Code 582
Flight Software Systems Branch

Mission Name

Core Flight System (CFS)
Data Storage (DS)
Version 2.4.1.0
APPLICATION USER'S GUIDE

Goddard Space Flight Center

Greenbelt, Maryland

National Aeronautics and
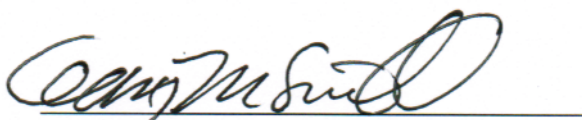Space Administration

FORWORD

This Core Flight System (CFS) Data Storage (DS) Application User's Guide provides guidance for the Flight Operations Team (FOT) for the CFS DS Application.

This is one of a set of enhanced User's Guides for the CFS Product Documentation Suite. While the main audience is the FOT, the Guides also help serve the needs of flight software developers; Flight Software Sustaining Engineering (FSSE), Integration and Test (I&T), and others who support missions which use CFS.

This generic CFS DS Guide is set up so that missions can convert it into a mission-specific guide.

The signatures on this page apply to the document as distributed before mission tailoring.
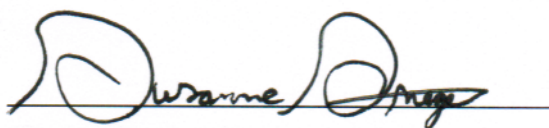
AUTHOR

Gary M Smith

Technical Writer

6/24/13

Date

APPROVALS

Susanne Strege / 582

6/24/13

Date

Core Flight Executive (cFE) Core Flight System (CFS) / Product Development Lead (PDL)

Charles Wildermann / 582

6-24-2013

Date

Flight Software Systems Branch / Head

UPDATE HISTORY

Version	Date	Description	Affected Pages
Draft 0.1	04/10/13	1st Draft	All
1.0	6/24/13	First Release	All
1.1	01/20/15	Added failed table load information to sections 3.1 and 3.2	3-1
1.2	07/22/15	Added configuration parameter describing use of Critical Data Store to save and restore the current storage enable/disable state.	Table of Contents, pages 3-1 and A-10

CONTENTS

The controlled copy of this document is located online at https://fsb.gsfc.nasa.gov/CFS/

TABLE OF FIGURES

TABLES

The controlled copy of this document is located online at https://fsb.gsfc.nasa.gov/CFS/

The controlled copy of this document is located online at https://fsb.gsfc.nasa.gov/CFS/

The controlled copy of this document is located online at https://fsb.gsfc.nasa.gov/CFS/

Chapter 1. Introduction to the CFS DS User's Guide

1.1 *Purpose and Scope of this Guide*

The primary purpose of this Application User's Guide is to help the Flight Operations Team (FOT) understand the DS application.

Many other purposes will be found for this Guide, including helping mission flight software personnel populate the ground system Record Definition Language (RDL) files in the ground system used later by the FOT, e.g., Advanced Spacecraft Integration & System Test software (ASIST).

Further purposes of this Guide are to help mission developers, system integration test team members, and FSSE to understand the DS application for their own specific needs, such as using the software to perform certain hardware tests.

If you want a more detailed understanding from a developer or FSSE standpoint, review separately the Doxygen-generated html documentation files, or review the source files directly.

As delivered, this is a generic document ready to insert mission defined values to serve the needs of specific missions.

1.2 *Acknowledgements*

This Application User's Guide relies heavily on the content of earlier heritage DS publications, presentations, and interviews with flight software engineers. Appendix A is based on information from CFS DS source code and reformatted for this publication. A publication like this is a team effort - thank you to the developers, the management team, and the entire Code 582 review team.

1.3 *Conventions and Terminology*

- In this document, *Data Storage Application*, *DS application*, and *DS* are used interchangeably, and refer to the CFS version unless specifically referring to heritage code.

- In this document, *flight controller*, *ground controller*, and *FOT* are used interchangeably.

- In this document, the percent sign (%) when followed by a string may indicate variable text. See Appendix A for references to the text that may be substituted in each case.

- *Core Flight Executive* is abbreviated cFE (lower case "c").

- *Partition* refers to the top level directory of a storage device. A storage device may be formatted so that it contains multiple partitions. Partitions may be fixed sized file systems or simply directory names.

- *File Location* refers to the full path file specification.

- *Pathname* refers to the full path file specification.

- *Basename* refers to the portion of the DS filename that follows the pathname, and precedes the sequence number.

- *Sequence number* refers to a counter, maintained by DS, which is used to uniquely identify a file. For example, filenames that use the size schema are named using the following notation: Basename + File Sequence Counter + Extension (e.g. InstrumentX12345678.xyz).

- *Full path file specification* refers to the Pathname, plus the Basename, plus a Sequence number, and finally, an extension. DS manages the "filename" as the full path file specification.

- *See Appendix B for more.*

1.4 **Related Documents**

Documents used in the preparation of this Guide are listed in the table below.

Table 1 Related Documents

Item No.	Document ID	Document Source
1	N/A	Walling, Scott. *Core Flight System Data Storage (DS) Application Design As Built [Design Presentation].* Greenbelt, MD: Goddard Space Flight Center, Code 582 (Flight Software Branch), 22 Nov 2001. PPT.
2	N/A	Strege, Susanne. *CFS Data Storage (DS) User's Guide.* Greenbelt: NASA Goddard Space Flight Center, Code 582, CFS Product Development Team, 1 March 2013. Doxygen Output. HTML.
3	582-2008-008	*CFS DS Requirements V1_1 080111.* Greenbelt: NASA Goddard Space Flight Center, Code 582, Flight Software Systems Branch, 1 August 2011. PDF.
4	464-FSW-HDBK-0138	Steven Slegel. *Solar Dynamics Observatory (SDO) Telemetry Output (TO) Users Guide Version 2.0.* Greenbelt: NASA Goddard Space Flight Center, SDO Flight Software Product Development, 18 April 2007. DOC.

1.5 **Assumptions**

1.5.1 Personnel

This Application User's Guide assumes the primary reader is a member of the FOT or the equivalent.

1.5.2 CFS DS Application

The following list summarizes the assumptions made about the CFS DS Application as documented in this Guide:

- The CFS DS code has not been modified.

- The CFS DS Application has been configured using the standard CFS DS configuration parameters.

- The cFE Application Programming Interface (API), and Operating System Abstraction Layer (OSAL) are being used.

- Consultative Committee for Space Data Systems (CCSDS) File Delivery Protocol (CFDP) is being used.

- DS assumes that a file system is being used. In order to get the data to the ground, a file transfer mechanism such as CFDP must be used; DS is not responsible for transferring the data.

- DS does not provide the interface to the hardware used for storing the data. DS simply writes to files and assumes that the interfaces required to write to the devices/partitions are available.

- Use of Electrically Erasable Programmable Read Only Memory (EEPROM) generally requires that writing be enabled prior to writing. DS does not provide this capability.

- A command is sent to DS to request DS housekeeping messages (i.e. from the CFS Scheduling [SCH] application).

- Until a disk is created and partitions formatted for DS to use to store the destination file, the DS application should be disabled. Once the disk has been created, and partitions formatted, the Enable command can be sent to enable the DS application to start opening and writing to files, as defined in the Destination File Table.

- The command mnemonics shown in this Guide were developed for, and assume the use of, ASIST ground control software.

1.6 *How to Use this Document*

Use this Application User's Guide as a learning tool before launch. Use the Appendix as a reference tool after launch. Read it offline as a paper printout, or search the electronic file. Use the Guide along with Doxygen generated .html files for more complete understanding. In contrast to this Guide, the Doxygen generated html files are primarily targeted to developers and FSSE.

Experienced flight controllers may only need to browse the Guide, and use the Appendix as needed. New flight controllers may wish to get more familiar with the entire Guide well before needing to rehearse operational scenarios or set up flight software procedures ("procs").

1.6.1 Providing Feedback

For CM reasons, if you find any item in this Guide that is in error, or want to be informed of any updates, please email feedback to the cFE/CFS PDL for validation and routing. As of the date of publication, the cFE/CFS PDL is Susie Strege (susanne.L.strege@nasa.gov).

Besides corrections of any errors, the CFS team is interested in your ideas on improvements that help understanding.

1.7 *Acronyms and Abbreviations*

Acronyms and abbreviations in this publication are shown in Table 2 below. Telemetry, command mnemonics, and similar terms are not shown here.

Table 2 Acronyms and Abbreviations

Abbreviation or Acronym	Description
ACS	Attitude Control System
API	Application Programming Interface
ApID	Application ID, also known as Message ID
ASIST	Advanced Spacecraft Integration & System Test software
C&DH	Command and Data Handling
CCSDS	Consultative Committee for Space Data Systems
CC	Command Code
CDS	Critical Data Store
CFDP	CCSDS File Delivery Protocol
cFE/CFE	Core Flight Executive
CFS	Core Flight System
CI	Command Ingest
Cmd	Command
DS	CFS Data Storage application
EEPROM	Electrically Erasable Programmable Read Only Memory
FOT	Flight Operations Team
FSSE	Flight Software Sustaining Engineering
GN&C	Guidance Navigation and Control
GPM	Global Precipitation Measurement
GSFC	Goddard Space Flight Center
HK	Housekeeping Application
I&T	Integration and Test
ITOS	Integration Test and Operations System
LRO	Lunar Reconnaissance Orbiter
LSB	Least Significant Bit
MID	Message ID

The controlled copy of this document is located online at https://fsb.gsfc.nasa.gov/CFS/

Abbreviation or Acronym	Description
MSB	Most Significant Bit
No-op	No operation
OS	Operating System
OSAL	Operating System Abstraction Layer
PDL	Product Development Lead
RAM	Random-Access Memory
RDL	Record Definition Language
SB	cFE Software Bus Application
SC	Stored Command application
SCH	CFS Scheduling Application
SDO	Solar Dynamics Observatory
ST-5	Space Technology 5
TBL	cFE Table Services Application
TIME	cFE Time Application
TLM	Telemetry
TO	Telemetry output
VCDU	Virtual Channel Data Unit

This page intentionally blank.

Chapter 2.   Introduction to the CFS DS Application

This chapter addresses the theory and internal operation of the CFS DS application.

2.1 *Heritage*

The CFS DS code derives from the Lunar Reconnaissance Orbiter (LRO) mission, which used data storage files. Most other heritage missions, such as the Solar Dynamics Observatory (SDO) and Space Technology 5 (ST-5), stored the data as messages or Virtual Channel Data Units (VCDUs) in buffers. In the CFS DS design, data is stored as cFE Software Bus messages in files, rather than buffers.

The CFS team took LRO's DS application and made it CFS compliant, adding configuration parameters in place of hard coded parameters, and making other changes to fit it into the larger CFS architecture.

2.2 *CFS DS Overview*

The CFS DS application performs three basic functions:

- It writes table-defined data packets (cFE Software Bus messages), into files for later transmission to the ground.

- It performs filtering on received packets, prior to storage.

- It stores files on a file system.

DS references two tables, the Packet Filter Table and the Destination File Table.

2.2.1 Writing Table-Defined Data Packets into Files

DS writes cFE Software Bus messages to table-defined data storage files. These files are generally stored on a device such as a solid state recorder. However, they could be stored on any storage device with a file system.

DS has no file download or playback capabilities; another application such as CFDP must be used to transfer to the ground the files created by DS.

2.2.2 Inputs to DS

Any cFE Software Bus message (a CCSDS compliant packet) can be written to files by DS.

- The cFE Software Bus (SB) application routes cFE Software Bus messages to DS. The messages can come from a number of sources; these sources can include ground commands via a command ingest application.

- The cFE Table Services application provides DS with knowledge of updates to the DS Destination File Table and the DS Packet Filter Table.

DS keeps a list of Message IDs (also known as Application ID, or *ApID* entries) to which it has subscribed. For fast look up of packet ApIDs, DS creates a hash table.

For the Packet Filter Table (though not the Destination File Table) when DS receives notice from cFE Table Services that a new table commit is pending, DS unsubscribes to the ApID entries in the old Packet Filter Table before the new table data is copied into DS table space. Typically the table service command sequence is load/validate/commit. DS then subscribes to the new ApIDs and creates a new hash table.

2.2.3 Filtering Data Packets

DS can store cFE Software Bus messages at different rates using two possible filtering schemes (Sequence or Time Based filtering) allowing "N of X" messages to be stored starting at offset 0 (zero).

2.2.3.1 Packet Filter Table

Each Packet Filter Table is identified by a character string descriptor whose length is configurable by the mission. Typically, a mission may set up multiple versions of the Packet Filter Table, such as for I&T; launch; orbit insertion; Guidance and Navigation; Attitude Control System (ACS); and nominal, for example.

The Packet Filter Table contains a *list of packets* (Message IDs) that DS will process, along with *filtering information* specific to each packet.

For each Message ID, the Packet Filter Table specifies the *storage and filtering information* shown in Figure 1 below.

Storage Element

- A zero based index into the Destination File Table

Filtering Elements

- The type of filter used by this entry: by count or by time (the filter type defines whether the input data is a value derived from the packet timestamp, or if the input data is derived from the header sequence count)
- The N and X arguments passed into the Filter algorithm that allows "N of every X" data packets to be stored
- The O argument passed into the Filter algorithm Filter that allows data packets to be stored starting with the packet at offset "O"

Figure 1 Description of Entry in the Packet Filter Table

The configuration parameter DS_FILTERS_PER_PACKET specifies the **storage and filtering information** per packet. This allows a packet to be stored in multiple destinations at different rates.

For example, each full entry would contain a Message ID, accompanied by:

[(Storage Element, Filtering Elements)[1] , (Storage Element, Filtering Elements)[2] , … (Storage Element, Filtering Elements)[A]] where A is the number specified by the configuration parameter.

The **filtering elements** include three (3) 16-bit parameters (N, X, and O). The filter algorithm allows "N of X" cFE Software Bus messages to be stored starting at offset O.

2.2.3.2 Filtering by Sequence

Filtering by sequence uses the packet sequence number to determine if the cFE Software Bus message should be filtered or not.

The filter factor has three parameters and can be expressed in terms of N number of packets out of X number of packets. The offset Y determines the starting point within X for selecting packets.

If X is greater than zero, the message will be stored if and only if the sequence number modulo X is less than N. For example, N=4, X=6 and O=2 will store 4 of 6 messages starting at offset 2.
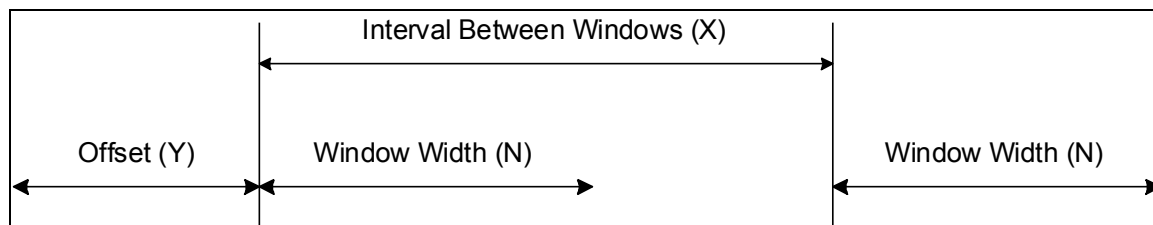
Figure 2 Filter Factor Parameters - Filtering by Sequence

If the packet sequence number minus Y mod X is less than N then the packet will be output:

```
if (((packet_sequence_counter - Y) MOD (X)) < N) then

      output the packet

else

      drop the packet

end if
```

2.2.3.2.1 *Filtering by Sequence, Example 1*

In Filtering by Sequence - Example 1, we send two out of every four packets, offset packet zero. This will send two packets and then skip two packets. If instead one wanted every other packet then one would specify one out of two packets.

