



NASA GSFC FLIGHT SOFTWARE SYSTEMS BRANCH

FSW VERSION DESCRIPTION DOCUMENT

CFS DATA STORAGE (DS) APPLICATION

BUILD: DS 2.6.0

RELEASE DATE: 8/30/2021

1.0 FSW VERSION DESCRIPTION

1.1 PURPOSE AND SUMMARY

This build is a minor build of the Data Storage (DS) application to resolve bugs with the DS application and ensure compatibility with cFS Caelum. This build does not include any new functionality.

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-1087	Migrate DS unit tests to distributed UT Assert	
GSFCCFS-1179	DS has static code analysis findings	In analysis on 7/10/2020, CodeSonar flagged the attached findings.
GSFCCFS-1256	Update DS to use new cFE Message Module	
GSFCCFS-1394	DS default pipe depth is too high	<p>The DS default pipe depth (256) is higher than the default max OS queue depth, which results in a run time error:</p> <p>EVS Port1 66/1/CFE_SB 4: CreatePipeErr:OS_QueueCreate returned -11,app DS</p> <p>EVS Port1 66/1/DS 2: Unable to create input pipe, err = 0xCA000005</p> <p>EVS Port1 66/1/DS 3: Application terminating, err = 0xCA000005</p> <p>1980-012-14:03:20.55729 DS application terminating, err = 0xCA000005</p> <p>Should avoid a run time error out-of-the-box.</p>
GSFCCFS-1396	DS can't load default file table	<p>The DS default file table doesn't validate on startup.</p> <p>EVS Port1 66/1/DS 11: Destination file table verify err: index = 0, invalid extension text</p> <p>EVS Port1 66/1/DS 10: Destination file table verify results: desc text = OK, good entries = 0, bad = 6, unused = 10</p> <p>EVS Port1 66/1/CFE_TBL 96: DS: Validation func reports table invalid (Stat=4294967295) for 'DS.FILE_TBL'</p> <p>EVS Port1 66/1/DS 8: Unable to load default Destination File Table: Filename = '/cf/ds_file_tbl.tbl', Error = 0xFFFFFFFF</p>

GSFCCFS-1401	DS 8000 does not match the code	<p>Requirement DS8000 states that the following is included in the DS Housekeeping packet:</p> <p>p) Destination file(s) state:</p> <ol style="list-style-type: none"> 1) File age 2) File size 3) File rate 4) Sequence count 5) Enable state 6) Open state 7) Filename <p>In reality, this is captured in the DS File Info packet. There are currently no requirements related to the File Info packet.</p>
GSFCCFS-1431	Add DS Requirement for Get File Info Command	<p>Adds a requirement for the DS Get File Info Command.</p> <p>Proposed requirement text:</p> <p>DS5017: Upon Receipt of a Get File Info command, DS shall generate a telemetry packet containing the following information for each destination file:</p> <ol style="list-style-type: none"> 1) File age 2) File size 3) File rate 4) Sequence count 5) Enable state 6) Open state 7) Filename
GSFCCFS-1437	DS "No Header" option is not working	RST Systems is strongly considering Data Storage (DS) files with no header. This option is advertised to be available by setting DS_FILE_HEADER_TYPE to 0 (instead of 1 for the standard cFE/DS headers), but it appears this functionality was removed when removing the GPM related header. Can we have this feature restored?
GSFCCFS-1475	DS does not build with eval-cert3	DS needs updates in order to build with eval-cert3 (Caelum preview)
GSFCCFS-1578	DS doxygen config file should be renamed for clarity	The filename "ds_config.txt" suggests that this a configuration file for the app itself as opposed to a configuration file for doxygen.
GSFCCFS-1587	DS should use const for function arguments where possible	
GSFCCFS-1620	DS event messages do not allow for extended message	Events that print out a messageID value use the 0x04X format specifier, which does not work for longer message IDs.

	IDs	
GSFCCFS-1643	DS_SET_FILTER_FILE_CC fails with extended message IDs	DS_SET_FILTER_FILE_CC – Message ID is getting truncated to 16 bits such that the table lookup fails for a valid message ID.
GSFCCFS-1709	DS should not pend forever on the software bus	
GSFCCFS-1714	DS should use strict resource IDs	

No new functionality was added in this build.

