Code 582
Flight Software Systems Branch

CFS HS Requirements

Aug 1, 2011

ID	Summary
7417	CFS HS Requirements Document

SR_Contains:

ID	ReqID	Text	Rationale	Heritage_Reference
7419		CFS Health and Safety (HS) Requirements		
7421		1.0 Introduction 1.1 Document Purpose The Core Flight Software System (CFS) Health and Safety (HS) Application will be developed by the Flight Software Branch (FSB) of the Software Engineering Division (SED). The purpose of this requirements specification is to define the requirements to be satisfied by the Health and Safety Application. This application is developed for re-use. For this reason, several nomenclatures are used in this document to identify configurations for a mission. The CFS is specified as a multi-platform product. Mission-specific features and customization requirements which are applicable for all platforms are tagged with <MISSION_DEFINED>. Platform-specific features and customizations requirements are tagged with either “<PLATFORM_DEFINED>” or “<OPTIONAL>.” Additional nomenclature is used along with the tag to specify a CFS default value for the platform-specific feature: “<PLATFORM_DEFINED, Default_Value>”. Reference platforms (single processor and multi-processor architectures) are defined to supply the default CFS application configuration. These configurations define the “maximum” CFS Application deployments such that any refined deployment is a subset of a reference platform. 1.2 Document Scope The scope of this document is limited to the specification of requirements for the Health and Safety Software requirements. These include functional, performance, qualification, and design requirements. 1.3 Document Organization		

This document is organized into three additional sections and several appendices.

Section 2 gives the Health and Safety context.

Section 3 documents the Health and Safety system design decisions and constraints.

Section 4 contains the Health and Safety functional and performance requirements.

Appendix A contains a list of abbreviations and acronyms used in this document.

1.4 Relevant Documents

1.4.1 Parent Documents

CFS Health and Safety Application Heritage Analysis 582-2007-028

1.4.2 Reference Documents

1. Operating System Abstraction Layer (OSAL) Library

2. cFE Application Developer's Guide 582-2007-001

3. cFE User's Guide

2.0 CFS Health and Safety Application Context

The Health and Safety (HS) application is responsible for

- Monitoring critical applications,
- Monitoring event messages,
- Managing the watchdog,
- Verifying CPU aliveness

- Reporting execution counters.

The context diagram (Figure 2.0) shows the CFS HS interfaces.

Inputs to the Health and Safety Application include:

1) Commands to the Health and Safety Application

2) App Info from cFE ES

3) Wake-up message

4) HK Request from the Scheduler Application

5) Event Messages

6) Table management requests

Outputs from the Health and Safety application include:

1) Health and Safety Application housekeeping message

2) HS Event messages

3) CPU Aliveness indicator to the UART

4) Reset Requests

5) Watchdog servicing

Tables used by the Health and Safety Application include:

1) Critical Application Table – defines the applications that are required to check-in

2) Execution Counter Table – defines the items (applications and child tasks) for which HS needs to maintain counters

3) Critical Event Table – defines critical events

Note that during the Heritage Analysis of the HS Application, it was determined that HS should also perform the following:

- Monitor and report CPU hogging of the processor
- Report CPU utilization per task

It is not clear that these functions can be made generic and where they should go (they may be better suited for the cFE or possibly need to be made a mission specific application). For that reason, they are currently omitted from the HS requirements document. The requirements will be addressed at a later date and, if necessary, added to the HS requirements document.