Table 3 Filtering by Sequence – Example 1

N	2
Y	0
X	4

Sequence Number	Send
0	✓
1	✓
2	
3	
4	✓
5	✓

Sequence Number	Send
6	
7	
8	✓
9	✓
10	

2.2.3.2.2 *Filtering by Sequence, Example 2*

In Filtering by Sequence - Example 2 we send two out of every four packets, offset packet one. This will send two packets and then skip two packets, starting with packet one:

Table 4 Filtering by Sequence – Example 2

N	2
Y	1
X	4

Sequence Number	Send
0	
1	✓
2	✓
3	
4	
5	✓
6	✓
7	
8	
9	✓
10	✓

2.2.3.3 Filtering by Spacecraft Time

Time based filtering uses the Spacecraft time code found in the CCSDS packet secondary header. The filter factor has three parameters and can be expressed in terms of N number of packets out of X number of Seconds/Sub Seconds. Note there are 16 Sub Seconds per Second.

The offset Y determines the starting point within the X for selecting packets. Messages are stored if and only if the message time stamp modulo X is less than N.

Fifteen bits out of the 64-bit CCSDS time code field are used. The constructed 15-bit time stamp used for DS filtering consists of the 11 least significant bits (LSBs) of the seconds field and the four most significant bits (MSBs) of the sub-seconds field. Messages will be stored if and only if the constructed 15-bit time stamp modulo X is less than N.

This type of filtering is rare in that the N value must be set to the desired 15-bit time stamp, which is hard to specify.
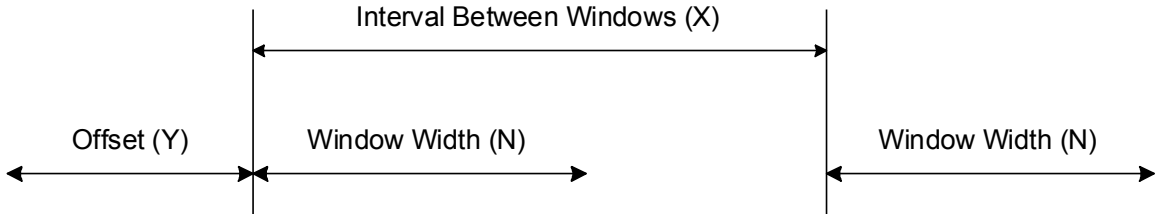
Figure 3 Filter Factor Parameters - Filtering by Spacecraft Time

The time code filter factor consists of the least significant 11 bits of the Lo_Word Seconds (a), and the most significant four bits of the Hi_Word Sub Seconds (b). Bit 16 is not used (c). For more, see tables below.

Table 5 Filtering by Time – Time Code Format from Packet

Hi_Word Seconds	Lo_Word Seconds	Hi_Word Sub Seconds	Lo_Word Sub Seconds
	X**xxxxaaaaaaaaaaa**	**bbbb**xxxxxxxxxxxx	

Table 6 Filtering by Time – Time Code Filter Format

c	aaaaaaaaaaa	Bbbb

The sign of X determines the type of filtering that will be performed:

```
if (X = 0) then

     don't output any packets

else if (X > 0) then

     sequence number filtering

else if (X < 0) then

     time based filtering

end if
```

If the packet time code (in time code filter format) minus Y mod X (the absolute value in time code filter format) is less than N then the packet will be output.

```
if (((packet_time_code - Y) MOD (ABS(X))) < N) then

     output the packet

else

     drop the packet

end if
```

2.2.3.3.1 *Filtering by Time, Example 1*

In Filtering by Time - Example 1, if a packet is generated four times a second, we send one packet per second. An offset of zero (0) will send the first packet generated per second. This works well for synchronous data that is generated at exactly the same time each second.

Table 7 Filtering by Time, Example 1

N	1
Y	0
X	16

Time (Sub Seconds)	Pkt Gen	Send
0	Pkt	✓
1		
2		
3		
4	Pkt	
5		
6		
7		
8	Pkt	
9		
10		
11		
12	Pkt	
13		
14		
15		
16	Pkt	✓

2.2.3.3.2 *Filtering by Time, Example 2*

In Filtering by Time - Example 2, no packets would be sent because no packets are generated in time slot zero (0). Note that the offset must be used to synchronize the time slot that we want to send with the time slot in which the packet is generated. The offset must be set to one (1) to make the filtering work the same as in the previous example.

Table 8 Filtering by Time, Example 2

N	1
Y	0
X	16

Time (Sub Seconds)	Pkt Gen	Send
0		
1	Pkt	
2		
3		
4		
5	Pkt	
6		
7		

Time (Sub Seconds)	Pkt Gen	Send
8		
9	Pkt	
10		
11		
12		
13	Pkt	
14		
15		
16		
17	Pkt	

2.2.3.3.3 *Filtering by Time, Example 3*

In Filtering by Time - Example 3, if a packet is generated four times a second, we send one packet per second. An offset of four (4) will send the second packet generated per second. This works well for synchronous data that is generated at exactly the same time each second. Note that an offset of one, two, or three (1, 2, or 3) will not send any packets because packets are not generated in those time slots.

Table 9 Filtering by Time, Example 3

N	1
Y	4
X	16

Time (Sub Seconds)	Pkt Gen	Send
0	Pkt	
1		
2		
3		
4	Pkt	X
5		
6		
7		
8	Pkt	
9		
10		
11		
12	Pkt	
13		
14		
15		

2.2.3.3.4 *Filtering by Time, Example 4*

In Filtering by Time - Example 4, if a packet is generated 10 times a second, we send two packets per second. Note that the X is 1/2 a second or eight sub seconds.

Table 10 Filtering by Time, Example 4

N	1
Y	0
X	8

Time (Sub Seconds)	Pkt Gen	Send
0	Pkt	X
1		
2	Pkt	
3		
4	Pkt	
5	Pkt	
6		
7	Pkt	
8	Pkt	X
9		
10	Pkt	
11		
12	Pkt	
13	Pkt	
14		
15	Pkt	

2.2.3.3.5 *Filtering by Time, Example 5*

In Filtering by Time - Example 5, if a packet is generated four times a second, we send two packets per second. Note that in this example packets are not generated at exactly the same times each second and we will not see every packet we expect.

Table 11 Filtering by Time, Example 5

N	1
Y	0
X	8

Time (Sub Seconds)	Pkt Gen	Send
0		
1	Pkt	
2		
3		
4	Pkt	
5		
6		
7		
8	Pkt	X
9		
10		
11		
12		
13	Pkt	

14		
15		

2.2.3.3.6 *Filtering by Time, Example 6*

To fix the problem in example 5, and to get every other packet, set the N to 3. This will send the packet as long as it falls within 3 slots (3 packets every 8 sub seconds).

Table 12 Filtering by Time, Example 6

N	3
Y	0
X	8

Time (Sub Seconds)	Pkt Gen	Send
0		
1	Pkt	X
2		X
3		
4	Pkt	
5		
6		
7		
8	Pkt	X
9		
10		
11		
12		
13	Pkt	
14		
15		

2.2.4 Writing Destination Files

The Destination File Table defines a list of data storage files along with creation information specific to each file. A configuration parameter determines the maximum number of files that can be defined in the Destination File Table.

The Destination File Table defines data storage files as shown below:

 Filename

- The file path
- The base portion of the filename
- The filename extension (including the dot)
- The file type: by count or by time (the file type denotes whether the file name includes an incrementing value or the Spacecraft time)

 State

- The enable/disable state for this destination file

 Limits

- The file **size** at which to close the file and open a new one
- The file **age** at which to close the file and open a new one

Sequence

- An initial value for a numerical sequence to be appended to the filename before the extension

As each file is closed, if the file type is by count, then the file type counter is incremented for the next file to be opened. (If the file type is by time, then the current Spacecraft time is always used, and Spacecraft time is always changing, resulting in a unique filename for each file.)

Each DS filename contains characters derived from a filename sequence counter or from the current Spacecraft time at the time the file was created.

Files are generally organized in a way that is meaningful to each mission. For example, some missions may wish to include all housekeeping data into one set of files, while others may wish to split out certain types of housekeeping data into separate files. The Destination File Table contains the scheme, or definitions, for the files into which cFE Software Bus messages will be saved.

Any packet sent on the Software Bus can be stored in a file. Typically Command and Data Handling (C&DH) housekeeping telemetry is saved into one file, for example, and Guidance, Navigation and Control (GN&C) housekeeping in another file. Instrument data may be saved in yet another file.

Science data, if it is low volume, may also be saved, though typically science data is larger volume and handled by a separate, dedicated module.

The Destination File Table is typically set up before launch and is not frequently changed.

2.2.4.1 More about DS Filenames

DS manages the filename as the Pathname, plus the Basename, plus a Sequence number, and finally, an extension. The entire length of the filename must not exceed the length specified by the OSAL configuration parameter OS_MAX_PATH_LEN.

Each entry in the Destination File Table contains a character string definition for Pathname, Basename and Extension and a type selection for the Sequence portion of the filename.

Depending on the Filename Type, DS will construct a character string that represents either the value of a filename sequence counter specific to that particular Destination File Table entry, or the current time of day in the format YYYYDDDHHMMSS.

If the file is sequence-based, the sequence characters will be based upon the next sequence number. For example, if the initial value for a numerical sequence = 1, Pathname = /ram and Basename = seq and extension = .dat, then the file would be named /ram/seq00000100.dat for the 100th file.

- For the maximum value a file sequence count will reach before rollover to zero, refer to the mission-specific DS config. parameter DS_MAX_SEQUENCE_COUNT.

- For the number of filename sequence count digits used when the filename type has been set to "count" rather than "time," refer to the mission-specific DS configuration parameter DS_SEQUENCE_DIGITS.

Tip: *As a reminder, the only portion of **sequence-based** filenames that change is the sequence counter, as assigned to the file at the time the file is opened. The path, base, and extension are*

constant as defined in the Destination File Table, unless changed by command and/or a table load.

Tip: Note that in sequence-based filenames the sequence count always rolls over to zero. The rollover to zero is not configurable to a different number. In practice, missions often reset the sequence count each orbit, or whenever the mission changes the base part of the filenames to identify a new data set. So the result is that in practice the numbers seldom get large enough to roll over.

If the file is **time-based**, the characters will be in the form YYYYDDDHHMMSS. For example if the file was created on day 123 at 14:15:22 using Pathname = /ram, Basename = tim and extension = .dat, the file would be named /ram/tim2010123141522.dat.

*Tip: As a reminder, the only portion of **time-based** filenames that change is the Spacecraft time, as assigned to the file at the time the file is opened. The path, base, and extension are constant as defined in the Destination File Table, unless changed by command and/or a table load.*

For the path, base and extension, refer to the mission-specific File Table for the appropriate entry.

2.2.4.2 Primary File Header

Each DS destination file contains a primary cFE file header with the information shown below.

Table 13 Primary File Header Contents

Element	Description
Content Type	Specifies cFE compliant files. Value will be set to 0x63464531
SubType	Specifies type of file. Value is defined via DS configuration parameter
Primary header length	Length of primary header
Spacecraft ID	Spacecraft that generated the file
Processor ID	Processor that generated the file
Application ID	Application that generated the file
File creation time	In seconds
File creation time	In sub-seconds

2.2.4.3 Secondary File Header

Each DS destination file contains a mission-optional secondary file header. Each mission has the option (controlled by configuration parameter) to choose not to include a secondary file header.

The format of the secondary file header is also configurable. This version of the DS application includes the ability to generate the CFS DS standard file header or the file header that conforms to the specifications of the Global Precipitation Measurement (GPM) mission. The tables below shows the contents of the DS standard file header and GPM file header.

Table 14 Secondary File Header Contents

Element	Description
File close time	Seconds portion of Spacecraft time that the file was closed
File close time	Subseconds portion of Spacecraft time that the file was closed
File table index	Destination File Table index
File name type	File type (count vs. time)
Qualified filename	Full path specification of file

Table 15 GPM File Header Contents

Element	Description
Source ID	String of four characters identifying the source of the file data
Sequence ID	File generation counter
File start time	Seconds portion of Spacecraft time that the file was opened
File stop time	Seconds portion of Spacecraft time that the file was closed
File size	Total size of the file in bytes
Spare	Spare bytes (20 total)

2.2.4.4 Updates by Command vs. Updates by Table

For configuration management, some missions prefer to make changes only by changing tables via a table load where a new table file can be uplinked via CFDP and loaded via cFE Table Services. However, DS supports changing individual table entry settings by command. For details, see Appendix A, section A.4, DS Commands.

2.2.4.5 Closing Files

Files can be defined to close at a maximum size (measured in bytes) and a maximum age (measured in seconds). When closed, a new file will be opened.

If the desired **maximum file size** would be exceeded, the application will close the file and open a new one with a unique name according to the rules set up in the Destination File Table. If the desired **maximum age** would be exceeded, the application will close the file and open a new one with a unique name according to the rules set up in the Destination File Table.

However, files are not closed based on whether the filename is sequence based or time based. **Both the file size and file age are checked.** If a file is time based but the file size is exceeded before the time is met, the file will still be closed.

If one wants the file to close by *age* then the file size limit needs to be set unreasonably large. The unreasonably large file size limit should never pass, making age the only meaningful check.

Similarly, if one wants the file to close by *file size* then the age limit needs to be set unreasonably large. The unreasonably large age limit should never pass, making the file size the only meaningful check.

To avoid creating empty files, the desired file size is only verified after the first packet is saved. So the first packet written may cause the file to exceed the desired size.

DS will always have at least one packet in every file. Its size is defined in the Destination File table, and can be changed by the FOT.

cFE Software Bus messages are stored in a file or files in their entirety, i.e. messages are not segmented between files.

