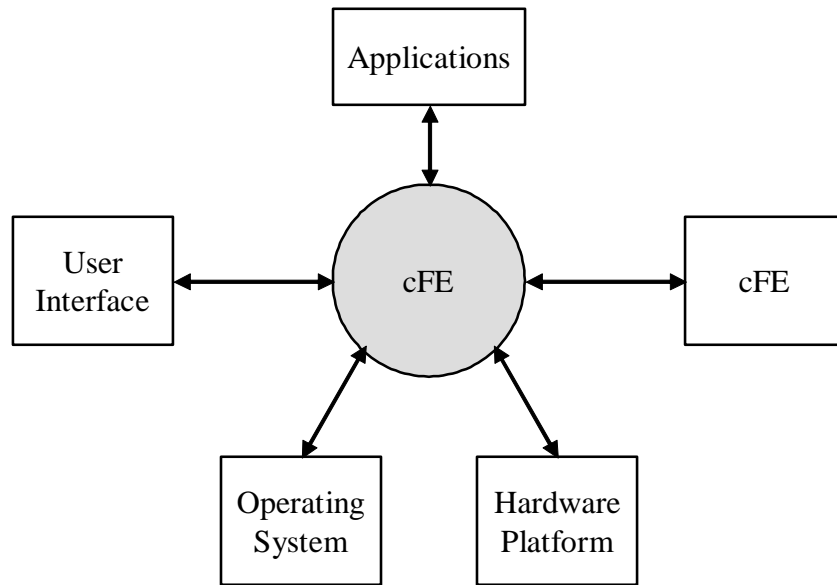


Figure 2-1 cFS System Context

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 (SSR). 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).

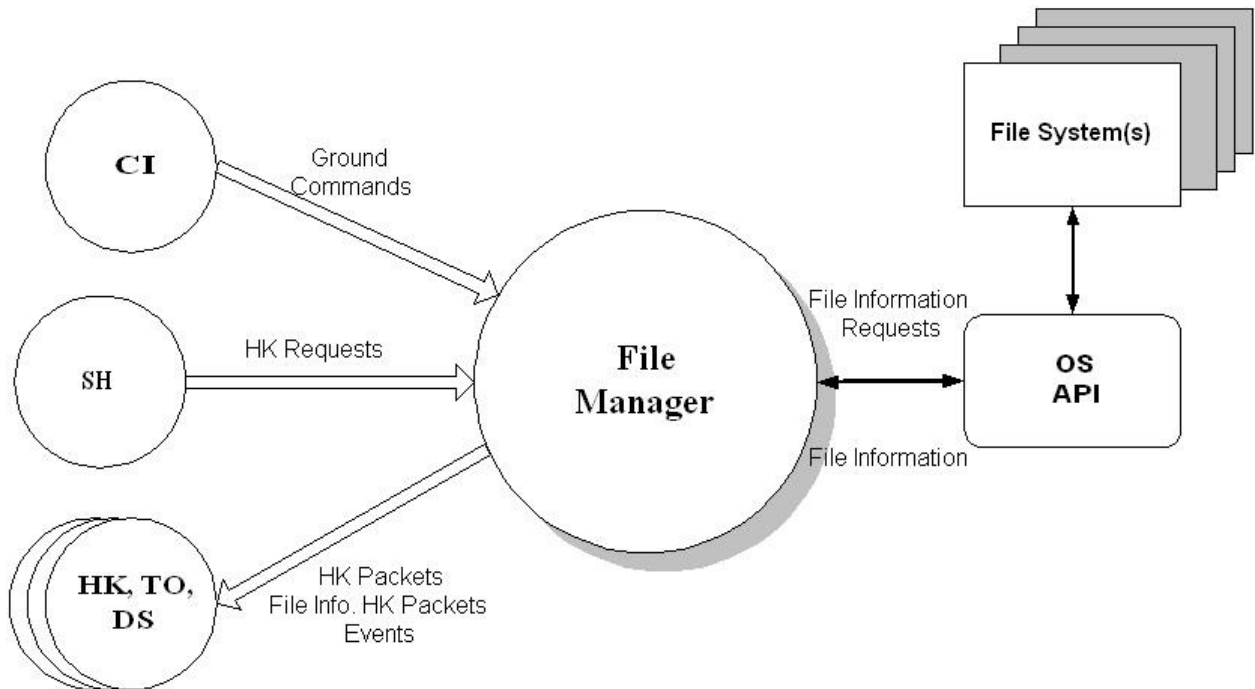


Figure 2.2 – cFS FM Context

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

2.3 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.

Procedure	Description
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.

Procedure	Description
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: <ol style="list-style-type: none"> 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

2.4 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

3 BUILD VERIFICATION TEST PREPARATION

3.1 SCENARIO 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.

3.2 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>`.

3.3 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.

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 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.