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.

Table 1.3-1 – Currently open DCRs

Key	Summary	Description
GSFCCFS-1173	DS could support a maximum number of files	The DS file table could be modified to (optionally) enforce a maximum number of files for each type of file. This could be useful for missions with memory constraints.
GSFCCFS-1018	DS should use CFE_FS_InitHeader	DS initializes and populates the CFE FS header itself (see ds_file.c, lines 273-278) This requires DS to understand the details of the header structure and could break if that structure changes. It should instead use the CFE_FS_InitHeader function which is designed to do exactly this.
GSFCCFS-766	ds should (optionally) add a timestamp for each packet stored	CCSDS telemetry packets include a timestamp in the CCSDS headers. Command packets, on the other hand, do not. Also, if CCSDS timestamps are generated by something other than the local CPU, the timestamp may reflect when the packet was generated but not when the packet was received/stored by DS. Thirdly, if the CCSDS timestamp is generated using a different clock that is not in sync, the timestamps may not coincide. This is particularly important in multi-CPU

		<p>environments, such as when cFS busses are connected via SBN.</p> <p>This will particularly help with replay using the ds_replay application as the timestamps will accurately reflect when DS received the packets and will be in the correct order.</p> <p>I suggest adding, for each packet stored in DS, a DS packet header containing a timestamp. This header could also include sequence count, message length (although easy to compute using the CCSDS header, a DS-generated length would make for easier access), byte position in file, or other fields.</p> <p>Of course, all of this adds to the amount of data stored in DS files, so all should be optional. The DS file header should include the necessary metadata to determine what the DS packet header will contain.</p>
GSFCCFS-759	DS file header should include additional metadata	<p>DS currently stores a number of fields in the DS file header (DS_FileHeader_t), namely the time the file was closed, the file name, the file table index, and the file name type.</p> <p>When reading DS-created files on other platforms with other configurations, it is possible to tease apart platform/mission-specific information but it would be easier to store the configuration in the header for easier analysis of DS files.</p> <p>Additional information should include, at least, the CCSDS time format (CFE_SB_PACKET_TIME_FORMAT), the CCSDS secondary header (CC, timestamp) endian-ness, byte alignment, even the DS file header endian-ness for the close timestamp.</p> <p>Also, as this will break compatibility with any DS-generated files from previous versions, including a DS header format version # should be included.</p>
GSFCCFS-752	DS should add a per-message and/or per-file checksum	<p>Files generated by DS on a spacecraft may be subject to SEU. As such, DS should add checksums to messages stored in the DS file, and/or store a checksum for the entire DS file at close time.</p>

GSFCCFS-737	DS file header values should be big-endian	As with CCSDS, which is standardizing on big-endian for message headers, the fields in the DS file header (close time, FileTableIndex, FileNameType) should be stored in big-endian order. (Should the same be true for the CFS file header?)
GSFCCFS-733	Incorporate ds_replay app	<p>The "ds_replay" application provides the ability to read flat files generated by the Data Store ("ds") application, re-injecting CCSDS packets onto the software bus at realtime, accelerated/slowed, or hz rates. ds_replay utilizes the CFE Task framework for simultaneous playback of multiple sources and multiple files can be queued for playback.</p> <p>ds_replay is configured and controlled via command messages at runtime.</p>

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	2.6.0	https://github.com/nasa/ds
Doxygen Documentation	Yes	N/A	https://github.com/nasa/ds
Unit Test Data	Yes	2.6.0	https://github.com/nasa/ds
FSW Make Files	Yes	2.6.0	https://github.com/nasa/ds

3.0 INSTALLATION PROCEDURES

In order to build and install the DS 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 ds)  
SET(TGT1_FILELIST cfe_es_startup.scr)
```

After DS 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 DS GitHub repository (<https://github.com/nasa/DS>) under the tag “2.6.0”.

Verification of the version can be done by sending a DS 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
DS	Data Storage
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
SMP	Symmetric Multiprocessing
T&C.....	Telemetry and Command
TBD.....	To Be Determined
URL.....	Universal Resource Locator
VDD	Version Description Document