NASA GSFC FLIGHT SOFTWARE SYSTEMS BRANCH

FSW VERSION DESCRIPTION DOCUMENT

CFS FM APPLICATION

BUILD: FM 2.5.3

RELEASE DATE: APRIL 1, 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.

This document serves as the notification of the Build 2.5.3.0 release of the cFS FM application.

File Manager (FM) version 2.5.3 is compatible with cFE builds 6.7.0 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.

Table 1.2-1 – DCRs Implemented in this Version

Key	Summary	Description
GSFCCFS-1063	FM Doxygen comments are out of date	FM doxygen comments are out of date. The attached file of discrepancies was provided by Gary Smith. The findings are from his efforts to rewrite the FM User Guide.
GSFCCFS-1035	Change parameter type in FM_ValidateTable	Why not pass TableData as a FM_FreeSpaceTable_t * directly and avoid this line? This would make the function more strongly typed and provide compiler warnings etc.. if an invalid pointer type is ever passed. Finding from JSC code review
GSFCCFS-1034	FM readme needs updates	FM readme needs to be scrubbed and updated to reflect current cFE/OSAL repositories
GSFCCFS-1033	CMakeLists.txt does not build FM tables	Finding from JSC code review
GSFCCFS-1029	if-else-if chains should end with "Else"	Finding from JSC code review. fm_child.c, line 596 fm_cmd_utils.c, lines 469, 427, 299
GSFCCFS-1028	Potential exclusion of null-terminator in FM	Finding from JSC code review. fm_child.c, line 1265, line 74
GSFCCFS-1027	Replace strcat with strncat in FM	Finding from JSC code review
GSFCCFS-1025	Replace all strcpy with strncpy in FM	Finding from JSC code review
GSFCCFS-1024	Uninitialized variables in FM	Finding from JSC code reviews

GSFCCFS-971	FM_ChildTask sends event if child can't be registered	In the FM_ChildTask function, an event message is sent if the child task can't be registered. However, the CFE_EVS_SendEvent call should not be sent if the child task is not registered. Recommend changing the error logging to CFE_ES_WriteToSyslog
GSFCCFS-928	FM Test Failure: UT_FM_CMDS_TEST_CFE_OSFILEAPI_StatHookIsDirectory()	Submitted by Allen Brown: A unit test was failing, UT_FM_CMDS_TEST_CFE_OSFILEAPI_StatHookIsDirectory(), because there was an assumed/uninitalized value. In that function, filestats->st_size = 0; needs to be added.
GSFCCFS-749	FM_VerifyFileClosed_Test_FileClosed() fails	In FM_VerifyFileClosed_Test_FileClosed(), the function under test eventually results in calling the OS_FDGetInfo() function which defaults to OS_FS_SUCCESS, flagging the file as open and failing the unit test. I would recommend setting the OS_FDGetInfo() return to OS_FS_ERR_INVALID_FD, but I'm thinking some of these errors I'm finding are due to utassert mismatches. There are at least 3 different directories in the CFS product line called "ut_assert" or "ut-assert". The one if the CF directory defaults to returning OS_FS_ERR_INVALID_FD. The one in the CFE tools directory defaults to OS_FS_SUCCESS. Are we supposed to use the ut-assert in the CF directory for all applications, or the one in the CFE tools directory?

GSFCCFS-748	FM unit test wrong strncpy parameter	(part of babelfish ticket #95) Function UT_FM_CMD_UTILS_TEST_CFE_ES_GetTaskInfoHook() has the following line: strncpy ((char *)TaskInfo->AppName, "appname", OS_MAX_PATH_LEN); However, TaskInfo->AppName is declared as: uint8 TaskName[OS_MAX_API_NAME] I recommend changing that n parameter in the strncpy to either OS_MAX_API_NAME or sizeof(TaskInfo->AppName). My OS_MAX_PATH_LEN is larger than OS_MAX_API_NAME, causing a buffer overrun and stack smashing.
GSFCCFS-747	FM_ProcessCmd_Test_GetOpenFiles() unit test is causing stack smashing	The FM_ProcessCmd_Test_GetOpenFiles() unit test is causing stack smashing because its acting on uninitialized data. This calls FM_GetOpenFilesData() which calls the OS_FDGetInfo() function which has no return code or hook specified, causing it to return OS_FS_SUCCESS. This causes the conditional to determine the entry is of an open file, which calls the following: strcpy(OpenFilesData[OpenFilesCount].LogicalName, FDTableEntry.Path) The file wasn't actually open and OS_FDGetInfo() didn't actually set FDTableEntry, so this variable, also instantiated on stack, is random data. Since its using strcpy rather than strncpy, its usually overrunning the buffer causing stack smashing. The results of the command aren't verified. Only the fact that the command was executed. So I recommend the unit test setup include a step to either set the OS_FDGetInfo() return code to OS_FS_ERR_INVALID_FD or add a hook to return OS_FS_ERR_INVALID_FD. I also recommend replacing the strcpy() calls with a safer function.

