NASA GSFC FLIGHT SOFTWARE SYSTEMS BRANCH

FSW VERSION DESCRIPTION DOCUMENT

CFS FILE MANAGER (FM) APPLICATION

BUILD: FM 2.5.4

RELEASE DATE: 11/20/2020

1.0 FSW VERSION DESCRIPTION

1.1 PURPOSE AND SUMMARY

The purpose of this build is to continue to refine the cFS File Manager (FM) application product. This build provides various bug fixes and enhancements, but does not include any new functionality. The primary purpose of this release is to ensure compatibility between the FM application and cFS Bootes.

This document serves as the notification of the Build 2.5.4.0 release of the cFS FM application.

File Manager (FM) version 2.5.4 is compatible with cFE builds 6.8.0 and above and OSAL builds 5.0 and above. Note that FM also depends on the cfs_lib (version 2.2.0 or later).

1.2 NEW/CHANGED FUNCTIONALITY IN THIS VERSION

Table 1.2-1 identifies the DCRs that have been implemented in this FSW version. For each DCR the "Key" column shows the corresponding DCR in the GSFC cFS tracking system.

Note that there is one significant change in FM functionality. The "Decompress File" command must now be enabled in the FM platform configuration in order to be used.

Table 1.2-1 – DCRs Implemented in this Version

Key	Summary	Description
GSFCCFS-1030	Remove magic numbers in FM	fm_child.c - magic number (TaskTextLen) in FM_ChildInit
GSFCCFS-1031	FM should check all function arguments for NULL	
GSFCCFS-1123	Update FM doxygen user guide	Suggest updating the doxygen user guide file and adding a doxygen configuration file to allow users to successfully generate the doxygen guide themselves.
GSFCCFS-1142	FM uses OS_FS_* Error Codes (soon deprecated)	FM uses OS_FS_SUCCESS which may soon be deprecated in cFE (details: https://github.com/nasa/osal/issues/262)
GSFCCFS-1134	Update FM to handle Decompress removed from cFE	The decompress functionality is being removed from cFE and placed into a separate library. FM needs the following updates to adapt to this change in cFE: 1. Update documentation to reflect that FM now depends on both cfs_lib and fs_lib 2. Update decompress function calls to use new name of function 3. Add a #define flag to determine whether the decompress command should be included (required for Gateway because the decompress functions have known issues)

GSFCCFS-1139	Update FM to handle CFE_TIME_FS2CFESeconds Deprecation	The functions CFE2FSSeconds and FS2CFESeconds will be deprecated in cFE (details: https://github.com/nasa/cFE/issues/519). Currently, FM uses CFE_TIME_FS2CFESeconds when getting the file time in the GetFileInfo command.
GSFCCFS-1140	FM relies on deprecated items	FM uses the CFE_OS_ERROR error code which will soon be deprecated in cFE (details: https://github.com/nasa/cFE/issues/580) FM also relies on deprecated event types and directory functions
GSFCCFS-1235	FM may have alignment problems on some platforms	FM uses uint8[] for command and telemetry packet headers. This can cause alignment issues (this has been experienced with other apps). Instead of the uint8[], the command and telemetry packets should use the actual header types to ensure alignment.
GSFCCFS-1244	FM has command and telemetry alignment issues	The Copy, Move and Rename commands have a uint16 Overwrite argument that requires an explicit 2-byte (uint16) pad to make the command 32-bit aligned. Also, the Housekeeping packet contains 10 uint8 variables that need an additional 2-byte pad to make it 32-bit aligned.

1.3 MISSING PLANNED FEATURES AND KNOWN PROBLEMS

Table 1.3-1 identifies currently open DCRs that are not addressed in this build. Any workarounds that may apply are identified.

Information on currently open DCRs is available at:

https://etdjira.gsfc.nasa.gov/projects/GSFCCFS/issues

Note that this is a restricted website that requires a server account. Additional DCRs may have been submitted after preparation of this VDD. A cFS FM DCR report containing a listing of open DCRs is available upon request for customers who do not have access to the restricted server. Please contact Elizabeth Timmons, elizabeth.timmons@nasa.gov.

Table 1.3-1 – Currently open DCRs

Key	Summary	Description
GSFCCFS-1121	FM should use OSAL functions to verify filenames	Describe the bug A few commands to the cFE core services, and possibly some cFS applications, are able to create files that cannot then be moved, renamed, deleted, or otherwise by FM. One example is EVS_WRITELOG2FILE (CFE_EVS_FILE_WRITE_LOG_DATA_CC). This command (and all others that take filenames in the cFE core) don't check the validity of a filename; they simply pull the filename from the command and call OS_creat() or OS_open() immediately. FM, on the other hand, calls CFS_IsValidFilename() to check for validity of provided filenames in all commands. So, if a command like EVS_WRITELOG2FILE is used to create a file with an invalid filename, FM cannot then access that file in any way.
GSFCCFS-1146	FM may be impacted by OSAL ticket 344	https://github.com/nasa/osal/issues/344
GSFCCFS-941	FM File Info Command hangs the FM application	The FM main task has an internal queue to pass commands to the FM child task. Most command are executed by the child task since the command execution time is unknown or variable. There have been at least two cases where an FM command somehow broke the FM main task and FM child task communication. The FM main task says the internal queue is full and the child task says it's waiting for the next command. In flight, this problem seemed to go away after 20 minutes and the FM child task reported the 3 queued commands had warnings. When it happened on the ground we didn't wait long enough to see if it would clear up. It appears the sem give/take got confused. Not sure how this can happen. Environment: Vxworks 6.7, CFE 6.4.2, FM 2.4.2.
GSFCCFS-1032	FM return statements not needed for void function	
GSFCCFS-1088	Migrate FM unit tests to distributed UT Assert	

