
**NASA GSFC FLIGHT SOFTWARE SYSTEMS BRANCH**

# FSW VERSION DESCRIPTION DOCUMENT

## CFS MEMORY MANAGER (MM) APPLICATION

**BUILD: MM 2.5.0**

**RELEASE DATE: 9/3/2021**

# 1.0 FSW VERSION DESCRIPTION

## 1.1 PURPOSE AND SUMMARY

The purpose of this build is to continue to refine the cFS Memory Manager (MM) 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 MM application and cFS Caelum.

This document serves as the notification of the Build 2.5.0 release of the cFS MM application.

## 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-1184 | MM has static code analysis findings | In analysis done on 7/10/2020, CodeSonar flagged the attached findings. |
| GSFCCFS-1262 | Update MM to use new cFE Message Module | |
| GSFCCFS-1422 | Migrate MM unit tests to distributed UT Assert | |
| GSFCCFS-1482 | MM does not build with eval-cert3 | |
| GSFCCFS-1498 | MM Uses Deprecated Boolean Values | MM needs to be updated in the following way:<br>`TRUE` should be `true`<br>`FALSE` should be `false`<br><br>This is applicable in mm_platform_cfg.h and mm_verify.h |
| GSFCCFS-1538 | MM configuration parameters should be verified where possible | Many MM mission and configuration parameters are not verified in the mm_verify.h file. At minimum, many should be verified against the maximum value allowed by their type in the code. Parameters related to length or size also need to be verified to be 32-bit aligned. |
| GSFCCFS-1540 | MM stores pipe config options in AppData unnecessarily | In MM_AppInit, pipe configuration options PipeDepth, LimitHk, and LimitCmd are stored when it appears they are unused. |
| GSFCCFS-1541 | MM should always use event messages after EVS registration | MM uses syslog messages throughout the AppInit function even after the app has registered with event services. |

| | | |
|---|---|---|
| GSFCCFS-1542 | MM should check return codes | Currently MM does not check the return code of CFE_MSG_GetFcnCode. On error, CFE_MSG_GetFcnCode sets the command code to 0 (noop). While this is not harmful, it does prevent the error from being accurately captured.<br><br>MM also does not currently check the return code of OS_lseek<br><br>Additionally, return codes need to be checked against the correct return values (CFE_PSP_* functions should check for CFE_PSP_SUCCESS not OS_SUCCESS) |
| GSFCCFS-1543 | MM: Use actual type instead of cast | In MM_HousekeepingCmd, when calling MM_VerifyCmdLength, better to pass &BufPtr->Msg rather than case the BufferPtr_t.<br><br>Alternatively, make the MM_HousekeepingCmd signature consistent with the command handling functions. |
| GSFCCFS-1544 | MM store housekeeping counters directly in packet | MM can store housekeeping counters directly in the HK packet rather than separately in the AppData. Avoids having to copy them when sending out the HK packet. |
| GSFCCFS-1545 | MM files may not include header | The file generated in MM_SymTblToFileCmd may not include the expected cFE file header. |
| GSFCCFS-1548 | MM should use const for function arguments where possible | |
| GSFCCFS-1550 | MM pipe depth should be based on the default msg limit | Pipe depth should be based on the default message limit |
| GSFCCFS-1551 | MM is inconsistent in where command and command err counts are incremented | There is no consistent approach taken to where command and command error counters are incremented. This has led to several places where counters are incremented incorrectly |
| GSFCCFS-1552 | MM uses cpuaddr and uint32 inconsistently | |
| GSFCCFS-1553 | MM event messages don't reference mem type consistently | |
| GSFCCFS-1554 | MM_OPT_CODE_XXX macros not used consistently | There appear to be cases where the MM_OPT_CODE_XXX macros are not used where they should be. |
| GSFCCFS-1555 | MM header subtype not checked | |

