CORE FLIGHT SOFTWARE SYSTEM (CFS)

LIMIT CHECKER (LC) APPLICATION

BUILD: 2.1.0.0

FSW VERSION DESCRIPTION DOCUMENT

RELEASE DATE: 8/29/2017

SIGNATURES

Approved by:

X SUSANNE STREGE Digitally signed by SUSANNE STREGE
Date: 2017.08.29 16:12:29 -04'00'

Susanne Strege/582
cFS Flight Software Product Development Lead

1.0 FSW VERSION DESCRIPTION

1.1 PURPOSE AND SUMMARY

The purpose of this build is to continue to refine the cFS Limit Checker (LC) application product. This build provides various bug fixes and enhancements, cosmetic fixes and a minor correction to the requirements document, minor clean up to remove old history logs and all references to Limit Checker eXtended (LCX), and new assert based unit tests.

The “eXtended” term for the LC application was added in the previous release to denote the extended capability for the application to detect stale data. This term has been deprecated in this release.

This document serves as the notification of the Build 2.1.0.0 release of the cFS LC application.

Limit Checker (LC) version 2.1.0.0 is compatible with cFE builds 6.5.0 and above and OSAL builds 4.2.0 and above.

1.2 NEW FUNCTIONALITY IN THIS VERSION

Table 1.2-1 identifies new FSW functionality that has been implemented and is integrated into this FSW version. Requirement references are included.

Table 1.2-1 – New Functionality in this Version

No.	FSB DCR # (or N/A)	Requirements	Functionality or Change Description
	N/A	N/A	None

Table 1.2-2 identifies changes to FSW functionality from a previously delivered FSW version and the DCRs and Trac Ticket numbers associated with these changes. See attachment 1 for a full listing of the DCRs and Trac Tickets included in this release.

Table 1.2-2 – Changes to Previously Delivered Functionality

Trac ticket references are proceeded with a ‘#’ character.

No.	FSB DCR or Trac # (or N/A)	Requirements	High Level Description of Functionality
1	3920	LC3002.4	LC3002.4 had not been implemented correctly in the code. When an actionpoint has passed the maximum number of consecutive fails, and its state is passive, LC issues an event message stating that the actionpoint has failed while passive. However, LC was not incrementing the global counter PassiveRTSExecCount that tracks how many RTS’s were not started due to LC’s state being passive or the actionpoint being passive itself. LC has been incrementing a counter (called PassiveAPCount) that tracks how many times the AP has failed while passive. To be compliant with the requirements LC shall also increment the PassiveRTSExecCount. The code has been updated to increment the PassiveRTSExecCount accordingly.
2	145597	LC9003	Misplaced else-case. In lc_app.c, in the function LC_TableInit, the final else-case was misplaced - it should be outside the bracket that follows it. This is evident because the else-case generates the event message "LC use of Critical Data Store disabled", which is impossible in its current location, but is guaranteed if the else-case is moved so that it's the else-case of "if ((LC_OperData.TableResults & LC_CDS_ENABLED) == LC_CDS_ENABLED)". The final else case that generates "LC use of Critical Data Store disabled" event message was moved so that it is the else case of "if ((LC_OperData.TableResults & LC_CDS_ENABLED) == LC_CDS_ENABLED)".
3	3972	LC2004	Values stored in the Watchpoint Results Table that caused a watchpoint to either trip false to true or true to false were not being stored correctly for values that are negative numbers or floating point decimal numbers. The watchpoints were comparing as intended but the value that tripped the watchpoint to change states was being stored in a uint32 variable. A uin8 DataType was added to the LC_WRTTransition_t structure to identify the values type that caused the watchpoint trip allowing the ground to union the uint32 value field.