2.3 *DS Software Context*

The figure below shows the software context for the CFS DS Application.
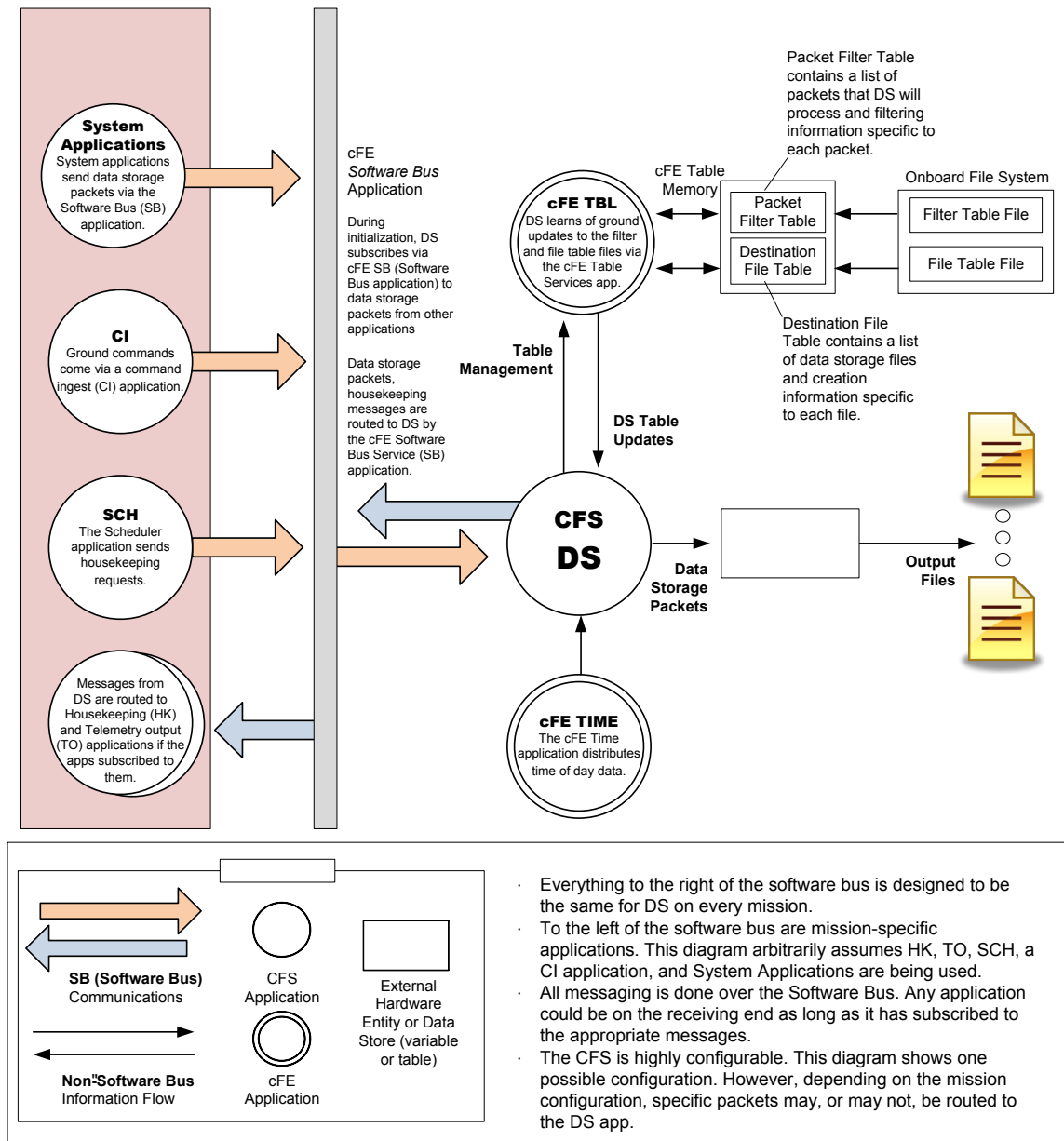
Figure 4 DS Application Software Context

2.3.1 System Applications

As shown in Figure 4 above, CFS system applications as configured for a particular mission may communicate with DS via message packets via the cFE Software Bus application.

2.3.2 CI Application

Ground commands are routed through a CI Application, as shown in Figure 4 above.

2.3.3 cFE SB Application

Message packets which will be filtered and saved by DS arrive via the cFE SB Application. As shown in Figure 4 above, CFS applications communicate with DS via SB.

2.3.4 SCH Application

The SCH application wakes up DS when SCH requests DS housekeeping data, as shown in Figure 4 above. Additionally, DS subscribes to message packets, and is awakened when it receives those packets.

2.3.5 cFE Table Services (TBL) Application

As shown in Figure 4 above, DS learns of any updates to the DS Packet Filter Table and Destination File Table through the cFE Table Services (TBL) application.

2.3.6 cFE Time (TIME)

cFE TIME (the cFE Time application) distributes Spacecraft time of day data used by DS, as shown in Figure 4 above.

2.3.7 Housekeeping (HK), Stored Command (SC)

As shown in Figure 4 above, any cFE Software Bus messages generated by DS are routed to whatever mission-specific applications subscribe to them, such as the Housekeeping (HK) application, a telemetry output (TO) application, and/or Stored Command (SC) Applications.

Figure 5 below shows the overall program flow of the DS, from start to exit.
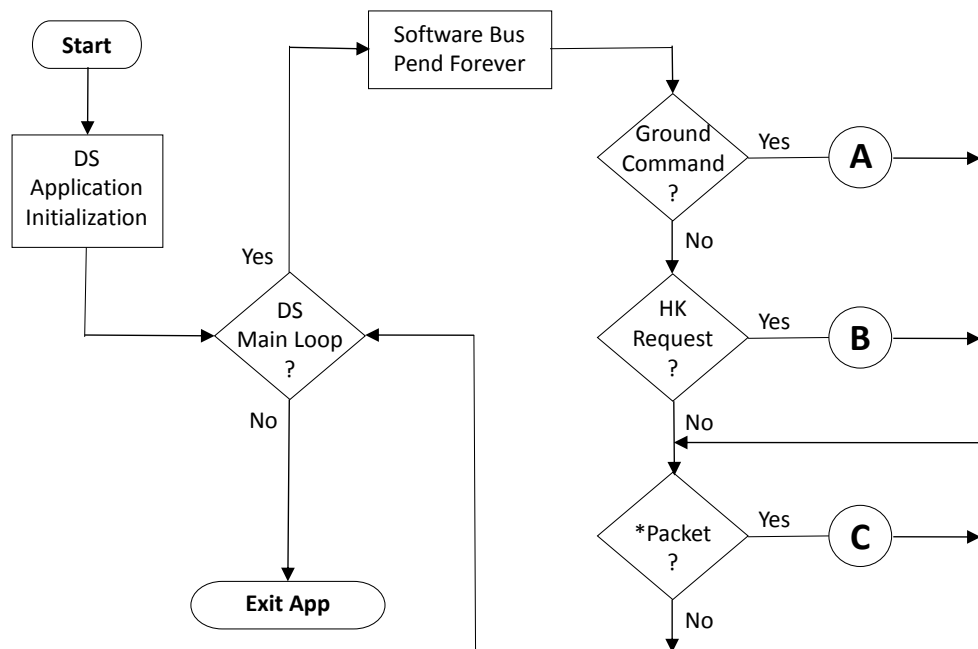
Figure 5 DS Overall Flow Control

Figure 6 below shows the flow control of DS for *ground* commands.
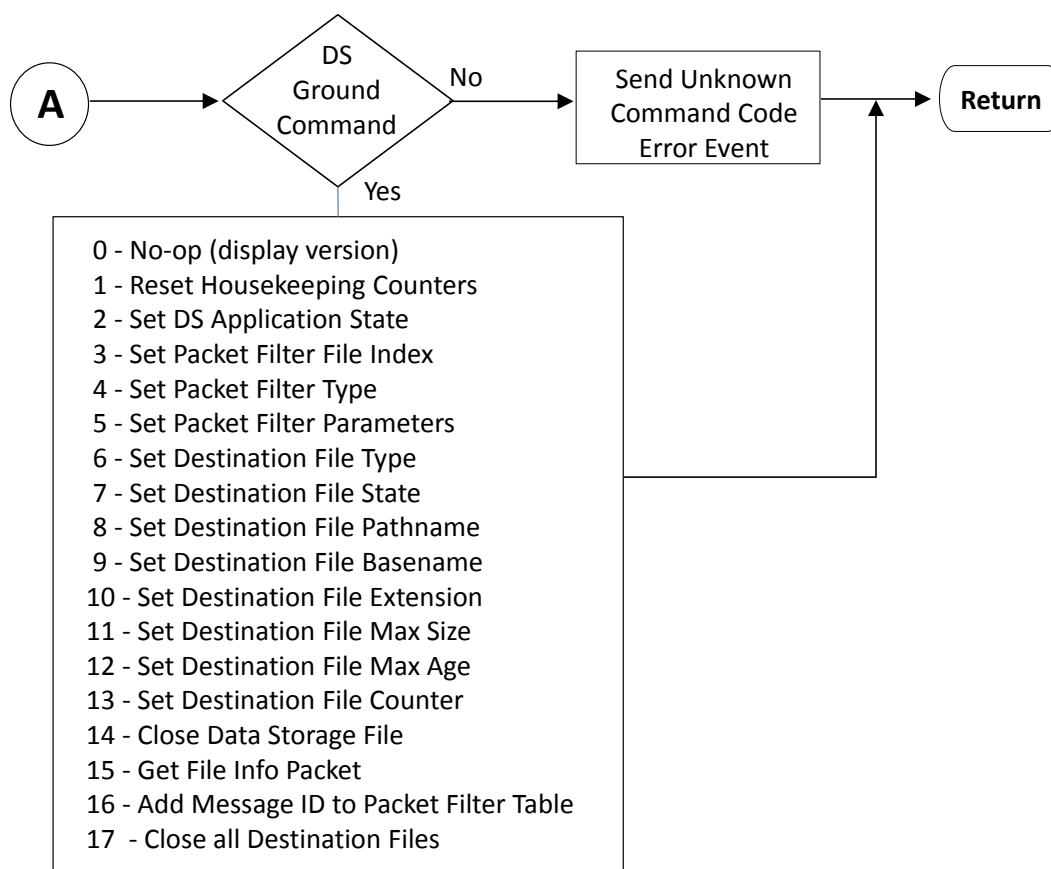
Figure 6 DS Flow Control Detail (A) - Ground Command

Figure 7 below shows the flow control of DS for *housekeeping* requests.
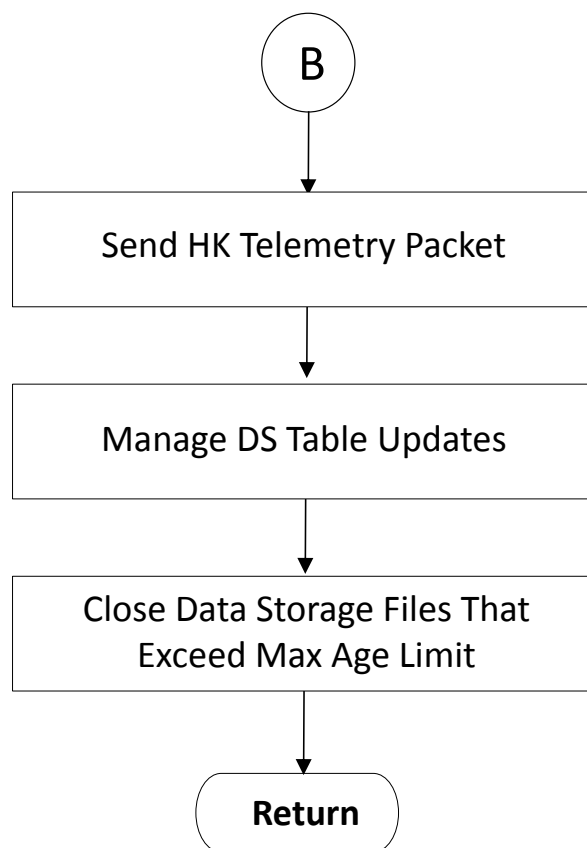
Figure 7 DS Flow Control Detail (B) - Housekeeping Request

Figure 8 below shows the flow control of DS for *Store Data*.
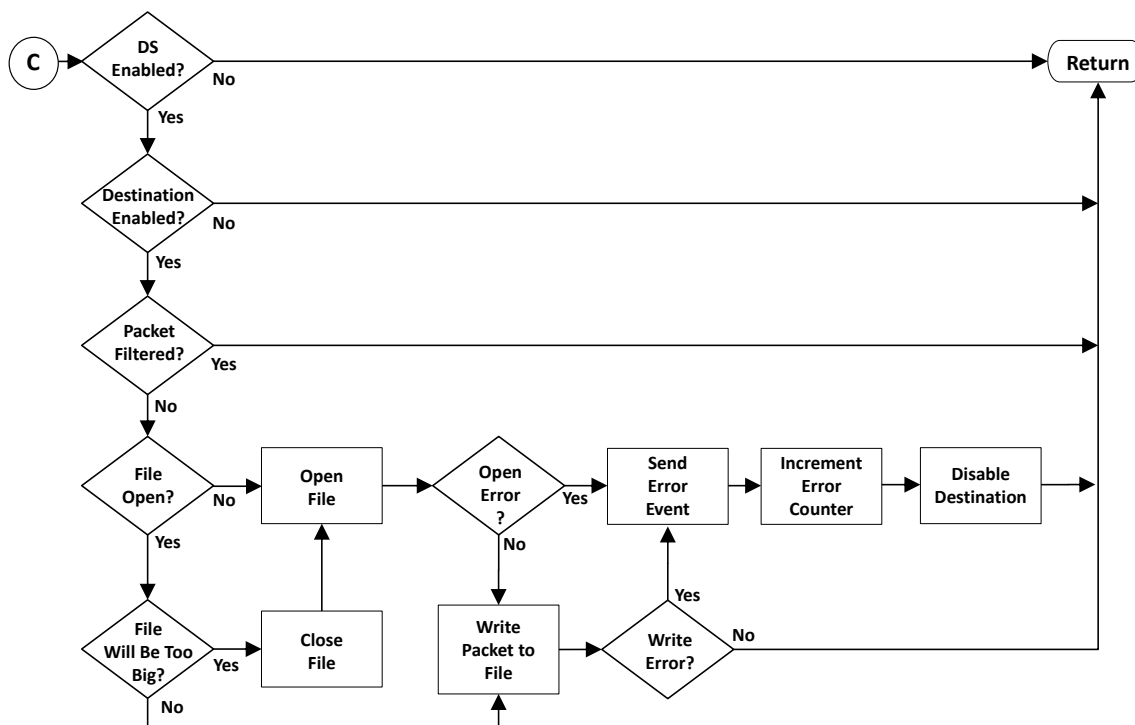
Figure 8 DS Flow Control Detail (C) - Store Data

Figure 9 below shows a visual description of how the two DS tables are used to control data storage.
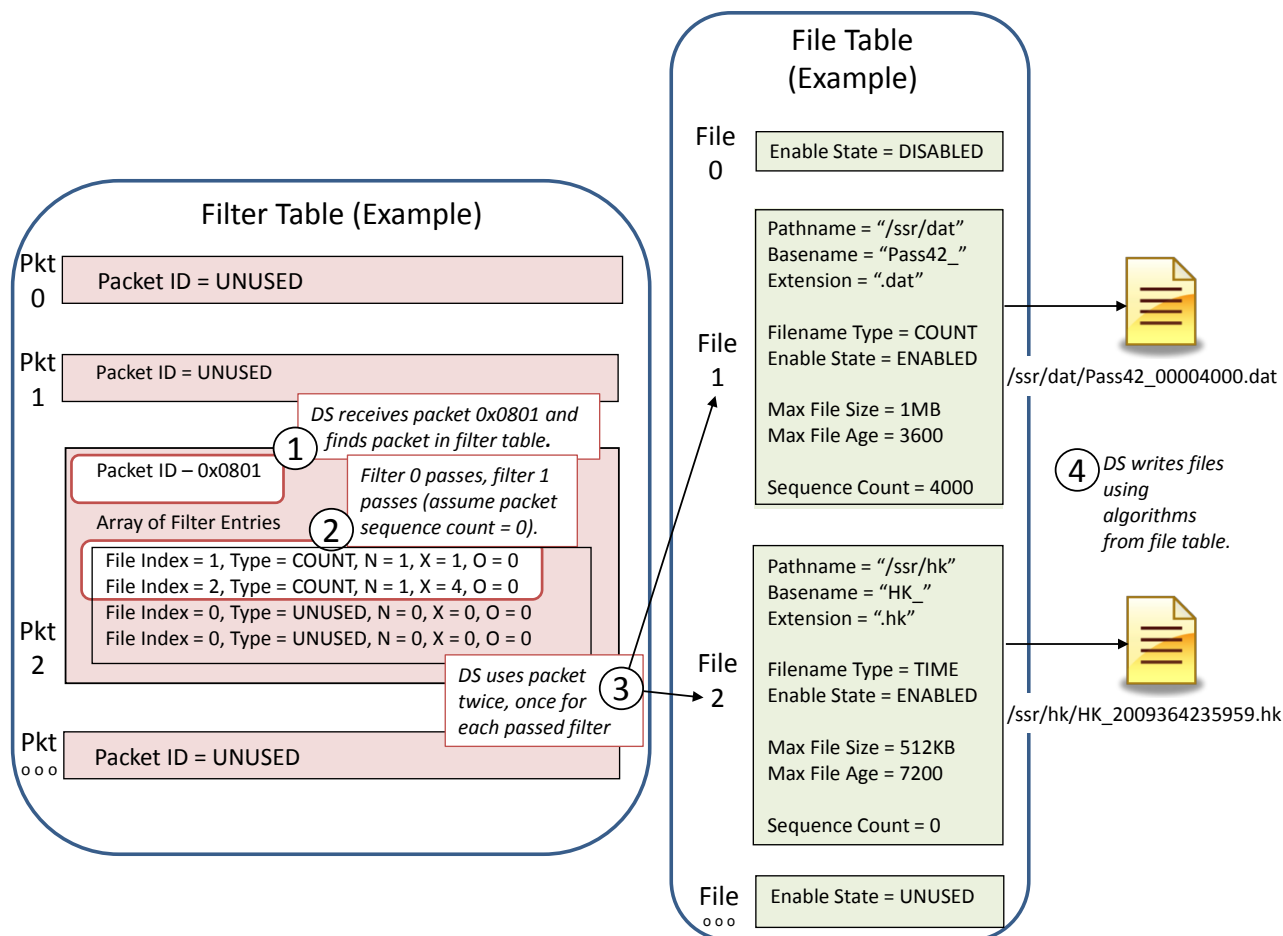
Figure 9 DS Input Packet Flow (Example)

This page intentionally blank.