| | | |
|---|---|---|
| | when loading files | |
| GSFCCFS-1557 | Macro format not robust | Throughout the app, the following macro format is used:<br>#if (MM_OPT_CODE_XXX_MEMTYPE == true)<br><br>Better to use #if defined(MM_OPT_CODE_MEM32_MEMTYPE) to avoid error where "true" isn't known to preprocessor. If MM_OPT_CODE_MEM32_MEMTYPE is missing, then the evaluation will succeed in that case. (0 == 0) |
| GSFCCFS-1560 | MM should use memcpy instead of CFE_PSP_MemCpy | |
| GSFCCFS-1563 | MM contains an untestable condition | In MM_DumpInEventCmd, there's an untestable branch condition in the for loop at line 762. CmdPtr->NumOfBytes was previously validated in MM_VerifyDumpInEventParams to always be <= MM_MAX_DUMP_INEVENT_BYTES. The '&&' portion of this condition check will always evaluate to true. This was discovered with unit test code coverage. |
| GSFCCFS-1565 | Avoid strcat | In MM_DumpInEventCmd, uses strcat where not needed. |
| GSFCCFS-1566 | MM_FillDumpInEventCmd should check argument for NULL | |
| GSFCCFS-1567 | MM_RESET_DBG_EID should be info | MM_RESET_DBG_EID should probably be informational to be consistent with the other command acknowledgment EIDs. |
| GSFCCFS-1572 | MM FillMem32 does not check command argument alignment | Command should check that the NumOfBytes argument is a multiple of 4. |
| GSFCCFS-1575 | Intended function of CFS_ResolveSymAddr needs clarification. | Is the intended functionality here being able to overwrite a function (looked up via symbol) with given memory? There is no size check that the new symbol data fits within the old symbol space. There's no unloading a symbol and loading it back. There's no check for dynamic dependencies for the new code.<br><br>At minimum, if this is expected to be an operational constraint, that needs to be made clear in documentation. |

| | | |
|---|---|---|
| GSFCCFS-1576 | MM doxygen config file should be renamed for clarity | The filename "mm_config.txt" suggests that this a configuration file for the app itself as opposed to a configuration file for doxygen. |
| GSFCCFS-1604 | MM has compiler error with older compilers | On Centos 7/gcc 4.8.5, MM has the following compiler error:<br><br>[ 91%] Building C object apps/mm/CMakeFiles/mm.dir/fsw/src/mm_app.c .o<br>/home/tbrain/repos/cert_testbed/apps/mm/fsw/src /mm_app.c: In function 'MM_AppPipe':<br>/home/tbrain/repos/cert_testbed/apps/mm/fsw/src /mm_app.c:286:10: error: passing argument 1 of 'MM_HousekeepingCmd' from incompatible pointer type [-Werror]<br>MM_HousekeepingCmd(&BufPtr->Msg);<br>^<br>In file included from<br>/home/tbrain/repos/cert_testbed/apps/mm/fsw/src /mm_app.c:64:0:<br>/home/tbrain/repos/cert_testbed/apps/mm/fsw/src /mm_app.h:169:6: note: expected 'struct CFE_MSG_CommandHeader_t *' but argument is of type 'union CFE_MSG_Message_t *'<br>void<br>MM_HousekeepingCmd(CFE_MSG_Command Header_t* msg); |
| GSFCCFS-1614 | MM documentation should clarify that SymbolTableMap feature doesn't work on Linux | |
| GSFCCFS-1625 | MM event messages do not allow for extended message IDs | Events that print out a messageID value use the 0x04X format specifier, which does not work for longer message IDs. |
| GSFCCFS-1644 | Prevent MM from using long filenames for dumps. | During testing, MM permitted a symbol table dump file with a 21 character filename. The file consequently cannot be processed by FM or other apps that hold to the 20 character limit.<br>In MM_SymTblToFileCmd, the filename is only checked against OS_MAX_PATH_LEN. Subsequently the file is created via a call to OS_SymbolTableDump which apparently does not check the filename length either.<br>A possible solution is to explicitly check the filename length in this function. Other MM code should be checked as well. |
| GSFCCFS-1704 | Update unit tests to latest cFS | Since making the main update to distributed UT-Assert, some other changes to UT-Assert have |

| | | |
|---|---|---|
| | | been made that require additional updates. |
| GSFCCFS-1705 | Fix Incorrect Return Status Values | Certain MM functions that return boolean status values are not setting the value to false when errors are detected. Others are not initializing the return status variable to the correct value resulting in an incorrect status value being returned. One function is conditionally checking a return status value that is always true. |
| GSFCCFS-1711 | MM should not pend forever on the software bus | |