No.	FSB DCR or Trac # (or N/A)	Requirements	High Level Description of Functionality
4	4046	N/A	LC - Table Definitions are Unclear. In theory, the number of watch points could be configured to an odd number. Let’s say 17 which are indexed 0 to 16. Which means 34 bits are needed in the HK packet to store the WP status. 8 bytes are allocated (because things are kept on a dword boundary). However when building the HK packet, LC indexes ahead to build the current byte. LC will pack status from the WP results table for 0 to 3 in byte 0, 4 to 7 in byte 1, 8 to 11 in byte 2, 12 to 15 in byte 3, and it should put 16 in byte 4. However it appears the code places 16 to 19 in byte 4 and then stops. Given the definition of the WP results table, 17 -19 does not exist and would index past the end of the WP results table which was allocated to be exactly 17 in size. Added conditional to verify.h to ensure these values are set correctly (must be multiples of 4 and 2 respectively).
5	4095	LC3006	LC3006 Item h) not implemented. Requirement LC3006 states: For each Actionpoint, the flight software shall maintain the following statistics in the dump-only Actionpoint Results Table: h) Total number of event messages sent This requirement is now implemented. A CumulativeEventMsgsSent counter has been added to the Actionpoint Results Table that increments for each message sent for that action point. Note: This counter will reset per LC4010/4011. Unit tests were updated to verify this counter is incremented/reset.
6	#5	N/A	LC application is not endian neutral. The LC application has conditional compilation based on the endianness of the target CPU, and it requires STRUCT_HIGH_BIT_FIRST or STRUCT_LOW_BIT_FIRST to be defined. As a result, it does not compile with the most recent OSAL which deprecates and (intentionally) does not define either of these ordering flags. The LC_GetSizedWPData() function has been modified so as to not depend on the target architecture's byte order, solving the build problem.

1.3 MISSING PLANNED FEATURES AND KNOWN PROBLEMS

Table 1.3-1 identifies the functions and known discrepancies that are absent from LC Build 2.1.0.0. Any workarounds that may apply are identified.

Information on currently open DCRs is available at:

http://tlserver3.gsfc.nasa.gov:7001/index.html.

Information on currently open Trac tickets is available at:

https://babelfish.arc.nasa.gov/trac/cfs_apps/report/1.

Note that these are restricted websites that requires a server account. Additional DCRs and/or Trac Tickets may have been submitted after preparation of this VDD. A cFS FM DCR report containing a listing of open DCRs and Trac tickets is available on request for customers who do not have access to the restricted servers. Please contact Susanne Strege, susie.strege@nasa.gov.

Table 1.3-1 – Functions absent from this Release

No.	FSB DCR or Trac #	Description	Reason for Absence	Affected Requirement or Component	Workaround	Planned Delivery
1	4117	LC - Add Trick Simulation Support (JSC Request)	Implementation is dependent on customer needs. Community input is needed.	N/A	None	Not Determined
2	#94	LC transitions Active APs to Passive When Application is in Passive Mode. During a GPM rehearsal there were several APs that were commanded "active" while the LC application state was in "passive" mode. Before operations could command the application state to "active" mode, some of the APs that were activated and had "tripped" causing the AP to transition back to passive mode. The purpose of changing a "tripped" APs state from active to passive is to prevent an RTS from getting initiated	Implementation is dependent on customer needs.	LC4005 No actual requirements to specify this behavior	None	Not Determined

		more than once. In "passive" mode, LC performs all limit tests as in "active" mode, but no stored command sequences are invoked as the result of AP failures. Having the AP's state transition while the application is in passive mode will make enabling APs with a low threshold while LC is in passive mode very difficult. The rational for this design feature (LRO heritage) needs to be clearly understood and documented. The LC user's guides (both doxygen and word/pdf) do not make this design feature clear. If no rational exists this design feature should be removed from LC.				
3	#80	LC does not support 64-bit integer or floats (doubles).	Implementation is dependent on customer needs.	LC2000	Patch is available in babelfish commit be65e0e	Not Determined

1.4 DEVELOPMENT TOOL VERSIONS ASSOCIATED WITH THIS FSW VERSION

Table 1.4-1 identifies the versions of development tools used to generate this FSW version:

Table 1.4-1 – Development Tool Versions Associated with this FSW Version

Tool Type.	Tool Name	Version Used
RTOS	BVTed with VxWorks 6.9, however, OSAL provides ability to use multiple OSes	6.9
Compiler	GNU	3.3.2
cFE	Core Flight Executive	6.5.0.0
OSAL	Operating System Abstraction Layer	4.2.0.0

2.0 DELIVERED PRODUCTS

Table 2-1 identifies the locations of FSW products relevant to this FSW Build. The version or date of the Build and where the product can be located are provided. Changes from a previous VDD are identified.