Chapter 3.   **CFS DS Normal Operations**

This chapter addresses the most common issues expected to arise during normal operations.

3.1 *DS Modes of Operation*

DS can be configured to start up enabled or disabled. A configuration parameter (DS_DEF_ENABLE_STATE) defines the state in which DS will start following a power-on reset. Another configuration parameter (DS_CDS_ENABLE_STATE) allows DS to restart in the last commanded state or DS_DEF_ENABLE_STATE following a processor reset.

In **enabled mode**, DS will begin to process packets immediately on startup. Processing packets means DS is checking to see if the received CFS Software Bus Message is included in the Packet Filter Table. If the message is included in the Packet Filter Table, DS runs the filtering algorithm to determine if the packet should be written to a file or not.

Upon startup or table update, if the Destination File Table and/or Packet Filter Table have failed to load, DS will not store any messages until both tables are loaded with valid data. Loading the Tables with valid data can be accomplished by using the cFE Table Services commands after creating a Table load image file and uploading it to the spacecraft's file system. Until both tables are loaded with valid data, DS will increment the IgnoredPktCounter for each received packet that is discarded.

The flow of this process is shown starting in Figure 8, DS Flow Control Detail (C) - Store Data, and described below starting in Section 3.2, DS Operation.

In **disabled mode**, DS will ignore message packets contained in the Packet Filter Table that DS has subscribed to until receipt of a valid command to enable the DS Application. DS will not increment the IgnoredPktCounter for each received packet that is discarded when in **disabled mode**.

3.2 *DS Operation*

When DS is enabled for storage and DS tables have loaded successfully, DS works as follows:

1. When DS receives a cFE Software Bus message that passes the filtering algorithm, the message is stored in a file or files based upon the Packet Filter Table information for that Message ID. The Packet Filter Table points to the index or indexes of the file or files defined in the Destination File Table. Once a message is stored, the DS housekeeping packet provides information about the open file.

2. Each housekeeping cycle, DS evaluates open files to determine if their maximum age has been exceeded. If their maximum age has been exceeded, then DS closes the file. The age test is done at housekeeping time because an old file needs to be closed even if there are

no packets being received. Effectively, this means the age is tested about every five seconds or so.

3. The check for maximum size is done as DS receives cFE Software Bus messages. If DS and the destination file are enabled, the destination file is open, and the cFE Software Bus message passes any filtering, then the size check is performed. If the total of the destination file's current size plus the new cFE Software Bus message size exceeds the file size limit, then the file is closed. DS opens a new file, and writes the cFE Software Bus message to the new file. A minimum of one message is stored in each file.

3.2.1 Filtering

The Packet Filter Table determines what cFE Software Bus messages DS will store via the Software Bus Message IDs and the rate the incoming messages will be stored. The DS application supports two types of filtering: *sequence number based* filtering and *time based* filtering.

Each filtering type relies on three filter factors, N, X, and O where N of X messages will be stored starting at offset O.

DS may store a single cFE Software Bus message using different filter schemes.

For more detail on filtering refer to Chapter 2.

3.2.2 Determining What Message IDs Are To Be Stored

The Message IDs to be stored are specified in the Packet Filter Table. The currently loaded Packet Filter Table may be dumped via cFE Table Services to obtain a listing of the stored Message IDs. Once the table is dumped and downlinked, the table contents are typically displayed on the mission-specific Packet Filter Table display page in ASIST, Integration Test and Operations System (ITOS), or other ground flight software.

3.2.3 Determining Where Files Are Being Stored

The locations of the files are determined by both the Packet Filter Table and the Destination File Table. The Packet Filter Table, for each Message ID, indicates the index(s) into the Destination File Table, specifying the file(s) in which the message is to be stored. For details about information in the Packet Filter Table, see Section 2.2.3.1, Packet Filter Table.

The Destination File Table defines data storage files and their location. For details about how the Destination File Table defines data storage files, see Section 2.2.4, Writing Destination Files.

To trace the Message ID to the File entry, both tables must be dumped and downlinked. Once the tables are dumped and downlinked, the table contents are typically displayed on the mission-specific Packet Filter Table display page in ASIST, ITOS, or other ground flight software.

3.3 *Initializing DS*

During initialization, DS subscribes to cFE Software Bus messages from other applications as defined in the Packet Filter Table.

DS attempts to load the Destination File Table and Packet Filter Table defined in the configuration parameters DS_DEF_DEST_FILENAME and DS_DEF_FILTER_FILENAME upon startup.

If the load fails for either of these tables, DS will still start. However, DS will not store any messages until both tables are loaded with valid data. This can be accomplished by using cFE

Table Services commands after creating a table load image file and uploading it to the spacecraft's file system.

3.3.1 Power-On Reset

On power-on reset, DS performs a cFE application-specific initialization:

1. Initializes housekeeping data and global variables.

2. Registers for cFE Services.

3. Creates Software Bus input pipe.

4. Subscribes to ground commands and housekeeping requests.

5. Creates Packet Filter Table and Destination File Table.

6. Loads table data from default table files.

7. Subscribes to message packets named in the Packet Filter Table.

3.3.2 Processor Reset

On processor reset, all files that were open before processor reset will be closed, regardless of file size or age.

On processor reset, DS performs the same initialization as power-on reset unless the mission has implemented Critical Data Store (CDS), as described below.

1. If the configuration parameter DS_MAKE_TABLES_CRITICAL is set to "1", then cFE Table Services will load data for both DS tables from the CDS instead of from the default table data files.

2. Regardless of table criticality, DS will at startup attempt to retrieve the current filename sequence counters and destination enable/disable state values for each entry in the Destination File Table that is also stored in the CDS.

3.4 *Event Messages*

There are four levels of event types:

- Critical

- Error

- Information

- Debug

The levels are not used for programmatic control by CFS or DS. In other words, CFS and DS do not treat the level labels differently. However, ASIST, ITOS, or other ground flight control software may display event messages in different colors or otherwise differentiate them.

Debug level messages are typically used before launch for testing but then reconfigured off to avoid cluttering flight operator telemetry.

3.5 *Normal DS Packet Processor Operation*

DS files are not created until a packet has passed the filter tests and is ready to be written to the file. This avoids potentially empty files.

Each packet received by DS (including DS commands and the DS housekeeping packet) is processed as a candidate for being written to a file.

If DS is enabled for packet processing, and the Message ID is found in the Packet Filter Table, then the packet filters are processed by evaluating the filter algorithm using parameters specific to each packet filter.

- If any passed packet filter references an enabled entry in the Destination File Table then the packet will be written to that file. A single packet may thus be written to multiple DS files.

- If the DS file does not yet exist, it will be created and initialized with both a cFE file header and a DS secondary file header.

3.6 *Packet Filter Table Definition*

The Packet Filter Table consists of a series of packet entries. Each packet entry contains the Message IDs for DS to store.

The maximum number of entries is defined by the configuration parameter DS_PACKETS_IN_FILTER_TABLE.

The number of filters per packet entry, and thus, (1) the maximum number of destination files to which a single packet can be written at a time; and (2) the rate that a single packet can be written to those destination files, is defined by the configuration parameter DS_FILTERS_PER_PACKET.

Table 16 Packet Filter Table Elements

Element	Description
File Table Index	The zero based index into the Destination File Table. (Defines which data storage file is associated with this packet filter. It is common for multiple packet filters to reference the same data storage file.)
Filter Type	The type of filter used by this entry DS_BY_COUNT or DS_BY_TIME. (Defines the filter algorithm input data. If Filter Type is time then the input data is a 15 bit value derived from the packet timestamp. If Filter Type is count then the input data is the 14 bit packet header sequence count.)
Algorithm_N	The N argument passed into the Filter algorithm. (The filter algorithm allows "N of every X" data packets to be stored, starting with the packet at offset "O of X".)
Algorithm_X	The X argument passed into the Filter algorithm. (The filter algorithm allows "N of every X" data packets to be stored, starting with the packet

	at offset "O of X".)
Algorithm_O	The O argument passed into the Filter algorithm

3.7 *Destination File Table Definition*

The Destination File Table contains the entries described below.

Table 17 Destination File Table Contents

Element	Description
Pathname	The full path file specification. Also known as file location.
Basename	Contains the base portion of the filename.
Extension	Contains the extension (including the dot).
Filename Type	Defines the sequence portion of the filename as either an incrementing value ("count") or time of day.
File State	Defines the enable/disable state for this destination file.
Max File Size	Close current file when it gets this big.
Max File Age	Close current file when it gets this old.
Sequence Counter	Initial value for filename sequence counter. This value is referenced only when filename type is "count".

This page intentionally blank.

Chapter 4.   **Understanding DS Operational Constraints**

4.1 *File Constraints*

4.1.1 Maximum File Age

Each entry in the Destination File Table contains a maximum age value. This value represents the length of time, in seconds, that a particular data storage file may remain open.

As part of processing the housekeeping request command, DS checks the age of all open data storage files.

If the current file age exceeds the maximum age, the DS will close the file. A new file will be created only when another packet is received that passes the filter tests and targets that particular entry in the Destination File Table.

4.1.2 Maximum File Size

Each entry in the Destination File Table contains a maximum file size value. This number represents the maximum file size, in bytes, for a particular data storage file.

Prior to writing a packet to an existing data storage file, DS performs a test to see if the current file size plus the size of the packet would exceed the maximum size for that particular file.

If the resulting file size would exceed the maximum file size, then DS will close the old file and open a new file before writing the packet data.

Note that there is no size test performed on the first packet written to a new data storage file. This scheme ensures that every packet can be written to a file and that there are no empty files, but at the cost that a file with a single large packet may exceed the size limit.

This page intentionally blank

Chapter 5.   **Frequently Asked Questions (FAQs)**

5.1 *How do I downlink the files saved by DS?*

DS has nothing to do with downlinking packets and files. A file delivery application such as the CFDP (CF) application is required to downlink the files created by DS.

5.2 *Are there any issues in limiting data?*

One early task for FOT is to set up the Packet Filter Table to limit collection of data to the desired amount between downlinking times. To do this, FOT should understand the Packet Filter Table. Remember that the Destination File Table doesn't have anything to do with limiting data. The Packet Filter Table alone "limits" the data.

5.3 *How can I keep files open?*

There may arise a need to record all data when either not in contact with the ground, or when in contact, if the downlink rate is not high enough for all data. Later, at some point, it can be decided whether to downlink or delete the files.

There are no settings to keep files open indefinitely, but to effectively bypass size and age settings, these can be defined at the maximum value (4 billion hex), a value higher than either disc size or mission lifetime, respectively. The application will continue to check the values, but this setting will effectively keep the file open until manually closed by command.

This page intentionally blank

Appendix A   **CFS DS Reference**

A.1   HK Packet Telemetry

The telemetry in this section is contained in packets related to housekeeping.

Table 18 Telemetry Data - Application State

Name	AppEnableState
Description	AppEnableState denotes whether the DS application is enabled or disabled.
Data Type	Unsigned Eight Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_AppEnaState*

Table 19 Telemetry Data - Commands Accepted Counter

Name	CmdAcceptedCounter
Description	CmdAcceptedCounter contains the count of valid commands received.
Data Type	Unsigned Eight Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_CMDPC*

Table 20 Telemetry Data - Commands Rejected Counter

Name	CmdRejectedCounter
Description	CmdRejectedCounter contains the number of commands with process errors.
Data Type	Unsigned Eight Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_CMDEC*

Table 21 Telemetry Data - Destination Table Error Counter

Name	DestTblErrCounter
Description	DestTblErrCounter contains the count of the number of times that cFE Table Services attempted and failed to get Destination File Table data.
Data Type	Unsigned Eight Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_DestPtrErrCnt*

Table 22 Telemetry Data – Count of Destination File Table Loads

Name	DestTblLoadCounter
Description	DestTblLoadCounter contains the count of the number of times that cFE Table Services signaled new Destination File Table data.
Data Type	Unsigned Eight Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_DestLoadCnt*

Table 23 Telemetry Data - Count of Packets Discarded While DS Was Disabled

Name	DisabledPktCounter
Description	DisabledPktCounter contains the count of the number of packets not processed because the packet processor was disabled.
Data Type	Unsigned 32 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_DisabledPktCnt*

Table 24 Telemetry Data – Count of Good Updates to Secondary Header

Name	FileUpdateCounter
Description	FileUpdateCounter contains the count of the number of files with the secondary header successfully updated prior to being closed.

Data Type	Unsigned 16 Bit IntegerUnsigned 16 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpua_DS_FileUpdCnt*

Table 25 Telemetry Data – File Update Error Counter

Name	FileUpdateErrCounter
Description	FileUpdateErrCounter contains the count of the number of errors generated while unsuccessfully attempting to update the secondary file header.
Data Type	Unsigned 16 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DSFileUpdErrCnt*

Table 26 Telemetry Data – File Write Counter

Name	FileWriteCounter
Description	FileWriteCounter contains the count of the total number of successful file writes.
Data Type	Unsigned 16 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FileWriteCnt*

Table 27 Telemetry Data - Total Number of File Write Errors

Name	FileWriteErrCounter
Description	FileWriteErrCounter contains the count of the number of unsuccessful destination file writes.
Data Type	Unsigned 16 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FileWriteErrCnt*

Table 28 Telemetry Data - Filtered Packet Counter

Name	FilteredPktCounter

Description	FilteredPktCounter contains a count of packets discarded, because they failed all filter tests, including because of disabled destination files.
Data Type	Unsigned 32 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FilteredPktCnt*

Table 29 Telemetry Data – Packet Filter Table Error Counter

Name	FilterTblErrCounter
Description	FilterTblErrCounter contains the count of the number of times that cFE Table Services was unsuccessful in attempting to get Packet Filter Table data.
Data Type	Unsigned Eight Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FilterPtrErrCnt*

Table 30 Telemetry Data – Packet Filter Table Load Counter

Name	FilterTblLoadCounter
Description	FilterTblLoadCounter contains the count of the number of times that cFE Table Services signaled new Packet Filter Table data was available.
Data Type	Unsigned Eight Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FilterLoadCnt*

Table 31 Telemetry Data - Count of Packets Discarded - Packet Has No Filter

Name	IgnoredPktCounter
Description	IgnoredPktCounter contains the count of the number of packets not processed because tables were not loaded or because the packet was not in the Packet Filter Table.
Data Type	Unsigned 32 Bit Integer

Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_IgnoredPktCnt*

Table 32 Telemetry Data - Passed Packet Counter

Name	PassedPktCounter
Description	PassedPktCounter contains the count of the number of packets processed that passed at least one filter test.
Data Type	Unsigned 32 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_PassedPktCnt*

A.2   File Telemetry

The telemetry in this section is contained in packets related to file information.

