



NASA GSFC FLIGHT SOFTWARE SYSTEMS BRANCH

FSW VERSION DESCRIPTION DOCUMENT

CFS STORED COMMANDS (SC) APPLICATION

BUILD: SC 3.0.0

RELEASE DATE: NOVEMBER 16, 2020

1.0 FSW VERSION DESCRIPTION

1.1 PURPOSE AND SUMMARY

The purpose of this build is to continue to refine the cFS Stored Commands (SC) 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 SC application and cFS Bootes.

This document serves as the notification of the Build 3.0.0 release of the cFS SC application.

Stored Commands (SC) version 3.0.0 is compatible with cFE builds 6.8.0 and above and OSAL builds 5.0 and above.

NOTE: This version of SC alters the format of the tables used for Relative Time Sequences (RTSs) and Absolute Time Sequences (ATSs). This is a **breaking change** that will prevent tables from previous versions of the application from working with this version of the application. The RTS tables now use a 32-bit time tag and ATS tables now use a 32-bit command number. Additionally, the SC application only supports commands that are 32-bit aligned. These changes are necessary in order to be compatible with the strict alignment enforced by cFS Bootes.

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-905	SC must be tested with cFE 6.6	
GSFCCFS-1073	SC Readme needs updates	Readme file includes out of date information on sources for cFE and OSAL.
GSFCCFS-1158	SC does not build against cFE 6.8 with OMIT_DEPRECATED=true and -Werror	
GSFCCFS-1187	SC Pointer precision loss casting pointer down to (int)	Pointer precision loss casting pointer down to (int), causing segfault on 64-bit. See: SC/fsw/src/sc_rtsrq.c Line 641 in 43392d7 CFE_SB_SendMsg((CFE_SB_MsgPtr_t)((int)&CmdPkt));
GSFCCFS-1215	Add files to generate SC doxygen guide	SC needs config file to generate doxygen user guide
GSFCCFS-1227	SC does not build tables in cmake build	The SC CMakeLists.txt file does not build the SC tables.

GSFCCFS-774	Full RTS buffer drops final command	<p>A mission recently encountered an error in the SC application. An RTS was loaded on Flatsat with a series of short (8-byte) commands that completely filled the RTS buffer. On this mission, an RTS buffer is 300 bytes long, so 30 commands consisting of a 2-byte time tag plus 8-byte CCSDS command will fill the buffer with no space left over. When the RTS was run, only 29 commands executed. The final command was dropped without warning.</p> <p>The cause of the problem appears to be as follows: In the function SC_GetNextRtsCommand(), when looking for the next command in an RTS, the code performs an end-of-buffer check using a < comparison when it should use a <= comparison, and thus treats the end of the buffer as occurring one word (2 bytes) too soon. The minimum length of an RTS command is 10 bytes, but the code skips the contents starting at byte 290 as not being long enough to be a valid command and exits the RTS without executing the command.</p> <p>In such a case, one would still expect the supposedly incomplete command at the end of the buffer to be rejected as an error. However, the code that validates the RTS in the function SC_ParseRts() performs the length computations correctly and doesn't detect an error. As a result, the final command in the RTS is ignored without any warning.</p>
GSFCCFS-1223	SC does not compile on test machine	<p>SC experiences alignment problems on testing machine.</p> <p>sc/fsw/src/sc_state.c: In function 'SC_GetNextRtsCommand': sc/fsw/src/sc_state.c:187: error: cast increases required alignment of target type sc/fsw/src/sc_state.c:213: error: cast increases required alignment of target type sc/fsw/src/sc_state.c:293: error: format '%ld' expects type 'long int', but argument 5 has type 'unsigned int'</p>

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://etd.jira.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 SC 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-1185	SC has static code analysis findings	In analysis done on 7/10/2020, CodeSonar flagged the attached findings.

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	3.0.0	https://github.com/nasa/SC
Doxygen Documentation	Yes	N/A	https://github.com/nasa/SC
Unit Test Data	Yes	3.0.0	https://github.com/nasa/SC
FSW Make Files	Yes	3.0.0	https://github.com/nasa/SC

3.0 INSTALLATION PROCEDURES

In order to build and install the SC 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 sc)
SET(TGT1_FILELIST cfe_es_startup.scr)
```

After SC 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 SC GitHub repository (<https://github.com/nasa/SC>) under the tag “3.0.0”.

Verification of the version can be done by sending an SC NOOP command that 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
POSIX.....	Portable Operating System Interface
RTOS	Real-Time Operating System
SC	Stored Commands
SMP	Symmetric Multiprocessing
T&C.....	Telemetry and Command
TBD.....	To Be Determined
URL.....	Universal Resource Locator
VDD	Version Description Document