Table 2-1 – Delivered Products and their Locations

Software Element	Changed with this Version?	New Version or Date	Location
Executable for this build	Yes	2.1.0.0	Not applicable. Executables must be created for the specific mission/platform
Installation Procedures & Special Instructions **(See Section 3.0)**	No	3.1	See cFS Deployment Guide babelfish.arc.nasa.gov (in git system TOOLS master branch) and http://sourceforge.net/projects/coreflightexec
Source Code of this FSW Build	Yes	2.1.0.0	Tlserver3.ndc.nasa.gov. MKS label LC-ALL-Build2.1.0.0_AUG29-2017 babelfish.arc.nasa.gov (in git system lc_app_master branch) and http://sourceforge.net/projects/cfs-lc
FSW Build Plan	N/A	N/A	None
Annotated S/W Detailed Design Docs	No	N/A	fsb.gsfc.nasa.gov/cFS
Ground System T&C Database	Yes	2.1.0.0	Tlserver3.ndc.nasa.gov. MKS label LC-ALL-Build2.1.0.0_AUG29-2017 babelfish.arc.nasa.gov (in git system lc_app_master branch) and http://sourceforge.net/projects/cfs-lc

Software Element	Changed with this Version?	New Version or Date	Location
Ground System Scripts developed by FSB	Yes	2.1.0.0	Tlserver3.ndc.nasa.gov. MKS label LC-ALL-Build2.1.0.0_AUG29-2017 babelfish.arc.nasa.gov (in git system lc_app_master branch) and http://sourceforge.net/projects/cfs-lc
Simulator and Test Data Generator Software	No	N/A	None
Executable - Ground Tools associated with FSW (tools to build stored command loads, etc.)	No	N/A	None
Source Code - Ground Tools associated with FSW (tools to build stored command loads, etc.)	No	N/A	Perl scripts to generate ground database and build verification procedures from templates (see cFS Deployment Guide)
Unit Test Procedures	Yes	2.1.0.0	Tlserver3.ndc.nasa.gov. MKS label LC-ALL-Build2.1.0.0_AUG29-2017 babelfish.arc.nasa.gov (in git system lc_app_master branch) and http://sourceforge.net/projects/cfs-lc
Unit Test Data	Yes	2.1.0.0	Tlserver3.ndc.nasa.gov. MKS label LC-ALL-Build2.1.0.0_AUG29-2017 babelfish.arc.nasa.gov (in git system lc_app_master branch) and http://sourceforge.net/projects/cfs-lc
Unit Test Results	Yes	2016/11/22	Tlserver3.ndc.nasa.gov. MKS label LC-ALL-Build2.1.0.0_AUG29-2017 babelfish.arc.nasa.gov (in git system lc_app_master branch) and http://sourceforge.net/projects/cfs-lc

Software Element	Changed with this Version?	New Version or Date	Location
FSW Make Files	No	2.1.0.0	Tlserver3.ndc.nasa.gov. MKS label LC-ALL-Build2.1.0.0_AUG29-2017 babelfish.arc.nasa.gov (in git system lc_app_master branch) and http://sourceforge.net/projects/cfs-lc
Linker & Compiler Configuration Files	No	2.1.0.0	Tlserver3.ndc.nasa.gov. MKS label LC-ALL-Build2.1.0.0_AUG29-2017 babelfish.arc.nasa.gov (in git system lc_app_master branch) and http://sourceforge.net/projects/cfs-lc
Requirements version (from MKS)	Yes	1.2	MKS label – version 1.2

3.0 INSTALLATION PROCEDURES

Table 3-1 identifies the nominal FSW Installation Procedure(s) for this FSW Build onto the intended target system (including the commercial applications used and the configuration settings). The procedure version identifier, the date of the procedure and where it can be located are also provided.