## 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 MM 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 | Type |
|---|---|---|---|
| GSFCCFS-1613 | MM Should include the status of the EEPROM write enable in housekeeping telemetry | MM includes commands to enable or disable writing to EEPROM. That setting cannot currently be verified in telemetry. The status of the write enable for each EEPROM bank should be added to the housekeeping packet. | Enhancement |
| GSFCCFS-1574 | Unnecessary use of CFS_IsValidFilename | OS_OpenCreate will fail if the filename was invalid - no need to check it twice. | Enhancement |
| GSFCCFS-1573 | Change order of size checks for optimization | In MM_VerifyLoadWIDParams, it is more efficient to check SizeInBytes before calling CFE_PSP_MemValidateRange. This same optimization would be applicable in other functions. | Enhancement |

| | | | |
|---|---|---|---|
| GSFCCFS-1571 | Perf counter in MM_SegmentBreak may be unnecessary | The performance marker only wraps a task delay. | Enhancement |
| GSFCCFS-1570 | MM clarify names of constants | mm configuration parameters for sizes often don't include the word SIZE, making their purpose a little less clear. | Enhancement |
| GSFCCFS-1569 | Function names MM_ResetHk and MM_WriteMemWID are misleading | | Enhancement |
| GSFCCFS-1568 | Make MM MemType into an enum | | Enhancement |
| GSFCCFS-1564 | MM_DumpInEventCmd wastes event characters | In MM_DumpInEventCmd, including the "0x" and trailing space on the hex digits unnecessarily limits how much can actually be dumped in the event. | Enhancement |
| GSFCCFS-1562 | MM Investigate use of static variables | In MM_DumpInEventCmd, EventString is declared as static to keep it off the stack. Need to investigate whether this is necessary. | Enhancement |
| GSFCCFS-1561 | Generally better to initialize return code to false | | Enhancement |
| GSFCCFS-1558 | In MM_DumpMemToFileCmd, would be faster to checksum while writing | | Enhancement |
| GSFCCFS-1556 | In MM_DumpMemToFile command could use CFE_FS_InitHeader | | Enhancement |
| GSFCCFS-1546 | Investigate whether Eeprom and Symtable handing belongs in MM | Some discussion needed in the future to reevaluate whether Eeprom and symbol table handling belongs in MM since not every platform has them. | Enhancement |
| GSFCCFS-1132 | Reevaluate how MM uses intlock | Memory Manager uses OS_IntLock() and OS_IntUnlock(). That's the only place any of the OS_Int* functions are used in the GSFC apps. These functions will likely be deprecated in the next OSAL release.<br><br>I'm guessing it does this with the intent of loading the mem block "atomically" with the hope of preventing another task from writing | Defect |

| | | | |
|---|---|---|---|
| | | to it while this happens?<br><br>IntLock/Unlock has always been a no-op on POSIX, and furthermore even on platforms where it does something, it will not achieve that exclusivity effect on Multi-Core CPUs, as it only locks the interrupts on the core which calls it, and other cores continue to run anyway, interrupt or not.<br><br>Probably worth re-evaulating what MM is trying to achieve with the intlock... might be able to simply take it out with no loss of function. | |
| GSFCCFS-925 | MM Unclear app configuration assumptions | Discovered and added comments to the config file that some app configuration values have assumptions on modulo-2 or modulo-4 sizes. For example, these 16 and 32-bit dump routines would fail if these config #defines are not set with the appropriate values. | 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 | 2.5.0 | https://github.com/nasa/mm |
| Doxygen Documentation | Yes | N/A | https://github.com/nasa/mm |
| Unit Test Data | Yes | 2.5.0 | https://github.com/nasa/mm |
| FSW Make Files | Yes | 2.5.0 | https://github.com/nasa/mm |

## 3.0 INSTALLATION PROCEDURES

In order to build and install the MM 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 mm)
SET(TGT1_FILELIST cfe_es_startup.scr)
```

After MM 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 MM GitHub repository (https://github.com/nasa/MM) under the tag "2.5.0".

Verification of the version can be done by sending an MM 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
FSB ..... Flight Software Branch
FSW ..... Flight Software
GSFC ..... Goddard Space Flight Center
I&T ..... Integration & Test
JSC ..... Johnson Space Center
MD ..... Memory Dwell
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