Table 33 Telemetry Data - File Enable State

Name	EnableState
Description	EnableState contains the current state of the DS application. The possible states are enabled or disabled.
Data Type	Unsigned 16 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FileState[].EnableState*

Table 34 Telemetry Data - File Age

Name	FileAge
Description	FileAge contains the current age of the file, in seconds.
Data Type	Unsigned 32 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FileState[].FileAge*

Table 35 Telemetry Data - Filename

Name	FileName
Description	FileName contains the full filename of the file, if open. The full filename includes the Pathname, plus the Basename, plus a Sequence number, and finally, an extension. Basename refers to the portion of the Data Storage filename that follows the pathname, and precedes the sequence number. Sequence number refers to a counter maintained by DS used to uniquely identify a file.
Data Type	Char
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FileState[].FileName*

Table 36 Telemetry Data –File Rate of Growth

Name	FileRate
Description	FileRate contains the average rate of growth of the current file in bytes per second since the last housekeeping request.
Data Type	Unsigned 32 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FileState[].FileRate*

Table 37 Telemetry Data – File Size

Name	FileSize
Description	FileSize contains the size of the current file in bytes.
Data Type	Unsigned 32 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FileState[].FileSize*

Table 38 Telemetry Data – File Open State

Name	OpenState
Description	OpenState contains the open or close state of the current file.
Data Type	Unsigned 16 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FileState[].OpenState*

Table 39 Telemetry Data - Filename Sequence Count

Name	SequenceCount
Description	SequenceCount contains the sequence counter value that will be inserted into the filename of the next file to be created.
Data Type	Unsigned 32 Bit Integer
Telemetry Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_FileState[].FileSeq*

A.3 Configuration Parameters

This section shows the DS Configuration Parameters provided as a default by the CFS DS application. Space is provided below for these values to be updated by the mission. If the mission has configured non-default values, they will be documented in the flight software RDL database.

While configuration parameters cannot be changed by the FOT, and are generally never changed after launch except by FSSE, the FOT needs to know the mission-specific values that have been incorporated into the software at the time the software was finalized and compiled.

(From an FSSE standpoint, configuration parameters are defined in the source code in ds_platform_cfg.h)

Table 40 Configuration Parameter - Application Pipe Depth

Configuration Parameter	DS_APP_PIPE_DEPTH
Value	*CFS DS Default: 256*
Purpose	Application Pipe Depth
Description	• This parameter defines the depth of the DS input pipe. • The depth should be deep enough to accommodate all of the DS command packets and all of the subscribed telemetry packets that might be generated by applications with a priority higher than the DS application.
Limits	The value must be greater than zero and cannot exceed the definition of CFE_SB_MAX_PIPE_DEPTH.

Table 41 Configuration Parameter - Application Pipe Name

Configuration Parameter	DS_APP_PIPE_NAME
Value	*CFS DS Default: "DS_CMD_PIPE"*
Purpose	Application Pipe Name
Description	• This parameter defines a portion of the logical name used during the creation of the DS input pipe. • The logical name may also be used as an identifier when accessing status for the pipe via cFE Software Bus Services. • The entire logical name is "DS.DS_CMD_PIPE".
Limits	The string length (including string terminator) cannot exceed the maximum size allowed by the Operating System (OS) (OS_MAX_API_NAME). The limit is not verified.

Table 42 Configuration Parameter - Destination File Table - Basename Buffer Size

Configuration Parameter	DS_BASENAME_BUFSIZE
Value	*CFS DS Default: OS_MAX_PATH_LEN*
Purpose	Destination File Table - Basename Buffer Size
Description	• This parameter further defines the size of the Destination File Table by setting the size of the basename buffer for each file entry. • Note that the buffer must contain both the string and the string terminator - so the max string length is one less than the buffer size.
Limits	• The buffer size must be greater than zero and a multiple of four bytes for alignment. • The value cannot exceed the maximum filename size allowed by the OS (OS_MAX_PATH_LEN).

Table 43 Configuration Parameter - Store DS Packet Processor State in CDS

Configuration Parameter	DS_CDS_ENABLE_STATE
Value	*CFS DS Default: 1*
Purpose	Enable restore of the DS Packet Processor State from the Critical Data Store following a processor reset
Description	• Set this parameter to a value of one and DS will restore the current packet processor state from the Critical Data Store (CDS) following a processor reset. After a processor reset, the packet processor state will be restored to the same state as before the reset. After a power cycle or any other reset that does not maintain the Critical Data Store, the packet processor state will be set to DS_DEF_ENABLE_STATE. • Set the value to zero and DS will not restore the current packet processor state from the Critical Data Store following a reset. The packet processor state will be set do DS_DEF_ENABLE_STATE.
Limits	This parameter must be set to zero or one.

Table 44 Configuration Parameter - Destination File Table - Default Table Filename

Configuration Parameter	DS_DEF_DEST_FILENAME
Value	*CFS DS Default: "/cf/apps/ds_file_tbl.tbl"*
Purpose	Destination File Table - Default Table Filename

Description	This parameter defines the default filename for the Destination File Table.
Limits	The string length (including string terminator) cannot exceed the maximum filename size allowed by the OS [(OS_MAX_PATH_LEN). The limit is not verified.]

Table 45 Configuration Parameter - Default DS Packet Processor State

Configuration Parameter	DS_DEF_ENABLE_STATE
Value	*CFS DS Default: 1*
Purpose	Default DS Packet Processor State
Description	• Set this parameter to a value of one and DS will begin to process packets immediately on startup. • Set the value to zero and DS will ignore data packets until receipt of a valid DS_ENABLE command.
Limits	This parameter must be set to zero or one.

Table 46 Configuration Parameter - Packet Filter Table - Default Table Filename

Configuration Parameter	DS_DEF_FILTER_FILENAME
Value	*CFS DS Default: " /cf/apps/ds_filter_tbl.tbl"*
Purpose	Packet Filter Table - Default Table Filename
Description	This parameter defines the default filename for the Packet Filter Table.
Limits	The string length (including string terminator) cannot exceed the maximum filename size allowed by the OS [(OS_MAX_PATH_LEN). The limit is not verified.]

Table 47 Configuration Parameter - Common Table File - Descriptor Text Buffer Size

Configuration Parameter	DS_DESCRIPTOR_BUFSIZE
Value	*CFS DS Default: 32*
Purpose	Common Table File - Descriptor Text Buffer Size

Description	• This parameter defines the size of the Descriptor Text fields in both the Destination File Table and the Packet Filter Table. • The buffer includes the string terminator.
Limits	The buffer size must be greater than zero and a multiple of four bytes for alignment. There is no upper limit.

Table 48 Configuration Parameter - Destination File Table - Number of Files

Configuration Parameter	DS_DEST_FILE_CNT
Value	*CFS DS Default: 16*
Purpose	Destination File Table - Number of Files
Description	• This parameter defines the size of the DS Destination File Table by setting the number of file entries in the table. • The number should be large enough to provide an entry for all the destination files defined for the project. • Maintenance will be simplified if file index 'n' always describes the same file - even if that file is not in use at the present time.
Limits	The number must be greater than zero but there is no upper enforced limit for this parameter.

Table 49 Configuration Parameter - Destination File Table - Logical Table Name

Configuration Parameter	DS_DESTINATION_TBL_NAME
Value	*CFS DS Default: "FILE_TBL"*
Purpose	Destination File Table - Logical Table Name
Description	• This parameter defines the name of the DS Destination File Table when referenced via cFE Table Services. • Note that this parameter is not a filename, but rather the table specific portion of the logical name. • The entire logical name for this table is "DS.FILE_TBL".
Limits	The string length (including string terminator) cannot exceed the cFE configuration parameter CFE_TBL_MAX_NAME_LENGTH. The limit is not verified.

Table 50 Configuration Parameter - Destination File Table - Extension Buffer Size

Configuration Parameter	DS_EXTENSION_BUFSIZE
Value	*CFS DS Default: 8*
Purpose	Destination File Table - Extension Buffer Size
Description	• This parameter further defines the size of the Destination File Table by setting the size of the extension buffer for each file entry. • Note that the buffer must contain both the string and the string terminator - so the max string length is one less than the buffer size.
Limits	• The buffer size must be greater than zero and a multiple of four bytes for alignment. • The value cannot exceed the maximum filename size allowed by the OS (OS_MAX_PATH_LEN).

Table 51 Configuration Parameter - Data Storage File - cFE File Header Description

Configuration Parameter	DS_FILE_HDR_DESCRIPTION
Value	*CFS DS Default: "DS data storage file"*
Purpose	Data Storage File - cFE File Header Description
Description	This parameter defines a mission-specific text string that may be used to identify Data Storage files.
Limits	The string length including string terminator cannot exceed CFE_FS_HDR_DESC_MAX_LEN. The limit is not verified.

Table 52 Configuration Parameter - Data Storage File - cFE File Header Sub-Type

Configuration Parameter	DS_FILE_HDR_SUBTYPE
	CFS DS Default: 12345
Purpose	Data Storage File - cFE File Header Sub-Type
Description	This parameter defines a mission-specific value that is used to identify a Data Storage file.
Limits	The file header data type for the value is 32 bits unsigned, thus the value can be anything from zero to 4,294,967,295. The limit is not verified.

Table 53 Configuration Parameter - File Header Type Selection

Configuration Parameter	DS_FILE_HEADER_TYPE
Value	*CFS DS Default: 1*
Purpose	File Header Type Selection
Description	Set this parameter to select the type of file header that will be the first record written to each Data Storage File.
Limits	This parameter must be set to one of the following: • 0 = none -- set this value to have no file header • 1 = CFE -- set this value to use cFE file headers • 2 = GPM -- set this value to use GPM file headers

Table 54 Configuration Parameter - Data Storage File - Minimum Age Limit

Configuration Parameter	DS_FILE_MIN_AGE_LIMIT
Value	*CFS DS Default: 60*
Purpose	Data Storage File - Minimum Age Limit
Description	This parameter defines the lower limit for commands and table entries that define the age (in seconds) when a data storage file is automatically closed.
Limits	None. **Caution:** A very small value will allow an age limit that closes files too frequently, while a very large value will effectively prevent files from ever being closed due to age.

Table 55 Configuration Parameter - Data Storage File - Minimum Size Limit

Configuration Parameter	DS_FILE_MIN_SIZE_LIMIT
Value	*CFS DS Default: 1024*
Purpose	Data Storage File - Minimum Size Limit
Description	This parameter defines the lower limit for commands and table entries that define the size (in bytes) when a data storage file is automatically closed.

Limits	**Caution:** A very small value will allow a size limit that closes files too frequently, while a very large value will effectively prevent files from ever being closed due to size.

Table 56 Configuration Parameter - Packet Filter Table - Logical Table Name

Configuration Parameter	DS_FILTER_TBL_NAME
Value	*CFS DS Default: "FILTER_TBL"*
Purpose	Packet Filter Table - Logical Table Name
Description	• This parameter defines the name of the DS Packet Filter Table when referenced via cFE Table Services. • Note that this parameter is not a filename. It is the table specific portion of the logical name. • The entire logical name for this table is "DS.FILTER_TBL".
Limits	The string length (including string terminator) cannot exceed the cFE configuration parameter CFE_TBL_MAX_NAME_LENGTH. The limit is not verified.

Table 57 Configuration Parameter - Packet Filter Table - Filters Per Packet

Configuration Parameter	DS_FILTERS_PER_PACKET
Value	*CFS DS Default: 4*
Purpose	Packet Filter Table - Filters Per Packet
Description	• This parameter further defines the size of the DS Packet Filter Table by setting the number of filters per packet entry. • This is the maximum number of destination files to which a single packet can be written (at one time).
Limits	The number of filters per packet must be greater than zero and not greater than DS_DEST_FILE_CNT.

Table 58 Configuration Parameter - Make DS Tables Critical

Configuration Parameter	DS_MAKE_TABLES_CRITICAL
Value	*CFS DS Default: 0*

Purpose	Make DS Tables Critical
Description	Set this parameter to a value of one to make the DS tables critical, otherwise set to zero.
Limits	This parameter must be set to zero or one.

Table 59 Configuration Parameter - Filename Sequence Count - Max Counter Value

Configuration Parameter	DS_MAX_SEQUENCE_COUNT
Value	*CFS DS Default: 99999999*
Purpose	Filename Sequence Count - Max Counter Value
Description	This parameter defines the maximum value a file sequence count will reach before rollover.
Limits	The value must be greater than zero and should not have more digits than the number of sequence count digits defined for DS_SEQUENCE_DIGITS, above.

Table 60 Configuration Parameter - Mission Specific Version Number for DS Application

Configuration Parameter	DS_MISSION_REV
Value	*CFS DS Default: 0*
Purpose	Mission Specific Version Number for DS Application
Description	• An application version number consists of four parts: major version number, minor version number, revision number and mission specific revision number. • The mission specific revision number is defined here and the other parts are defined in "ds_version.h".
Limits	Must be defined as a numeric value that is greater than or equal to zero.

Table 61 Configuration Parameter - Move Files to Downlink Directory After Close Selection

Configuration Parameter	DS_MOVE_FILES
Value	*CFS DS Default: FALSE*

Purpose	Move Files to Downlink Directory After Close Selection
Description	• Set this parameter to enable the code and structures to automatically move DS files to another directory after closing the files. • The intended use for this setting is to move files from a working directory into a directory from which the files can be downlinked. • Note that even after enabling this feature, files will not be moved if the move pathname in the Destination File Table is null.
Limits	This parameter must be set to one of the following: • TRUE = add move pathname field to Destination File Table. • FALSE = do not add move pathname to Destination File Table.

Table 62 Configuration Parameter - Packet Filter Table - Number of Packets

Configuration Parameter	DS_PACKETS_IN_FILTER_TABLE
Value	*CFS DS Default: 256*
Purpose	Packet Filter Table - Number of Packets
Description	• This parameter defines the size of the DS Packet Filter Table by setting the number of packet entries in the table. • The number should be large enough to provide an entry for each command and telemetry packet subject to data storage.
Limits	• The number must be greater than zero but there is no upper enforced limit for this parameter. • CAUTION: Setting the size equal to the number of packets that might be subject to data storage, rather than the total number of packets defined for this project, will reduce the table file size, possibly significantly.

Table 63 Configuration Parameter - Destination File Table - Pathname Buffer Size

Configuration Parameter	DS_PATHNAME_BUFSIZE
Value	*CFS DS Default: OS_MAX_PATH_LEN*
Purpose	Destination File Table - Pathname Buffer Size
Description	• This parameter further defines the size of the Destination File Table by setting the size of the pathname buffer for each file entry. • Note that the buffer must contain both the string and the string terminator - so the max string length is one less than the buffer size.

Limits	• The buffer size must be greater than zero and a multiple of four bytes for alignment. • The value cannot exceed the maximum filename size allowed by the OS (OS_MAX_PATH_LEN).

Table 64 Configuration Parameter - Application per Packet Pipe Limit

Configuration Parameter	DS_PER_PACKET_PIPE_LIMIT
Value	*CFS DS Default: 50*
Purpose	Application per Packet Pipe Limit
Description	• This parameter defines the per packet pipe limit. • This is the max number of packets with the same Message ID that may be in the DS input pipe at any one time. • This value should be large enough to accommodate a burst of packets (usually event packets) plus a suitable margin.
Limits	The value must be greater than zero and cannot exceed the definition of DS_APP_PIPE_DEPTH.