GSFCCFS-1120	FM compatibility issue with OSAL	While integrating File Manager app 2.5.2 with cFE 6.7.1 (OSAL 5.0.1) I ran into an issue. FM has a command that allows users to receive a telemetry packet listing all of the open files. In order to do this FM needs to be able to query OSAL's file stream resource objects.
GSFCCFS-1112	FM uses the same event ID for multiple events	In several places, FM uses the same event ID for different events (sometimes up to 6). While the multiple events are properly documented in the fm_events.h file, this still violates the general guidance to have unique event IDs.
GSFCCFS-746	FM unit tests Test(s) are not agnostic to configuration	(part of babelfish ticket #95) Test(s) are not agnostic to configuration i.e. the assertion that reads: UtAssert_True (Ut_CFE_EVS_EventSent(FM_TABLE_VERIFY_EID, CFE_EVS_INFORMATION, "Free Space Table verify results: good entries = 0, bad = 8, unused = 0"),"Free Space Table verify results: good entries = 0, bad = 8, unused = 0"); assumes the FM_TABLE_ENTRY_COUNT macro in the fm_platform_cfg.h file is set to 8. With the following change: char ExpectedEventText[CFE_EVS_MAX_MESSAGE_LENGTH]; ... snprintf(ExpectedEventText, CFE_EVS_MAX_MESSAGE_LENGTH, "Free Space Table verify results: good entries = 0, bad = %u, unused = 0", FM_TABLE_ENTRY_COUNT); UtAssert_True (Ut_CFE_EVS_EventSent(FM_TABLE_VERIFY_EID, CFE_EVS_INFORMATION, ExpectedEventText), ExpectedEventText); users can run the unit tests with their own fm_platform_cfg.h and run the unit tests on the as-built, by including their copy of fm_platform_cfg.h rather than the one included with fm.

GSFCCFS-1092	FM has compilation error with cFE6.7.0	Looks like a struct in OSAL has changed which is causing an error in FM.
GSFCCFS-745	FM Unit Test Issue #1	(part of babelfish ticket #95) The FM_ValidateTable_Test_InvalidState() test allocates the FM_FreeSpaceTable_t TableData variable on the stack but the assumes the contents are zero'd out. In the FM_ValidateTable() function first checks the FileSys[i].State value. Since its allocated on stack, it contains garbage data causing it to have random behavior. Recommend adding the following line before the line that states the State to 99: memset(&TableData, 0, sizeof(TableData));
GSFCCFS-1129	FM DirListToFile command fails	Issue with failing command found when testing FM v2.5.3. I attempted to run the first FM BVT and it is now failing on a command that passed previously. The Dir List To File command now fails to create the specified file. I am getting EID 68 every time I send the command. The OS error is -2 that is being reported in the event message. The log file is attached. To find errors, search for "<!>". I looked back in the previous test and this command passed. If you need that log file, let me know and I can send it to you. A little more investigation reveals that the command seems to be dropping the output filename. If you look at the command sent and the error event message, FM is attempting to create a NULL filename.
GSFCCFS-1128	Duplicated event ID in FM	An error in the new base+offset EID convention found in build testing of FM v2.5.3. #define FM_COPY_SRC_BASE_EID 100 #define FM_COPY_SRC_INVAL_ERR_EID (FM_COPY_SRC_BASE_EID + FM_FNAME_INVALID_EID_OFFSET) 100 #define FM_COPY_SRC_DNE_ERR_EID (FM_COPY_SRC_BASE_EID +