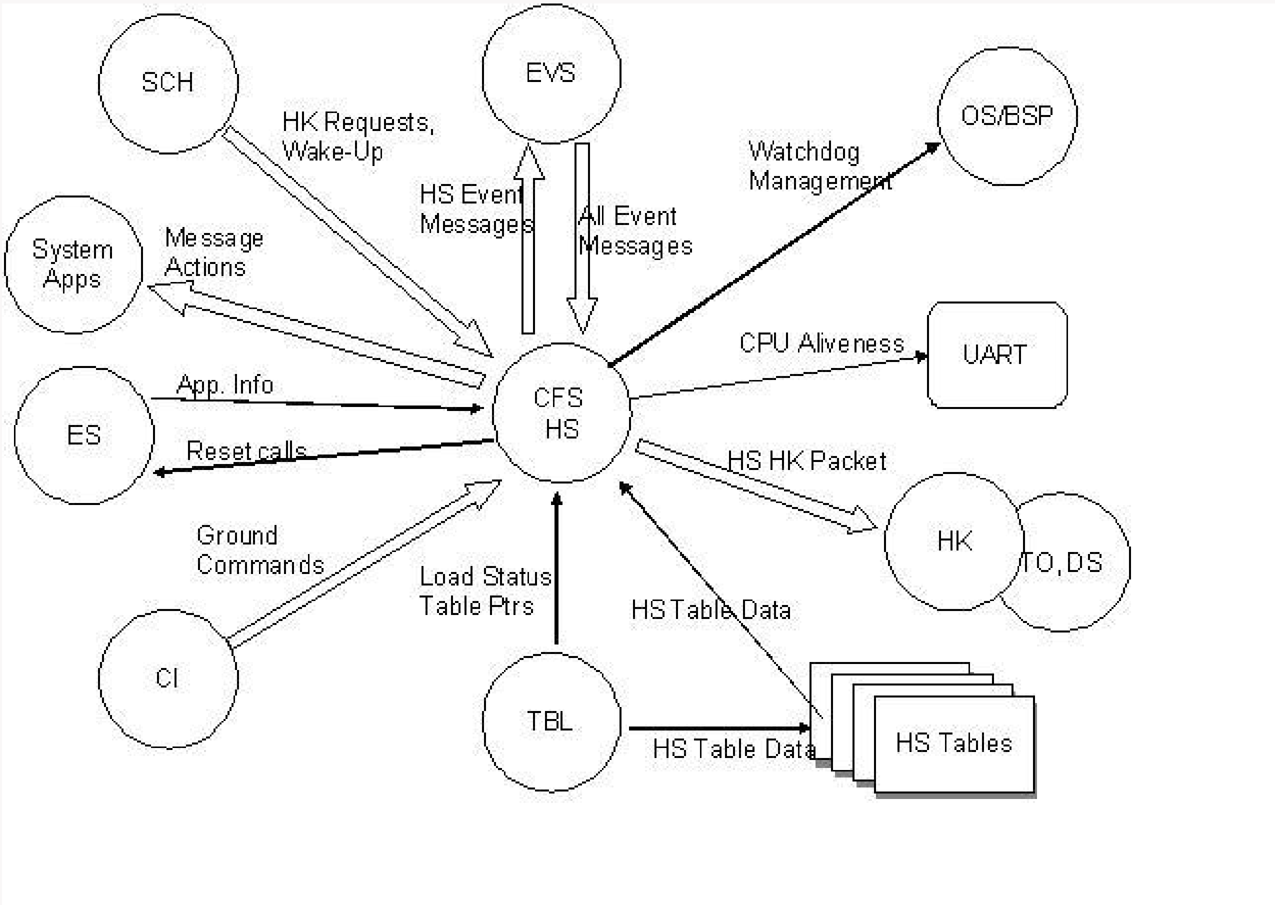

Figure 2.0 – CFS HS Context

2.1 Assumptions

The following list summarizes the assumptions made by the CFS Health and Safety Application:

- cFE API and OSAL are being used

- OS/BSP provides the interface to the watchdog timer. The timeout value is specified within the OS/BSP. The BSP will allow for the timer to be large enough to allow the system to startup without the watchdog expiring. The OS/BSP will provide the ability to set a watchdog timeout. Note that the OS provides the ability to set the watchdog timeout without having to know what the timeout value should be (default value defined in OS BSP) , thus removing coupling between HS and OS/BSP.

- Applications must use the cFE ES “Run Loop” paradigm in order for HS to perform the Critical Application Check-in.

- Critical Application Monitoring is only done on Applications. If an application spawns child tasks, it is the application’s responsibility to monitor the child task(s).

- In order to maintain the Execution Counters, Items (Application main tasks and child tasks) must use the appropriate cFE APIs.

3.0 Design Specifications

The Health and Safety Application’s requirements and design are based on the results of the CFS heritage analysis effort. The results of the heritage analysis are documented in the CFS Health and Safety Application Heritage Analysis document.