Table 65 Configuration Parameter - Housekeeping Request Frequency

Configuration Parameter	DS_SECS_PER_HK_CYCLE
Value	*CFS DS Default: 4*
Purpose	Housekeeping Request Frequency
Description	• Set this parameter equal to the number of seconds between housekeeping request commands. • This number is mission specific and must match the frequency used by the source of the command - often the scheduler task. • The value is used by the DS application to measure file age and also as a factor in the calculation of file growth rates.
Limits	This parameter must be greater than zero.

Table 66 Configuration Parameter - Filename Sequence Count - Number of Digits

Configuration Parameter	DS_SEQUENCE_DIGITS
Value	*CFS DS Default: 8*

Purpose	Filename Sequence Count - Number of Digits
Description	• This parameter defines the number of filename sequence count digits used when the filename type has been set to "count" rather than "time". • Sequence counts are padded with leading zero's to create fixed length strings.
Limits	• The number of sequence count digits must be greater than zero - even if there is no intention of later setting the filename type to "count". • This value should match the definition for DS_MAX_SEQUENCE_COUNT, below.

Table 67 Configuration Parameter - Data Storage File - Total Filename Size

Configuration Parameter	DS_TOTAL_FNAME_BUFSIZE
Value	*CFS DS Default: OS_MAX_PATH_LEN*
Purpose	Data Storage File - Total Filename Size
Description	This parameter defines the maximum size of a filename after combining the pathname, basename, sequence and extension.
Limits	• The buffer size must be greater than zero and a multiple of four bytes for alignment. • The buffer size (including string terminator) cannot exceed the value allowed by the OS (OS_MAX_PATH_LEN).

A.4 DS Commands

The tables in this section show detailed descriptions of all the standard commands available to ground controllers for use with the DS application.

Table 68 Command 0 – No-op

CFS DS Command Code	0
Command Name	No-Operation (No-op) Command
Description	• This command will increment the command execution counter and send an event containing the version number of the application. • The command is often used as a general test to verify the application is still "alive".
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_NOOP*
Command Verification	Successful execution of this command may be verified with the following telemetry: • $sc_$cpu_DS_CMDPC (or as defined by the mission) - command execution counter will increment • The Event ID 31 informational event message (No-op) will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length
Failure Evidence	Evidence of failure may be found in the following telemetry: • $sc_$cpu_DS_CMDEC (or as defined by the mission) - command error counter will increment • The Event ID 32 error event message (No-op) will be sent
Criticality	None

Table 69 Command 1 – Reset Housekeeping Telemetry Counters

CFS DS Command Code	1
Command Name	Reset Housekeeping Telemetry Counters Command

Description	The ability to reset the housekeeping counters is Important for testing and on orbit flight operations in order to start with a "clean slate". This command will set the following housekeeping counters to zero: • Command Execution Counter *($sc_$cpu_DS_CMDPC)** • Command Error Counter *($sc_$cpu_DS_CMDEC) ** • Ignored Packet Counter *($sc_$cpu_DS_IgnoredPktCnt) ** • Filtered Packet Counter *($sc_$cpu_DS_FilteredPktCnt) ** • Passed Packet Counter *($sc_$cpu_DS_PassedPktCnt) ** • File Access Counter *($sc_$cpu_DS_FileWriteCnt) ** • File Access Error Counter *($sc_$cpu_DS_FileWriteErrCnt) ** • Destination Table Load Counter *($sc_$cpu_DS_DestLoadCnt) ** • Filter Table Load Counter *($sc_$cpu_DS_FilterLoadCnt) ** • Destination Table Pointer Error Counter *($sc_$cpu_DS_DestPtrErrCnt) ** • Packet Filter Table Pointer Error Counter *($sc_$cpu_DS_FilterPtrErrCnt) ** ** or as defined by the mission*
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_ResetCtrs*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 33 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 34 error event message (Reset Housekeeping Counters) will be sent
Criticality	None

Table 70 Command 2 – Set Enable/Disable State For DS Application

CFS DS Command Code	2
Command Name	Set Enable/Disable State For DS Application
Description	This command will modify the Enable/Disable State selection for the DS application. No packets are stored while DS is disabled. The Disable command stops DS from filtering and storing messages. Since DS has multiple tables that are tightly coupled, FOT may use this command when needed to stop DS from new use of tables until all tables are updated.
Command Mnemonic(s)	*CFS DS Default:* • *$sc_$cpu_DS_Enable (fixed command arg = enable)* • *$sc_$cpu_DS_Disable (fixed command arg = disable)*
Mission Specific Command Mnemonic(s)	Mission Defined
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 35 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid enable/disable state selection
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 36 error event message (Set Application Enable/Disable State) will be sent
Criticality	**Potential flood of OS errors.** FOT needs to ensure that the disk has been initialized and partitions created before enabling DS to avoid a flooding of OS errors when trying to create and write to files.

Table 71 Command 3 – Set Destination File For Packet Filter Table Entry

CFS DS Command Code	3

Command Name	Set Destination File For Packet Filter Table Entry
Description	This command will modify the Destination File selection for the indicated entry in the Packet Filter Table.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_SetFilterFile*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 37 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Packet Filter Table index • Invalid filter parameters index • Invalid destination file selection • Packet Filter Table is not currently loaded • Cannot modify unused Packet Filter Table entry
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 38 error event message (Set Filter File Index) will be sent
Criticality	None

Table 72 Command 4 – Set Filter Type For Packet Filter Table Entry

CFS DS Command Code	4
Command Name	Set Filter Type For Packet Filter Table Entry
Description	This command will modify the Filter Type selection for the indicated entry in the Packet Filter Table.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_SetFilterType*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The DS_FTYPE_CMD_EID debug event message will be sent

Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Packet Filter Table index • Invalid filter parameters index • Invalid filter type selection • Packet Filter Table is not currently loaded • Cannot modify unused Packet Filter Table entry
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 40 error event message (Set Filter Type) will be sent
Criticality	None

Table 73 Command 5 – Set Algorithm Parameters For Packet Filter Table Entry

CFS DS Command Code	DS_SET_FILTER_PARMS_CC 5
Command Name	Set Algorithm Parameters For Packet Filter Table Entry
Description	This command will modify the Algorithm Parameters for the indicated entry in the Packet Filter Table.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_SetFilterParams*
Command Verification	Successful execution of this command may be verified with the following telemetry: • $sc_$cpu_DS_CMDPC (or as defined by the mission) - command execution counter will increment • The Event ID 41 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Packet Filter Table index • Invalid filter parameters index • Algorithm_N value cannot be greater than Algorithm_X • Packet Filter Table is not currently loaded • Cannot modify unused Packet Filter Table entry
Failure Evidence	Evidence of failure may be found in the following telemetry: • $sc_$cpu_DS_CMDEC (or as defined by the mission) - command error counter will increment • The Event ID 42 error event message (Set Filter Parameters) will be sent

Criticality	None

Table 74 Command 6 – Set Filename Type For Destination File Table Entry

CFS DS Command Code	6
Command Name	Set Filename Type For Destination File Table Entry
Description	This command will modify the Filename Type selection for the indicated entry in the Destination File Table.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_SetFileType*
Command Verification	Successful execution of this command may be verified with the following telemetry: • $sc_$cpu_DS_CMDPC (or as defined by the mission) - command execution counter will increment • The Event ID 43 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Destination File Table index • Invalid filename type selection • Destination File Table is not currently loaded
Failure Evidence	Evidence of failure may be found in the following telemetry: • $sc_$cpu_DS_CMDEC (or as defined by the mission) - command error counter will increment • The Event ID 44 error event message (Set Filename Type) will be sent
Criticality	None

Table 75 Command 7 – Set Enable/Disable State For Destination File Table Entry

CFS DS Command Code	7
Command Name	Set Enable/Disable State For Destination File Table Entry

Description	This command will modify the Enabled/Disabled State selection for the indicated entry in the Destination File Table. The enabled state provides the ground the ability to write files in a file set without having to load a table. The disabled state provides ground ability to quickly stop writing files in a file set without having to load a table.
Command Mnemonic(s)	*CFS DS Default* • *$sc_$cpu_DS_EnableFile (fixed command arg = enable)* • *$sc_$cpu_DS_DisableFile (fixed command arg = disable)*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 45 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Destination File Table index • Invalid destination state selection • Destination File Table is not currently loaded
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 46 error event message (Set File Enable/Disable State) will be sent
Criticality	**Potential packet loss.** It is the responsibility of the user to verify the FileID specifying the destination to be disabled. The user should use caution when issuing this command.

Table 76 Command 8 – Set Pathname For Destination File Table Entry

CFS DS Command Code	8
Command Name	Set Pathname For Destination File Table Entry
Description	This command will modify the Pathname portion of the filename for the indicated entry in the Destination File Table.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_SetPath*

Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 47 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Destination File Table index • Invalid pathname string (empty, no terminator, bad characters) • Destination File Table is not currently loaded
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 48 error event message (Set Filename Path) will be sent
Criticality	**Potential packet loss.** It is the responsibility of the user to verify the new Pathname and ensure it is a valid path. The user should use caution when issuing this command.

Table 77 Command 9 – Set Basename For Destination File Table Entry

CFS DS Command Code	9
Command Name	Set Basename For Destination File Table Entry
Description	This command will modify the Basename portion of the filename for the indicated entry in the Destination File Table. • This command affects the next file opened in the File Set. • This command does not affect the File Sequence Counter.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_SetBaseName*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 49 debug event message will be sent

Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Destination File Table index • Invalid basename string (no terminator, bad characters) • Destination File Table is not currently loaded
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 50 error event message (Set File Basename) will be sent
Criticality	None

Table 78 Command 10 – Set Extension For Destination File Table Entry

CFS DS Command Code	10
Command Name	Set Extension For Destination File Table Entry
Description	This command will modify the Extension portion of the filename for the indicated entry in the Destination File Table.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_SetExtension*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 51 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Destination File Table index • Invalid extension string (empty, no terminator, bad characters) • Destination File Table is not currently loaded
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 52 error event message (Set Filename Extension) will be sent
Criticality	None

Table 79 Command 11 – Set Max File Size For Destination File Table Entry

CFS DS Command Code	11
Command Name	Set Max File Size For Destination File Table Entry
Description	This command will modify the max file size selection for the indicated entry in the Destination File Table.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_SetMaxFileSize*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 53 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Destination File Table index • Invalid max file size selection • Destination File Table is not currently loaded
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 54 error event message (Set Max File Size) will be sent
Criticality	**Potential to fill the storage device and/or hog CPU.** Downlinking many small files can be CPU intensive. It is the responsibility of the user to verify the File Size. The user should use caution when issuing this command.

Table 80 Command 12 – Set Max File Age For Destination File Table Entry

CFS DS Command Code	12
Command Name	Set Max File Age For Destination File Table Entry
Description	This command will modify the max file age selection for the indicated entry in the Destination File Table.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_SetMaxFileAge*

Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 55 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Destination File Table index • Invalid max file age selection • Destination File Table is not currently loaded
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 56 error event message (Set Max File Age) will be sent
Criticality	**Potential to fill the storage device and/or hog CPU.** Downlinking many small files can be CPU intensive. It is the responsibility of the user to verify the File Size. The user should use caution when issuing this command.

Table 81 Command 13 – Set Sequence Count For Destination File Table Entry

CFS DS Command Code	DS_SET_DEST_COUNT_CC 13
Command Name	Set Sequence Count For Destination File Table Entry
Description	This command will modify the sequence count value for the indicated entry in the Destination File Table. • Use of this command affects the next file opened in the File Set. • This command does not affect the File Sequence Counter. • This command could be used in the event of a power-on reset so that sequence numbers continue. • However, in a cFE Processor reset, use of the command for this purpose should be unnecessary because the sequence numbers should already be preserved.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_SetFileSeqCtr*

Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 57 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Destination File Table index • Invalid sequence count value • Destination File Table is not currently loaded
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 58 error event message (Set Sequence Count) will be sent
Criticality	None

Table 82 Command 14 – Close Destination File

CFS DS Command Code	DS_CLOSE_FILE_CC 14
Command Name	Close Destination File
Description	This command will close the indicated Destination File so that it can be downlinked.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_CloseFile*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 59 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Invalid Destination File Table index • Destination File Table is not currently loaded

Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 60 error event message (Close Destination File) will be sent
Criticality	None

Table 83 Command 15 – Get File Info Packet

CFS DS Command Code	15
Command Name	Get File Info Packet
Description	This command will send the DS File Info Packet.
Command Mnemonic(s)	*CFS DS Default: (N/A)*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The DS_FileInfoPkt_t packet will be sent
Error Condition	This command may fail for the following reason(s): • Invalid command packet length
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 63 error event message (Get File Info) will be sent
Criticality	None

Table 84 Command 16 – Add Message ID to Packet Filter Table

CFS DS Command Code	16
Command Name	Add Message ID To Packet Filter Table
Description	This command will change the Message ID selection for an unused Packet Filter Table entry to the indicated value.

Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_AddMID*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 64 debug event message will be sent
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Message ID is invalid (can be anything but zero) • Packet Filter Table is not currently loaded • Message ID already exists in Packet Filter Table
Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 65 error event message (Add Message ID to Packet Filter Table) will be sent
Criticality	None

Table 85 Command 17 – Close All Destination Files

CFS DS Command Code	17
Command Name	Close All Destination Files
Description	• This command will close all open Destination Files. • NOTE: Using this command may incur a performance hit based upon the number and size of the files being closed.
Command Mnemonic(s)	*CFS DS Default: $sc_$cpu_DS_CloseAll*
Command Verification	Successful execution of this command may be verified with the following telemetry: • *$sc_$cpu_DS_CMDPC (or as defined by the mission)* - Command execution counter will increment • The Event ID 66 debug event message will be sent (before launch, when debug events are generally turned off)
Error Conditions	This command may fail for the following reason(s): • Invalid command packet length • Destination File Table is not currently loaded

Failure Evidence	Evidence of failure may be found in the following telemetry: • *$sc_$cpu_DS_CMDEC (or as defined by the mission)* - Command error counter will increment • The Event ID 67 error message will be sent
Criticality	None

A.5 Event Messages

This section shows cFE Software Bus event messages. Event messages are not mission specific. Generally, if supported by ground flight software systems (e.g., ASIST or ITOS), the Event ID numbers shown in the subsections below should appear in telemetry, no matter the mission.

A.5.1 Event Messages - Critical

The tables in this section show **critical** event messages for the DS application.

Table 86 Event ID 3 (Critical) - Application Termination

Event ID Number	3
Event Name	'Application Termination Event'
Event Message	'Application terminating, err = 0x%08X'
Type	CRITICAL
Cause	• This event message is issued when the application performs a voluntary termination. The termination may be caused by a cFE command requesting the application be terminated - or, the cause may result from the DS application having received an error when invoking a cFE or operating system (OS) function necessary for normal operation. • The value displayed is either zero (if cFE requested the termination) or the return code from the function that failed.

A.5.2 Event Messages - Error

The tables in this section show **error** event messages for the DS application.

Table 87 Event ID 2 (Error) - Application Initialization

Event ID Number	2
Event Name	'Application Initialization (error)'
Event Message	• 'Unable to register for EVS services, err = 0x%08X' • 'Unable to create input pipe, err = 0x%08X' • 'Unable to subscribe to housekeeping request commands, err = 0x%08X' • 'Unable to subscribe to 1Hz wakeup commands, err = 0x%08X' • 'Unable to subscribe to DS commands, err = 0x%08X'

Type	ERROR
Cause	• This event message is issued when the application receives an error while performing cFE initialization. • The value displayed is the return code from the function that failed.