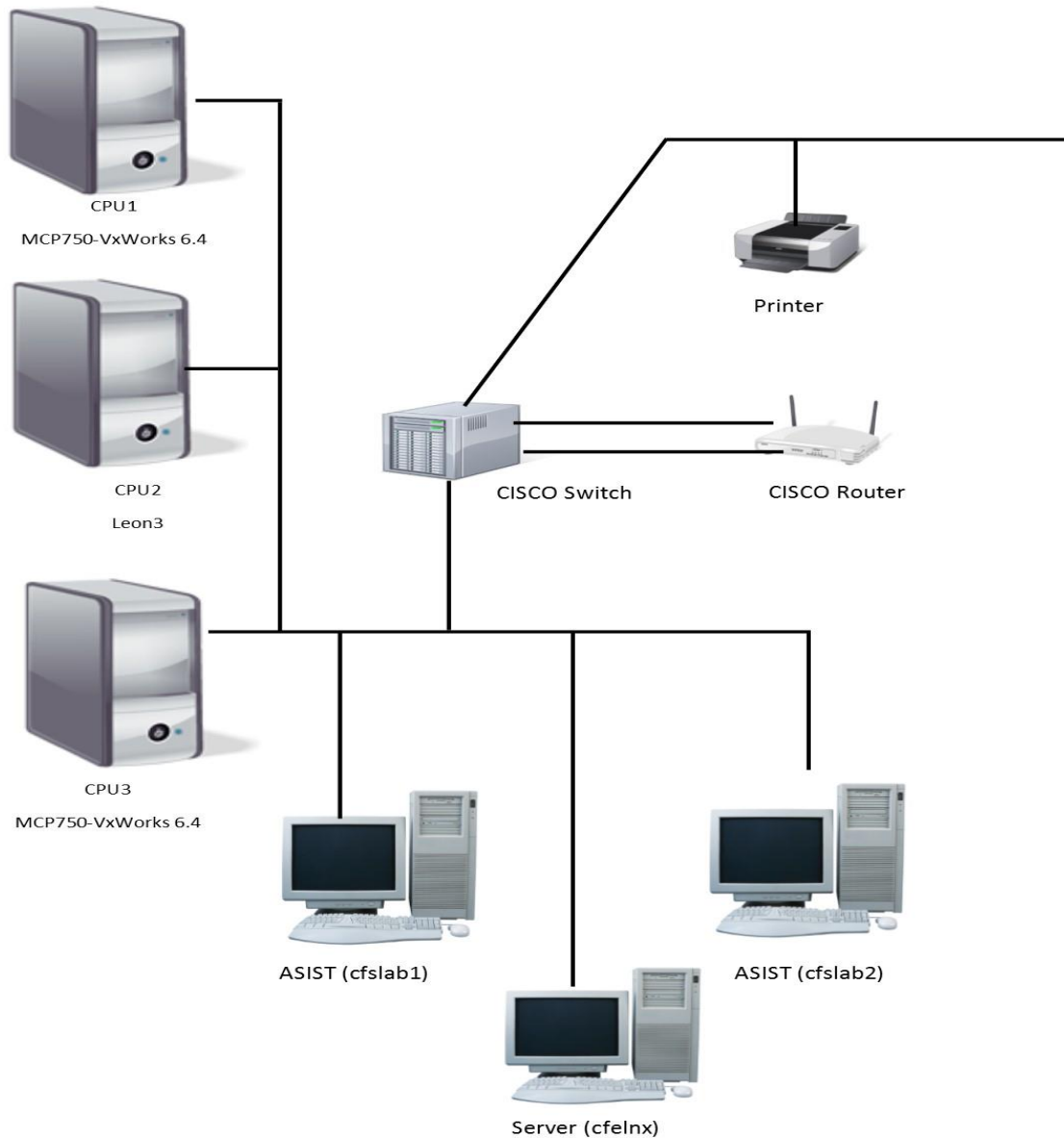


Figure 4-1 cFS FSW Development and Testing Facility

4.2 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

4.3 REQUIREMENTS PARTIALLY TESTED

No requirements were partially tested.

4.4 REQUIREMENTS/FUNCTIONALITY DEFERRED

No requirements were deferred for later build testing

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 VERIFICATION 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:

- 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

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_filerefname_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_filerefname_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

Procedure	Description	Requirements Tested
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

Procedure	Description	Requirements Tested
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: <ol style="list-style-type: none"> 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 ANALYSIS REQUIREMENTS VERIFICATION

No requirements were verified using analysis.

5.4 FAILED REQUIREMENTS

No requirements failed during the final FM build 2.5.2.0 testing.

5.5 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](#), 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](#), for DCR details.

No new DCRs were generated during FM 2.5.2.0 testing.

5.5.1 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.

5.5.2 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

5.6 NOTES

None.

5.7 FOLLOW-ON

None

APPENDIX A - 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.

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	
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	

Telemetry	Test Procedure(s)	Notes/Comments
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	

Id	Event Message	Test Procedure(s)	Notes/Comments
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		

Id	Event Message	Test Procedure(s)	Notes/Comments
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		

Id	Event Message	Test Procedure(s)	Notes/Comments
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		

Id	Event Message	Test Procedure(s)	Notes/Comments
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	