The HS Application uses the counters maintained by the cFE’s Executive Services core application. Applications that are to be monitored by HS are required to use the CFE_ES_RunLoop function. For Critical Application Monitoring, HS uses the Critical

Application Table to define the applications that need to be monitored. Each element in the table defines a monitor point, thus an application can be repeated in the table multiple times if a different action and/or threshold is desired. Each HS cycle, HS processes the Critical Application table and determines if any of the entries in the table is not executing, taking the appropriate action if the HS determines that the Application is, in fact, not executing.

The Critical Events Table contains a list of events that should be monitored and the actions that should be taken upon receipt of that event.

OS/BSP functions are used to service the watchdog. If HS doesn't execute, the watchdog will expire causing the processor the reset.

HS provides the capability to report the desired string to the UART.

3.1 Design Constraints

Since the cFE/CFS architecture provide the ability to add and remove applications at runtime, HS relies on tables to define which applications and events to monitor. In addition, tables are used to define the execution counters to report.

Currently the cFE does not support providing execution counters for device drivers or ISRs, therefore HS is only able to report execution counters for applications (i.e. main tasks) and child tasks.

4.0 Subsystem Requirements

7425	CFS-510	The CFS shall monitor application execution and take the appropriate action if one or more critical applications fails to execute.		LRO,SDO
7427	CFS-511	The CFS shall monitor all event messages, looking for critical events and taking the appropriate action		LRO,SDO
7429	CFS-512	The CFS shall maintain the watchdog timer.		LRO,SDO
7431	CFS-513	The CFS shall monitor and report the CPU Aliveness.		LRO,SDO

7423

5.0 Detailed Requirements

5.1 Basic Requirements

The following requirements are basic requirements of Health and Safety. Some of them are included here to avoid repeating these requirements for each applicable requirement.

7433	HS1000	Upon receipt of a No-Op command, HS shall increment the HS Valid Command Counter and generate an event message.	Debug command to verify application is alive.	LRO, SDO
7435	HS1001	Upon receipt of a Reset command, HS shall reset the following housekeeping variables to a value of zero: a) Valid Ground Command Counter b) Ground Command Rejected Counter.	Important for testing and on-orbit flight operations in order to start with a “clean slate”.	LRO, SDO
7437	HS1002	For all HS commands, if the length contained in the message header is not equal to the expected length, HS shall reject the command and issue an event message.	Basic command verification in the event of SEU or memory corruption.	LRO, SDO
7439	HS1003	If HS accepts any command as valid, HS shall execute the command, increment the HS Valid Command Counter and issue an event message.	Operators require feedback on command execution.	LRO, SDO
7441	HS1004	If HS rejects any command, HS shall abort the command execution, increment the HS Command Rejected Counter and issue an error event message.	Operators require feedback on command execution.	LRO, SDO
7539		**5.2 Critical Application Monitoring** The Health and Safety Application is responsible for ensuring that the critical applications, as defined in a Critical Application Table are executing. The cFE Executive Services (ES) provides the API, CFE_ES_RunLoop that each critical application must use in order for HS to monitor its execution. Note that the HS application treats each entry in the Critical Application Table as a separate entry, therefore, an application can be defined more than one time using more than one entry in the table. This can be useful for defining multiple actions.		
7443	HS2000	The HS Application shall verify that each application defined in the Critical Application Table is executing.	HS uses a table to define the critical application. Tables are used so that the list can be configured for a mission and can be easily changed.	LRO, SDO