Table 88 Event ID 6 (Error) - Critical Data Store Access

Event ID Number	6
Event Name	'Critical Data Store Access (error)'
Event Message	'Critical Data Store access error = 0x%08X'
Type	ERROR
Cause	• The DS application stores the current values for Destination Filename sequence counts in the CDS. This ensures that DS will not overwrite old data storage files following a processor reset. • This event indicates an error at startup as DS is initializing access to the CDS. Subsequent CDS errors are ignored by DS.

Table 89 Event ID 8 (Error) - Table Create

Event ID Number	8
Event Name	'Table Create (error)'
Event Message	• 'Unable to register Filter Table: Error = 0x%08X' • 'Unable to register Destination File Table: Error = 0x%08X' • 'Unable to load default Destination File Table: Filename = '%s', Error = 0x%08X' • 'Unable to load default Filter Table: Filename = '%s', Error = 0x%08X'
Type	ERROR
Cause	• At startup, the DS application must create both a Packet Filter Table and a Destination File Table. After creating the tables, DS must then load default table data for each of the files. If DS is unable to create either table then the application will terminate. If DS is able to create both tables and is then unable to load both tables then DS will operate in a reduced function mode - thus allowing table load error recovery via the cFE Table Services command interface. • The values displayed indicate the cFE Table Services API function return code and the table filename, if appropriate.

Table 90 Event ID 11 (Error) - Destination File Validation

Event ID Number	11
Event Name	'Destination File Validation (error)'
Event Message	• 'Destination file table verify err: invalid descriptor text' • 'Destination file table verify err: index = %d, invalid pathname text' • 'Destination file table verify err: index = %d, invalid basename text' • 'Destination file table verify err: index = %d, invalid extension text' • 'Destination file table verify err: index = %d, filename type = %d' • 'Destination file table verify err: index = %d, file enable state = %d' • 'Destination file table verify err: index = %d, max file size = %d' • 'Destination file table verify err: index = %d, max file age = %d' • 'Destination file table verify err: index = %d, sequence count = %d'
Type	ERROR
Cause	• These are the errors that the DS application might encounter when verifying the contents of a Destination File Table. Note that only the first error detected for each table file will result in an event being generated. The remaining entries are still tested to establish the result counters but no further error events will be generated during this call to the verify function. • The event text will indicate the table index value (where appropriate), the invalid field name and the field value for non-string fields. If a string field is invalid then attempts to display the string could result in other errors.

Table 91 Event ID 13 (Error) – Packet Filter Table Validation

Event ID Number	13
Event Name	'Packet Filter Table Validation (error)'
Event Messages	• 'Filter table verify err: invalid descriptor text' • 'Filter table verify err: MID = 0x%04X, index = %d, filter = %d, file table index = %d' • 'Filter table verify err: MID = 0x%04X, index = %d, filter = %d, filter type = %d' • 'Filter table verify err: MID = 0x%04X, index = %d, filter = %d, filter parms N = %d, X = %d, O = %d'
Type	ERROR

Cause	• These are the errors that the DS application might encounter when verifying the contents of a Packet Filter Table. Note that only the first error detected for each table file will result in an event being generated. The remaining entries are still tested to establish the result counters but no further error events will be generated during this call to the verify function. • The event text will indicate the Message ID, table index value and filter index value (where appropriate), the field name and the field value for non-string fields. If a string field is invalid then attempts to display the string could result in other errors.

Table 92 Event ID 14 (Error) - Filename Create

Event ID Number	14
Event Name	'Filename Create (error)'
Event Message	'FILE NAME error: dest = %d, path = '%s', base = '%s', seq = '%s', ext = '%s''
Type	ERROR
Cause	• This error occurs when an attempt to create a destination filename fails. The expected cause of this error is for the sum of the lengths of each filename element to be invalid, even though each individual element length is valid. Note that each of the individual element strings has been previously verified. • The event text will indicate the Destination File Table index, the file path string, the file basename string, the filename sequence string and the extension portion of the filename. • Note that the filename create error handler will also disable the destination to prevent the immediate re-occurrence of the error as more packets are written to this destination.

Table 93 Event ID 15 (Error) - File Create

Event ID Number	15
Event Name	'File Create (error)'
Event Message	'FILE CREATE error: result = %d, dest = %d, name = '%s''
Type	ERROR

Cause	• This error occurs when an attempt to create a destination file fails. The most common cause of this error is insufficient free space available on the target file system (disk). However, it is possible to specify a filename that is legal but still invalid (i.e. the path string has legal chars but does not exist). • The event text will indicate the result from the system function call to create the file, the Destination File Table index and the filename. • Note that the file create error handler will also disable the destination to prevent the immediate re-occurrence of the error as more packets are written to this destination.

Table 94 Event ID 16 (Error) - File Write

Event ID Number	16
Event Name	'File Write (error)'
Event Message	'FILE WRITE error: result = %d, length = %d, dest = %d, name = '%s"
Type	ERROR
Cause	• This error occurs when an attempt to write to a destination file fails. • The most common cause of this error is insufficient free space available on the target file system (disk). Other causes for this error imply some sort of operational or file system failure and will require careful analysis. • The event text will indicate the result from the system function call to write to the file, the Destination File Table index and the filename. • Note that the file write error handler will also close the file and disable the destination to prevent the immediate re-occurrence of the error as more packets are written to this destination.

Table 95 Event ID 21 (Error) - Invalid Packet

Event ID Number	21
Event Name	'Invalid Packet (command code)'
Event Message	'Invalid command code: MID = 0x%04X, CC = %d'
Type	ERROR
Cause	• This error occurs when a DS application command is received that has an undefined command code (CC). Ground systems generally prevent these errors so the cause is more likely to result from sending "raw" ground commands or on-board stored commands. • The event text will indicate the Message ID (MID) and command code.

Table 96 Event ID 22 (Error) - Housekeeping Request

Event ID Number	22
Event Name	'Housekeeping Request (error)'
Event Message	'Invalid HK request length: expected = %d, actual = %d'
Type	ERROR
Cause	• This event signals the failed execution of a HK request command. • The cause of the failure is an invalid command packet length.

Table 97 Event ID 32 (Error) - No-op

Event ID Number	32
Event Name	'No-op command (error)'
Event Message	'Invalid NOOP command length: expected = %d, actual = %d'
Type	ERROR
Cause	This event signals the failed execution of a NOOP command. The cause of the failure is an invalid command packet length.

Table 98 Event ID 34 (Error) - Reset Housekeeping Counters

Event ID Number	34
Event Name	'Reset Housekeeping Counters command (error)'
Event Message	'Invalid RESET command length: expected = %d, actual = %d'
Type	ERROR
Cause	This event signals the failed execution of a RESET command. The cause of the failure is an invalid command packet length.

Table 99 Event ID 36 (Error) - Set Application Enable/Disable State

Event ID Number	36
Event Name	'Set Application Enable/Disable State Command (error)'

Event Message	• 'Invalid APP STATE command length: expected = %d, actual = %d' • 'Invalid APP STATE command arg: app state = %d'
Type	ERROR
Cause	• This event signals the failed execution of a command to set the application enable/disable state. The cause of the failure may be an invalid command packet length or an invalid enable/disable state value. • The event text will indicate the cause of the failure.

Table 100 Event ID 38 (Error) - Set Filter File Index

Event ID Number	38
Event Name	'Set Filter File Index command (error)'
Event Message	• 'Invalid FILTER FILE command length: expected = %d, actual = %d' • 'Invalid FILTER FILE command arg: invalid messageID = 0x%04X' • 'Invalid FILTER FILE command arg: filter parameters index = %d' • 'Invalid FILTER FILE command arg: file table index = %d' • 'Invalid FILTER FILE command: packet filter table is not loaded' • 'Invalid FILTER FILE command: Message ID 0x%04X is not in filter table'
Type	ERROR
Cause	• This event signals the failed execution of a command to set the Destination File Table index for a Packet Filter Table entry. The cause of the failure may be an invalid command packet length, an invalid Message ID or an invalid filter parameters index. • The failure may also result from not having a Packet Filter Table loaded at the time the command was invoked or because the Message ID was not found in the Packet Filter Table. • The event text will indicate the cause of the failure.

Table 101 Event ID 40 (Error) - Set Filter Type

Event ID Number	40
Event Name	'Set Filter Type command (error)'
Event Message	• 'Invalid FILTER TYPE command length: expected = %d, actual = %d' • 'Invalid FILTER TYPE command arg: invalid messageID = 0x%04X' • 'Invalid FILTER TYPE command arg: filter parameters index = %d' • 'Invalid FILTER TYPE command arg: filter type = %d' • 'Invalid FILTER TYPE command: packet filter table is not loaded' • 'Invalid FILTER TYPE command: Message ID 0x%04X is not in filter table'

Type	ERROR
Cause	• This event signals the failed execution of a command to set the filter type for a Packet Filter Table entry. The cause of the failure may be an invalid command packet length, an invalid Message ID, an invalid Filter Parameters index or an invalid filter type. • The failure may also result from not having a Packet Filter Table loaded at the time the command was invoked or because the Message ID was not found in the Packet Filter Table. • The event text will indicate the cause of the failure.

Table 102 Event ID 42 (Error) - Set Filter Parameters

Event ID Number	42
Event Name	'Set Filter Parameters command (error)'
Event Message	• 'Invalid FILTER PARMS command length: expected = %d, actual = %d' • 'Invalid FILTER PARMS command arg: invalid messageID = 0x%04X' • 'Invalid FILTER PARMS command arg: filter parameters index = %d' • 'Invalid FILTER PARMS command arg: N = %d, X = %d, O = %d' • 'Invalid FILTER PARMS command: packet filter table is not loaded' • 'Invalid FILTER PARMS command: Message ID 0x%04X is not in filter table'
Type	ERROR3
Cause	• This event signals the failed execution of a command to set the filter parameters for a Packet Filter Table entry. The cause of the failure may be an invalid command packet length, an invalid Message ID, an invalid filter parameters index or an invalid filter parameters value (N,X,O). • The failure may also result from not having a Packet Filter Table loaded at the time the command was invoked or because the Message ID was not found in the Packet Filter Table. • The event text will indicate the cause of the failure.

Table 103 Event ID 44 (Error) - Set Filename Type

Event ID Number	44
Event Name	'Set Filename Type command (error)'
Event Message	• 'Invalid DEST TYPE command length: expected = %d, actual = %d' • 'Invalid DEST TYPE command arg: file table index = %d' • 'Invalid DEST TYPE command arg: filename type = %d' • 'Invalid DEST TYPE command: destination file table is not loaded'

Type	ERROR
Cause	• This event signals the failed execution of a command to set the filename type for the selected Destination File Table entry. The cause of the failure may be an invalid command packet length, an invalid Destination File Table index or an invalid filename type. • The failure may also result from not having a Destination File Table loaded at the time the command was invoked. • The event text will indicate the cause of the failure.

Table 104 Event ID 46 (Error) - Set File Enable/Disable State

Event ID Number	46
Event Name	'Set File Enable/Disable State command (error)'
Event Message	• 'Invalid DEST STATE command length: expected = %d, actual = %d' • 'Invalid DEST STATE command arg: file table index = %d' • 'Invalid DEST STATE command arg: file state = %d' • 'Invalid DEST STATE command: destination file table is not loaded'
Type	ERROR
Cause	• This event signals the failed execution of a command to set the enable/disable state for the selected Destination File Table entry. The cause of the failure may be an invalid command packet length, an invalid Destination File Table index or an invalid state selection. • The failure may also result from not having a Destination File Table loaded at the time the command was invoked. • The event text will indicate the cause of the failure.

Table 105 Event ID 48 (Error) - Set Filename Path

Event ID Number	48
Event Name	'Set Filename Path command (error)'
Event Message	• 'Invalid DEST PATH command length: expected = %d, actual = %d' • 'Invalid DEST PATH command arg: file table index = %d' • 'Invalid DEST PATH command arg: pathname' • 'Invalid DEST PATH command: destination file table is not loaded'
Type	ERROR

Cause	• This event signals the failed execution of a command to set the filename path text for a Destination File Table entry. The cause of the failure may be an invalid command packet length, an invalid Destination File Table index or invalid filename path text. • The failure may also result from not having a Destination File Table loaded at the time the command was invoked. • The event text will indicate the cause of the failure.

Table 106 Event ID 50 (Error) - Set File Basename

Event ID Number	50
Event Name	'Set File Basename command (error)'
Event Message	• 'Invalid DEST BASE command length: expected = %d, actual = %d' • 'Invalid DEST BASE command arg: file table index = %d' • 'Invalid DEST BASE command arg: base filename' • 'Invalid DEST BASE command: destination file table is not loaded'
Type	ERROR
Cause	• This event signals the failed execution of a command to set the base filename text for a Destination File Table entry. The cause of the failure may be an invalid command packet length, an invalid Destination File Table index or invalid base filename text. • The failure may also result from not having a Destination File Table loaded at the time the command was invoked. • The event text will indicate the cause of the failure.

Table 107 Event ID 52 (Error) - Set Filename Extension

Event ID Number	52
Event Name	'Set Filename Extension command (error)'
Event Message	• 'Invalid DEST EXT command length: expected = %d, actual = %d' • 'Invalid DEST EXT command arg: file table index = %d' • 'Invalid DEST EXT command arg: extension' • 'Invalid DEST EXT command: destination file table is not loaded'
Type	ERROR

Cause	• This event signals the failed execution of a command to set the filename extension text for a Destination File Table entry. The cause of the failure may be an invalid command packet length, an invalid Destination File Table index or invalid filename extension text. • The failure may also result from not having a Destination File Table loaded at the time the command was invoked. • The event text will indicate the cause of the failure.

Table 108 Event ID 54 (Error) - Set Max File Size

Event ID Number	54
Event Name	'Set Max File Size command (error)'
Event Message	• 'Invalid DEST SIZE command length: expected = %d, actual = %d' • 'Invalid DEST SIZE command arg: file table index = %d' • 'Invalid DEST SIZE command arg: size limit = %d' • 'Invalid DEST SIZE command: destination file table is not loaded'
Type	ERROR
Cause	• This event signals the failed execution of a command to set the file size limit for a Destination File Table entry. The cause of the failure may be an invalid command packet length, an invalid Destination File Table index or an invalid file size limit. • The failure may also result from not having a Destination File Table loaded at the time the command was invoked. • The event text will indicate the cause of the failure.

Table 109 Event ID 55 (Error) - Set Max File Age

Event ID Number	55
Event Name	'Set Max File Age command (error)'
Event Message	'DEST AGE command: file table index = %d, age limit = %d'
Type	DEBUG
Cause	• This event signals the successful execution of a command to set the file age limit for a Destination File Table entry. • The event text will indicate the Destination File Table index, and the new file age limit.

Table 110 Event ID 56 (Error) - Set Max File Age

Event ID Number	56

Event Name	'Set Max File Age command (error)'
Event Message	• 'Invalid DEST AGE command length: expected = %d, actual = %d' • 'Invalid DEST AGE command arg: file table index = %d' • 'Invalid DEST AGE command arg: age limit = %d' • 'Invalid DEST AGE command: destination file table is not loaded'
Type	ERROR
Cause	• This event signals the failed execution of a command to set the file age limit for a Destination File Table entry. The cause of the failure may be an invalid command packet length, an invalid Destination File Table index or an invalid file age limit. • The failure may also result from not having a Destination File Table loaded at the time the command was invoked. • The event text will indicate the cause of the failure.