		FM_FNAME_DNE_EID_OFFSET) 101 #define FM_COPY_SRC_ISDIR_ERR_EID (FM_COPY_SRC_BASE_EID + FM_FNAME_ISDIR_EID_OFFSET) 102 #define FM_COPY_SRC_UNKNOWN_ERR_EID (FM_COPY_SRC_BASE_EID + FM_FNAME_UNKNOWN_EID_OFFSET) 105 #define FM_COPY_TGT_BASE_EID (FM_COPY_SRC_BASE_EID + FM_FNAME_NUM_OFFSETS) 106 #define FM_COPY_TGT_INVAL_ERR_EID (FM_COPY_TGT_BASE_EID + FM_FNAME_INVALID_EID_OFFSET) 106 #define FM_COPY_TGT_EXIST_ERR_EID (FM_COPY_TGT_BASE_EID + FM_FNAME_ISOPEN_EID_OFFSET) 109 #define FM_COPY_TGT_ISDIR_ERR_EID (FM_COPY_TGT_BASE_EID + FM_FNAME_ISDIR_EID_OFFSET) 108 #define FM_COPY_TGT_UNKNOWN_ERR_EID (FM_COPY_TGT_BASE_EID + FM_FNAME_UNKNOWN_EID_OFFSET) 111 #define FM_COPY_CHILD_BASE_EID (FM_COPY_SRC_BASE_EID + FM_FNAME_NUM_OFFSETS) 106 #define FM_COPY_CHILD_DISABLED_ERR_EID (FM_COPY_CHILD_BASE_EID + FM_CHILD_DISABLED_EID_OFFSET) 106 However I think this line should be #define FM_COPY_CHILD_BASE_EID (FM_COPY_TGT_BASE_EID + FM_FNAME_NUM_OFFSETS)

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	Change Request Type
GSFCCFS-1032	FM return statements not needed for void function	Finding from JSC code review	Enhancement
GSFCCFS-1117	FM 2005.1 missing handling of overwrite flag	From IV&V Gap Analysis: In the FM2005.1 requirement ID, “If the command-specified destination file exists, FM shall reject the command.” The rationale behind the requirement serves in preventing files being overwritten. A user must delete the file first before performing this operation. At line 270 of the fm_cmds.c file, the function FM_VerifyFileNoExists() is called and if the file exists, a child task is invoked. However, there is no flag to set this, therefore, there is a missing overwrite flag. This should have possibly been in the same form as FM2004.1.	Requirement Mod
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.	Enhancement
GSFCCFS-1031	FM should check all function arguments for NULL	Finding from JSC code review	Defect

GSFCCFS-1030	Remove magic numbers in FM	fm_child.c - magic number (TaskTextLen) in FM_ChildInit	Defect
GSFCCFS-1133	FM functionality does not match requirements	During build verification testing of FM v2.5.3, it was discovered that although FM requirement 2008.1 was marked as "DELETED", the restriction is still enforced in the code (FM will not delete open files). Upon further investigation, it was discovered that the following requirements from the MKS record had also been marked as "DELETED" and therefore not migrated to Jira. Of these requirements, all of the restrictions are still enforced in the code except the restriction on the Move command. FM2002.2 DELETED: If the command-specified source file is open, FM shall reject the command [related to the copy command] FM2004.2 DELETED: If the command-specified source file is open, FM shall reject the command [related to the move command] FM2005.2 DELETED: If the command-specified source file is open, FM shall reject the command [related to the rename command] FM2008.1 DELETED: If the command-specified file is open, FM shall reject the command to delete the file. [related to the delete command] The rationale for deleting these requirements is unknown. The current issue is that the FM code does not match the requirements, and the behavior is not consistent among similar cases.	Defect

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.	Enhancement

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.	Enhancement
GSFCCFS-1083	Add untar command to FM	Project customer 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.	Enhancement

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. The NICER instrument has experience 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. NICER Is Vxworks 6.7, CFE 6.4.2, FM 2.4.2.	Defect
GSFCCFS-1088	Migrate FM unit tests to distributed UT Assert		Enhancement

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. Message Body: 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"	Requirement Mod
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.	Enhancement

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	2.5.3	https://github.com/nasa/fm
Doxygen Documentation	Yes	N/A	https://github.com/nasa/fm
Unit Test Data	Yes	2.5.3	https://github.com/nasa/fm
FSW Make Files	Yes	2.5.3	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/master/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 “2.5.3”.

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