GSFCCFS-1083	Add untar command to FM	A mission has requested that untar capability be added in cFE. In design discussions with the framework team, it was decided that it made sense to pull the decompress capability out of the framework and into a library, and to then add an untar command to the FM app.
GSFCCFS-1026	Reduce redundant code in fm_cmd_utils.c	Most verify functions in fm_cmd_utils contain several instances of checking the FilenameState. Could that code be refactored into its own function that receives the set of valid return codes, and is able to validate the return code or report the errors (maybe a bitmap). This would eliminate a big portion of redundant code. Or could it use switch statements?
GSFCCFS-1058	FM_DELETE_INT_CC appears to be redundant with FM_DELETE_CC	The doxygen comment for FM_DELETE_INT_CC states: "This is a special version of the #FM_DELETE_CC command for use when the command is sent by another application, rather than from the ground. This version of the command will not generate a success event, nor will the command increment the command success counter. The intent is to avoid confusion resulting from telemetry representing the results of delete commands sent by other applications and those sent from the ground." However, this does not appear to be the case. Both FM_DELETE_INT_CC and FM_DELETE_CC call the same functions. It appears that FM_DELETE_INT_CC does increment the command success counter, but does not send an event message from the child task. I think that the need for FM_DELETE_INT_CC needs to be reevaluated and if it is needed, it should be updated to match its description.

GSFCCFS-965	Replace FM Internal Command Code with Internal MID instead	Currently, the FM app has an "internal" command code defined for a delete file request that originates from another app instead of the ground. However, the definition for that command code (FM_DELETE_INT_CC) is located in a the header file fm_msgdefs.h, which is located in the app's "fsw/src" directory, and strictly speaking, not accessible by other apps. Instead of using an internal command code, FM could use an internal message ID (e.g., FM_INTERNAL_CMD_MID) that can be defined in fm_msgids.h, which is located in the platform_inc directory. Other cFS apps could then access that command MID and send the internal delete command to the FM app without reaching into what should be an FM-local header file.
GSFCCFS-1113	Remove the FM Basic Counter Rollover Test Scenario	The FM Basic Counter Rollover Test Scenario exists in the test_and_ground/scenarios directory but a Test Procedure was never implemented. The reason for not implementing the procedure was that the test was deemed unnecessary at the time of the Scenario Review,
GSFCCFS-760	FM Delete Directory Error Requirement	Submitted based on 7/9/18 email. I suspect FM's child task error counter incremented. Even if this is the case the requirement may need some clarification. NASA-FSW-670: "If the specified directory contains at least one file or subdirectory, the command shall be rejected." Issue: The requirement states that the command shall be rejected, which implies that FM's HK_PACKET.ERRCOUNTER should increment. However, when attempting to delete a directory that contains a subdirectory, neither the error counter or the command counter are incremented. The command is rejected as expected, verified by reception of the FM_DELETE_DIR_EMPTY_ERR_EID and checking the HK_PACKET.CMD_COUNTER remains the same. Suggestion: Remove "the command shall be rejected", and replace with "a delete directory error message shall be issued"
GSFCCFS-1062	FM configuration parameter limits need clarification	A number of FM configuration parameters have limits for which the reason is obscure at best. Limits need to be re-evaluated and comments should give clear reasoning for the limit.

2.0 DELIVERED PRODUCTS

Table 2-1 identifies the locations of FSW products relevant to this FSW Build. The version or date of the Build and where the product can be located are provided. Changes from a previous VDD are identified.

Table 2-1 – Delivered Products and their Locations

Software Element	Changed with this Version?	New Version or Date	Location
Source Code of this FSW Build	Yes	2.5.4	https://github.com/nasa/fm
Doxygen Documentation	Yes	N/A	https://github.com/nasa/fm
Unit Test Data	Yes	2.5.4	https://github.com/nasa/fm
FSW Make Files	Yes	2.5.4	https://github.com/nasa/fm

3.0 INSTALLATION PROCEDURES

In order to build and install the FM application, it must be added to the cFE CMake build system. This is done by modifying the TGTX_APPLIST in the cFE targets.cmake file. This is shown in the trivial example below.

```
SET(TGT1_NAME cpu1)
SET(TGT1_APPLIST cfs_lib fm)
SET(TGT1_FILELIST cfe_es_startup.scr)
```

After FM is added to the targets.cmake file, it is built and installed using the standard cFE CMake build instructions. These instructions are available in cFE CMake documentation:

https://github.com/nasa/cFE/blob/main/cmake/README.md

4.0 CONFIGURATION SUMMARY AND VERSION IDENTIFICATION

This software can be found in the FM GitHub repository (https://github.com/nasa/FM) under the tag "2.5.4".

Verification of the version can be done by sending an FM NOOP command which produces an event message containing the version information. In addition, the initialization event message generated during the application startup provides the version information.

ACRONYMS

ACS	Attitude Control System
C&DH	Command and Data Handling
cFS	Core Flight System
CM	Configuration Management
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
DCR	Discrepancy/Change Request
ETU	Engineering Test Unit
FM	File Manager
FSB	Flight Software Branch
FSW	Flight Software
GSFC	Goddard Space Flight Center
I&T	Integration & Test
JSC	Johnson Space Center
POSIX	Portable Operating System Interface
RTOS	Real-Time Operating System
SMP	Symmetric Multiprocessing
T&C	Telemetry and Command
TBD	To Be Determined
URL	Universal Resource Locator
VDD	Version Description Document