Table 111 Event ID 58 (Error) - Set Sequence Count

Event ID Number	58
Event Name	'Set Sequence Count command (error)'
Event Message	• 'Invalid DEST COUNT command length: expected = %d, actual = %d' • 'Invalid DEST COUNT command arg: file table index = %d' • 'Invalid DEST COUNT command arg: sequence count = %d' • 'Invalid DEST COUNT command: destination file table is not loaded'
Type	ERROR
Cause	• This event signals the failed execution of a command to set the sequence count for a Destination File Table entry. The cause of the failure may be an invalid command packet length, an invalid Destination File Table index or an invalid sequence count. • The failure may also result from not having a Destination File Table loaded at the time the command was invoked. • The event text will indicate the cause of the failure.

Table 112 Event ID 60 (Error) - Close Destination File

Event ID Number	60
Event Name	'Close Destination File command (error)'
Event Message	• 'Invalid DEST CLOSE command length: expected = %d, actual = %d' • 'Invalid DEST CLOSE command arg: file table index = %d'

Type	ERROR
Cause	• This event signals the failed execution of a command to close the selected Destination File. The cause of the failure may be an invalid command packet length or an invalid Destination File Table index. • The event text will indicate the cause of the failure.

Table 113 Event ID 61 (Error) - File Move

Event ID Number	61
Event Name	'File Move (error)'
Event Message	• 'FILE MOVE error: dir name = '%s', filename = '%s'' • 'FILE MOVE error: src = '%s', tgt = '%s', result = %d'
Type	ERROR
Cause	• This event signals the failed attempt to move a Data Storage file from the temporary working directory it was created in, to the directory specified in the file definition table. The cause of the failure may be an invalid combined directory and filename length or the system file move function may have returned an error result. • The event text will indicate the cause of the failure.

Table 114 Event ID 63 (Error) - Get File Info

Event ID Number	63
Event Name	'Get File Info command (error)'
Event Message	'Invalid GET FILE INFO command length: expected = %d, actual = %d'
Type	ERROR
Cause	• This event signals the failed execution of a command to get the file info diagnostic packet. The cause of the failure is an invalid command packet length. • The event text will indicate the expected and actual packet size.

Table 115 Event ID 65 (Error) - Add Message ID to Packet Filter Table

Event ID Number	65
Event Name	'Add Message ID to Packet Filter Table command (error)'

Event Message	'Invalid ADD MID command length: expected = %d, actual = %d' 'Invalid ADD MID command arg: invalid MID = 0x%4X' 'Invalid ADD MID command: filter table is not loaded' 'Invalid ADD MID command: MID = 0x%4X is already in filter table at index = %d' 'Invalid ADD MID command: filter table is full'
Type	ERROR
Cause	- This event signals the failed execution of a command to add a new Message ID to the Packet Filter Table. - The cause of the failure may be an invalid command packet length or an invalid Message ID. - The failure may also result from not having a Packet Filter Table loaded at the time the command was invoked. The loaded table must have an unused entry available for the new Message ID and must not already contain the new Message ID. - The event text will indicate the cause of the failure.

Table 116 Event ID 67 (Error) - Close All Destination Files

Event ID Number	67
Event Name	'Close All Destination Files command (error)'
Event Message	'Invalid DEST CLOSE ALL command length: expected = %d, actual = %d'
Type	ERROR
Cause	- This event signals the failed execution of a command to close all open Destination Files. The cause of the failure can only be an invalid command packet length. - The event text will indicate the cause of the failure.

A.5.3   Event Messages - Information

The tables in this section show **information** type event messages for the DS application.

Table 117 Event ID 1 (Informational) - Application Initialization Event

Event ID Number	1
Event Name	'Application Initialization Event'
Event Message	'Application initialized, version %d.%d.%d.%d, data at 0x%08X'

Type	INFORMATION
Cause	This event message is issued when DS has successfully completed its initialization. • The first %d field contains the Application's Major Version Number. • The second %d field contains the Application's Minor Version Number. • The third %d field contains the Application's Revision Number. • The fourth %d field contains the Application's Mission Revision Number.

Table 118 Event ID 10 (Informational) - Destination File Table Validation Results

Event ID Number	10
Event Name	'Destination File Table Validation Results'
Event Message	'Destination file table verify results: desc text = %s, good = %d, bad = %d, unused = %d'
Type	INFORMATION
Cause	• This event describes the results of the Destination File Table validation function. The cFE Table Services Manager will call this function autonomously when the default table is loaded at startup and also whenever a table validate command (that targets this table) is processed. • The event text will indicate the descriptor text verification result (OK or bad), the number of table entries that were verified without error (good), the number of table entries that had one or more errors (bad) and the number of unused table entries (unused). Note that the totals also include the results of the verification test for the table description text. Thus, the sum of good + bad + unused results will equal the number of table entries plus one.

Table 119 Event ID 12 (Informational) – Packet Filter Table Validation Results

Event ID Number	12
Event Name	'Packet Filter Table Validation Results'
Event Message	'Filter table verify results: desc text = %s, good = %d, bad = %d, unused = %d'
Type	INFORMATION

Cause	• This event describes the results of the Packet Filter Table validation function. The cFE Table Services Manager will call this function autonomously when the default table is loaded at startup and also whenever a table validate command (that targets this table) is processed. • The event text will indicate the descriptor text verification result (OK or bad), the number of table entries that were verified without error (good), the number of table entries that had one or more errors (bad) and the number of unused table entries (unused). Note that the totals also include the results of the verification test for the table description text. Thus, the sum of good + bad + unused results will equal the number of table entries plus one.

Table 120 Event ID 31 (Informational) - No-op

Event ID Number	31
Event Name	'No-op command (info)'
Event Message	'NOOP command, Version %d.%d.%d.%d'
Type	INFORMATION
Cause	• This event signals the successful execution of an application NOOP command. The command is used primarily as an indicator that the DS application can receive commands and generate telemetry. • The first %d field contains the Major Version Number • The second %d field contains the Minor Version Number • The third %d field contains the Revision Number • The fourth %d field contains the Mission Revision Number.

A.5.4   *Event Messages - Debug*

The tables in this section show **debug** event messages for the DS application.

Table 121 Event ID 7 (Debug)

Event ID Number	7
Event Name	'Table Restored From CDS'
Event Message	'Destination File Table data restored from CDS' 'Filter Table data restored from CDS'
Type	DEBUG

Cause	• If configured to make DS tables critical, the DS application will try and recover the table data in use prior to a processor reset. • These startup initialization events indicate that the table data has been restored from the CDS.

Table 122 Event ID 33 (Debug)

Event ID Number	33
Event Name	'Reset Housekeeping Counters command'
Event Message	'Reset counters command'
Type	DEBUG
Cause	This event signals the successful execution of a RESET housekeeping counters command. The command is used primarily to clear counters that have already been examined.

Table 123 Event ID 35 (Debug)

Event ID Number	35
Event Name	'Set Application Enable/Disable State command'
Event Message	'APP STATE command: state = %d'
Type	DEBUG
Cause	• This event signals the successful execution of a command to set the application enable/disable state for processing data storage packets. The DS application is always enabled to receive and process commands. • The event text will indicate the new setting for the application enable/disable state.

Table 124 Event ID 37 (Debug)

Event ID Number	37
Event Name	'Set Filter File Index command'
Event Message	'FILTER FILE command: MID = 0x%X, index = %d, filter = %d, file = %d'
Type	DEBUG

Cause	• The Packet Filter Table consists of an array of packet structures, with each element in the array representing one packet Message ID. • Each packet structure contains an array of filter structures, and each filter structure contains a destination file selection. Thus, each packet has multiple filters and each filter selects a file. • File selections are expressed as an index into the Destination File Table. • This event signals the successful execution of a command to set the destination file index for a single filter entry in the Packet Filter Table. • The event text will indicate the packet Message ID, the Packet Filter Table index, the filter array index and the new Destination File Table index selection.

Table 125 Event ID 39 (Debug)

Event ID Number	39
Event Name	'Set Filter Type command'
Event Message	'FILTER TYPE command: MID = 0x%04X, index = %d, filter = %d, type = %d'
Type	DEBUG
Cause	• The Packet Filter Table consists of an array of packet structures, with each element in the array representing one packet Message ID. • Each packet structure contains an array of filter structures, and each filter structure contains a filter type selection. Thus, each packet has multiple filters and each filter selects a filter type. • This event signals the successful execution of a command to set the filter type for a single filter entry in the Packet Filter Table. • The event text will indicate the packet Message ID, the Packet Filter Table index, the filter array index and the new filter type selection.

Table 126 Event ID 41 (Debug)

Event ID Number	41
Event Name	'Set Filter Parameters command'
Event Message	'FILTER PARMS command: MID = 0x%04X, index = %d, filter = %d, N = %d, X = %d, O = %d'
Type	DEBUG

Cause	• The Packet Filter Table consists of an array of packet structures, with each element in the array representing one packet Message ID. • Each packet structure contains an array of filter structures, and each filter structure contains a set of filter parameters. Thus, each packet has multiple filters and each filter has unique filter parameters. • This event signals the successful execution of a command to set the filter parameters for a single filter entry in the Packet Filter Table. • The event text will indicate the packet Message ID, the Packet Filter Table index, the filter array index and the new filter parameters (N,X,O).

Table 127 Event ID 43 (Debug)

Event ID Number	43
Event Name	'Set Filename Type command (debug)'
Event Message	'DEST TYPE command: file table index = %d, filename type = %d'
Type	DEBUG
Cause	• This event signals the successful execution of a command to set the filename type for a Destination File Table entry. • The event text will indicate the Destination File Table index, and the new filename type.

Table 128 Event ID 45 (Debug)

Event ID Number	45
Event Name	'Set File Enable/Disable State command'
Event Message	'DEST STATE command: file table index = %d, file state = %d'
Type	DEBUG
Cause	• This event signals the successful execution of a command to set the enable/disable state for a Destination File Table entry. • The event text will indicate the Destination File Table index, and the new file enable/disable state.

Table 129 Event ID 47 (Debug)

Event ID Number	47
Event Name	'Set Filename Path command'
Event Message	'DEST PATH command: file table index = %d, pathname = '%s''

Type	DEBUG
Cause	• This event signals the successful execution of a command to set the path portion of the filename for a Destination File Table entry. • The event text will indicate the Destination File Table index, and the new filename path text.

Table 130 Event ID 49 (Debug)

Event ID Number	49
Event Name	'Set File Basename command (debug)'
Event Message	'DEST BASE command: file table index = %d, base filename = '%s''
Type	DEBUG
Cause	• This event signals the successful execution of a command to set the basename portion of the filename for a Destination File Table entry. • The event text will indicate the Destination File Table index, and the new file basename text.

Table 131 Event ID 51 (Debug)

Event ID Number	51
Event Name	'Set Filename Extension command'
Event Message	'DEST EXT command: file table index = %d, extension = '%s''
Type	DEBUG
Cause	• This event signals the successful execution of a command to set the extension portion of the filename for a Destination File Table entry. • The event text will indicate the Destination File Table index, and the new filename extension text.

Table 132 Event ID 53 (Debug)

Event ID Number	53
Event Name	'Set Max File Size command'
Event Message	'DEST SIZE command: file table index = %d, size limit = %d'

Type	DEBUG
Cause	• This event signals the successful execution of a command to set the file size limit for a Destination File Table entry. • The event text will indicate the Destination File Table index, and the new file size limit.

Table 133 Event ID 57 (Debug)

Event ID Number	57
Event Name	'Set Sequence Count command (debug)'
Event Message	'DEST COUNT command: file table index = %d, sequence count = %d'
Type	DEBUG
Cause	• This event signals the successful execution of a command to set the sequence count for a Destination File Table entry. • The event text will indicate the Destination File Table index, and the new sequence count.

Table 134 Event ID 59 (Debug)

Event ID Number	59
Event Name	'Close Destination File command'
Event Message	'DEST CLOSE command: file table index = %d'
Type	DEBUG
Cause	• This event signals the successful execution of a command to close the selected Destination File. • This command will succeed if the Destination File Table index is valid, regardless of whether there was an open file.

Table 135 Event ID 62 (Debug)

Event ID Number	62
Event Name	'Get File Info command'
Event Message	'GET FILE INFO command'

Type	DEBUG
Cause	• This event signals the successful execution of a command to get the file info diagnostic packet. • The event text will indicate the expected and actual packet size.

Table 136 Event ID 64 (Debug)

Event ID Number	64
Event Name	'Add Message ID to Packet Filter Table command'
Event Message	'ADD MID command: MID = 0x%04X, index = %d'
Type	DEBUG
Cause	• This event signals the successful execution of a command to add a new Message ID to the Packet Filter Table. • The Packet Filter Table must be loaded and have an unused entry available for adding the new Message ID. The new Message ID must not be zero and must not already exist in the table.

Table 137 Event ID 66 (Debug)

Event ID Number	66
Event Name	'Close All Destination Files command (debug)'
Event Message	'DEST CLOSE ALL command'
Type	DEBUG
Cause	• This event signals the successful execution of a command to close all open Destination Files. • This command will succeed regardless of whether there was an open file.

Appendix B   **Document Notes**

B.1   *Mission-Specific Conventions*

- This document presents selected *information in this dark orange font* that should be removed when tailoring this document for a mission.

- Command and Telemetry mnemonics are mission-specific. This document as delivered has "suggested" names that may or may not be used by the mission when the MOT creates the ground system database. In particular, the suggested names start with $sc_$cpu_DS which indicate a global setting for spacecraft, processor selection, and the DS subsystem. This has meaning if the mission has multiple spacecraft, each with a copy of cFE/CFS apps being executed, and/or multiple processors per spacecraft, each with a copy of cFE/CFS apps being executed. Most missions have neither and they do not prepend a $sc_$cpu_ selection to the front of the command name. However it is common for missions to differentiate the spacecraft subsystem commands from instrument commands by prepending a couple of characters (e.g. pw for power system) to all the command and telemetry names for that subsystem.

- The nomenclature of command and telemetry mnemonics is highly mission-specific, so this document does not attempt to include the actual command and telemetry database names in advance. For example, for MMS the telemetry mnemonics are defined in an RDL file for ASIST, and they will not exactly match the mnemonics in this Guide.

B.2   *Tailoring This Document*

This section is for anyone tailoring this Guide for a specific mission.

When tailoring this document for a particular mission:

- *Remove or replace text appearing in this dark orange font, with mission-specific values.*

- Review figures to be sure there are no conflicts with mission configurations. Edit figures with MS Visio or MS PowerPoint if necessary.

- Replace CFS defaults with mission defined values in Appendix A.

- Regenerate the table of contents (TOC); add Appendix page prefixes to TOC as needed. Table of Contents, Figures, and Tables can be updated automatically, but the letter prefix for Appendix pages must be added manually after update. To update all figure and table references in the document, when using the PC version of Word, select all, then choose F9.

- When complete, remove Appendix B.

This Guide is formatted using Microsoft Word styles. When adding new sections to the Guide, assign paragraphs to the styles shown in the table below. Center tables and figures horizontally on

the page. Use 15% grayscale in new table headings. For bullets in tables, assign the style shown below to set vertical spacing, then assign bullets using the bullet icons menu.

Table 138 Internal Document Styles

Type	Style to be Used	Justification
Chapter titles, subtitles, and subsections.	Heading 1 through 6	Left
First level bullets	"List Bullet 1" style.	Left
Second level bullets	"List Bullet 2" style	Left
Numbered lists	"Style List Enum 0"	Left
Names of code modules	Code	N/A
All text not otherwise tagged	"Body Text"	Full justification
Tables & Figures		
First row of tables	Table Header	Center
All other cells of tables	Table Cells	Left
Figure captions	Caption Figure	Center
Table captions	Caption Table	Center
Bullets in table cells	"Style Table Cells Vertical" + bullet	Left