Table 3-1 FSW Installation Procedure(s)

Destination (Target System)	Filename	Version and Date	Location
N/A	See cFS Deployment Guide	Version 3.1	Available with cFE open source release: http://sourceforge.net/projects/coreflightexec/ babelfish.arc.nasa.gov (in git system TOOLS master branch) and on Tlserver3.ndc.nasa.gov

4.0 CONFIGURATION SUMMARY AND VERSION IDENTIFICATION

LC Build 2.1.0.0 can be found on tlserver3.ndc.nasa.gov, sourceforge: http://sourceforge.net/projects/cfs-lc, and babelfish.arc.nasa.gov (in git system lc_app_master branch). Verification of the version can be done by sending an LC NOOP command which produces an event message containing the version information. In addition, the initialization event message generated during the application startup provides the version information.

5.0 SOFTWARE COPYRIGHT NOTICE

ACRONYMS

ACS	Attitude Control System
C&DH	Command and Data Handling
cFE	core Flight Executive
cFS	core Flight Software System
CM	Configuration Management
COTS	Commercial Off-The-Shelf
DCR	Discrepancy/Change Request
ETU	Engineering Test Unit
FSB	Flight Software Branch
FSW	Flight Software
I&T	Integration & Test
LC	Limit Checker
OSAL	Operating System Abstraction Layer
RTOS	Real-Time Operating System
T&C	Telemetry and Command
URL	Universal Resource Locator
VDD	Version Description Document

ATTACHMENT 1 – CFS LIMIT CHECKER BUILD 2.1.0.0 DCRS/TRAC TICKETS

Trac ticket references are proceeded with a '#' character.

No.	DCR/Trac Ticket #	Description	Type	Priority	State	Date Reported	Build Target
1	#39	Allow C99 code in APPS	defect	minor	Test Complete	01/28/2016	2.1.0.0
2	#5	LC application is not endian neutral. The LC application has conditional compilation based on the endianness of the target CPU, and it requires STRUCT_HIGH_BIT_FIRST or STRUCT_LOW_BIT_FIRST to be defined. As a result, it does not compile with the most recent OSAL which deprecates and (intentionally) does not define either of these ordering flags. The LC_GetSizedWPData() function has been modified so as to not depend on the target architecture's byte order, solving the build problem.	defect	major	Test Complete	03/06/2015	2.1.0.0
3	3920	GPM-IVV-1251 - Incorrect implementation of LC3002.4. LC3002.4 had not been implemented correctly in the code. When an actionpoint has passed the maximum number of consecutive fails, and its state is passive, LC issues an event message stating that the actionpoint has failed while passive. However, LC was not incrementing the global counter PassiveRTSExecCount that tracks how many RTS's were not started due to LC's state being passive or the actionpoint being passive itself. LC has been incrementing a counter (called PassiveAPCount) that tracks how many times the AP has failed while passive. To be compliant with the requirements LC shall also increment the PassiveRTSExecCount. The code has been updated to increment the PassiveRTSExecCount accordingly.	defect	moderate	Test Complete	10/04/2011	2.1.0.0
3	3972	GPM-IVV-1309 - LC - Incorrect value representations being stored in WP ResultsTable. Values stored in the Watchpoint Results Table that caused a watchpoint to either trip false to true or true to false were not being stored correctly for values that are negative numbers or floating point decimal numbers. The watchpoints were comparing as intended but the value that tripped the watchpoint to change states was being stored in a uint32 variable. A uin8 DataType was added to the LC_WRTTransition_t structure to identify the values type that caused the watchpoint trip allowing the ground to union the uint32 value field.	defect	moderate	Test Complete	01/24/2012	2.1.0.0
4	4046	LC - Table Definitions are Unclear. In theory, the number of watch points could be configured to an odd number. Let's say 17	defect	moderate	Test Complete	07/17/2012	2.1.0.0

		which are indexed 0 to 16. Which means 34 bits are needed in the HK packet to store the WP status. 8 bytes are allocated (because things are kept on a dword boundary). However when building the HK packet, LC indexes ahead to build the current byte. LC will pack status from the WP results table for 0 to 3 in byte 0, 4 to 7 in byte 1, 8 to 11 in byte 2, 12 to 15 in byte 3, and it should put 16 in byte 4. However it appears the code places 16 to 19 in byte 4 and then stops. Given the definition of the WP results table, 17 -19 does not exist and would index past the end of the WP results table which was allocated to be exactly 17 in size. Added to the doxygen comments regarding that these must be multiples of 4 and 2 respectively. Added conditional to verify.h to ensure these values are set correctly					
5	4095	IV&V CFS BVT Findings - LC3006 Item h) Not Implemented. Requirement LC3006 states: For each Actionpoint, the flight software shall maintain the following statistics in the dump-only Actionpoint Results Table: h) Total number of event messages sent This requirement is now implemented. A CumulativeEventMsgsSent counter has been added to the Actionpoint Results Table that increments for each message sent for that action point. Note: This counter will reset per LC4010/4011. Unit tests were updated to verify this counter is incremented/reset.					
5	145597	Misplaced else-case. In lc_app.c, in the function LC_TableInit, the final else-case was misplaced - it should be outside the bracket that follows it. This is evident because the else-case generates the event message "LC use of Critical Data Store disabled", which is impossible in its current location, but is guaranteed if the else-case is moved so that it's the else-case of "if ((LC_OperData.TableResults & LC_CDS_ENABLED) == LC_CDS_ENABLED)". The final else case that generates "LC use of Critical Data Store disabled" event message was moved so that it is the else case of "if ((LC_OperData.TableResults & LC_CDS_ENABLED) == LC_CDS_ENABLED)".	defect	moderate	Test Complete	05/24/2016	2.1.0.0
6	145598	Unused / Unneeded Variables in lc_app.c	defect	minor	Test Complete	05/24/2016	2.1.0.0
7	145731	Unit Test Makefile: Replace hard-coded paths with environment variables set in setvars.sh	enhancement	minor	Test Complete	07/22/2016	2.1.0.0

8	145735	Implement UT-Assert unit tests for the LCX application	enhancement	minor	Test Complete	07/29/2016	2.1.0.0
9	145736	Move function prototypes from .c files to .h file. In lc_app.c, lc_cmds.c, lc_action.c, and lc_watch.c, the function prototypes should be moved to the corresponding .h file, so the functions will be accessible by unit tests.	enhancement	minor	Test Complete	08/01/2016	2.1.0.0
10	145919	LC - CFE_EVS_SendEvent Format Warnings	defect	minor	Test Complete	10/24/2016	2.1.0.0
11	145931	LC - UT-Assert Unit Tests - Code Walkthrough Updates	enhancement	minor	Test Complete	10/26/2016	2.1.0.0
12	145933	Requirements document format issues	defect	major	Test Complete	10/27/2016	2.1.0.0
13	146007	Remove copyright symbol from comment blocks and MKS history.	defect/ enhancement	minor	Test Complete	01/22/2017	2.1.0.0
14	146194	Add padding between new variable in WRTTransition_t type. In the LC_WRTTransition_t type, a new data type variable was added as a uint8. Since this follows a uint32, the compiler will explicitly pad the next variable (CFE_TIME_SysTime_t Timestamp;) to a 32-bit boundary. In lc_tbl.h , added 3 byte padding to LC_WRTTransition_t after uint8 "DataType" in order to align CFE_TIME_SysTime_t Timestamp to the next 32-bit boundary.	enhancement	minor	Test Complete	06/30/2017	2.1.0.0
15	146229	Requirement 3002.4 is incorrectly worded. This requirement should reference the Passive RTS Execution Counter and does not. In addition the doxygen comments for the RTS Execution Counter were mission information on when the counter shall increment.	defect	minor	Test Complete	08/18/2017	2.1.0.0