7445	HS2000.1	If the entry indicates that the application is a cFE Core Application and it has not executed for the corresponding table-defined number of HS execution cycles, HS shall perform one of the table-defined actions a) cFE Processor Reset b) Send an Event message c) Send a Software Bus Message d) Perform No Action	Not able to individually start a cFE core application, therefore, need to perform a cFE Processor Reset. . Note that the Software Bus message can be any message, however, the intent it to allow the starting of an RTS (without tying HS to SC).	LRO
7447	HS2000.1.1	If the action is to perform a cFE Processor Reset and the Number of cFE Processor Resets is less than the <PLATFORM_DEFINED> Max Number of cFE Processor Resets , HS shall a) Increment the Number of cFE Processor Resets b) Set the Watchdog servicing flag to False c) Command the cFE Processor Reset.	Note that the watchdog flag is set to false in the event that the cFE Processor reset can't be performed.	LRO
7449	HS2000.1.2	If the action is to perform a cFE Processor Reset and the Number of cFE Processor Resets is greater-than-or-equal-to the <PLATFORM_DEFINED> Max Number of cFE Processor Resets , HS shall a) Send an event message.	No cFE Processor Reset will be performed. Need to prevent an infinite reset loop.	New
7451	HS2000.2	If the entry indicates that the application is not a cFE Core Application and it has not executed for the corresponding table-defined number of HS execution cycles, HS shall execute one of the table-defined actions: a) Restart the Application (that failed to check-in) b) cFE Processor Reset c) Send an Event Message d) Send a Software Bus Message e) Perform No Action	Note that each entry has a corresponding HS execution counter. This represents the number of HS execution cycles before the Application is declared "not there". Note also that an application can have more than one entry (could have one entry to restart the app and a second entry to perform a cFE) Processor	LRO, SDO

			reset. More than one application restart can be performed in one HS cycle. Note that the Software Bus message can be any message, however, the intent it to allow the starting of an RTS (without tying HS to SC).	
7453	HS2000.2.1	If the action is to perform a cFE Processor Reset and the Number of cFE Processor Resets is less than the <PLATFORM_DEFINED> Max Number of cFE Processor Resets , HS shall a) Increment the Number of cFE Processor Resets b) Set the Watchdog servicing flag to False c) Command the cFE Processor Reset.	Note that the watchdog flag is set to false in the event that the cFE Processor reset can't be performed.	LRO, SDO
7455	HS2000.2.2	If the action is to perform a cFE Processor Reset and the Number of cFE Processor Resets is greater-than-or-equal-to the <PLATFORM_DEFINED> Max Number of cFE Processor Resets , HS shall a) Send an event message.	No cFE Processor Reset will be performed. Need to prevent an infinite reset loop.	New
7457	HS2000.2.3	If the action is to perform an Application Restart, HS shall disable the entry in the Critical Application Table.	If an application restart does not fix the problem, disable the monitoring in order to prevent an infinite loop.	New
7459	HS2000.3	If the entry in the table references an unresolvable application (i.e. not registered with cFE), HS shall issue an event message	This is the case where an application is unknown to the cFE – maybe because of a misspelled application.	New

7461	HS2001	Upon receipt of an Enable Critical Application Monitoring Command, HS shall a) Enable all entries in the Critical Application Table b) Execute the Critical Application Table.	Enables maintenance to be done on a critical app (eg. Disable Critical Application Monitoring, modify/load critical app, Enable Critical App Monitoring. Note that nothing is preserved between a Disable command and an Enable command.	New
7463	HS2002	Upon receipt of a Disable Critical Application Monitoring Command, HS shall stop processing the Critical Application Table.	Enables maintenance to be done on a critical app.	New
7465	HS2003	HS shall support up to <PLATFORM_DEFINED> critical applications.	Bounds the Critical Application Table size.	LRO
7467	HS2004	Upon receipt of a Critical Application Table update indication, HS shall validate the Critical Application Table by validating the action	Validate the table for gross errors. Only able to valid the action field (all others are runtime checks)	LRO
7469	HS2004.1	If the Critical Application Table fails validation, HS shall issue an event message.	Can verify gross errors in table. If errors exist, HS can still perform other monitoring (just not critical application monitoring).	New
7541		**5.3 Execution Counter Management**		

7471	HS3000	The HS Application shall maintain the Execution Counters defined in the Execution Counter Table for up to <PLATFORM_DEFINED> number of Items.	Users define the Items that HS shall maintain counters for. Items can be Application (main task) and child tasks Note that the items must use the appropriate cFE APIs. Note that a value of zero can be used if a mission does not want to include Execution counters.	SDO
7473	HS3000.1	If the Item contained in the Execution Counter Table is unknown, HS shall a) Set the Execution Counter value for that entry to 0x’FFFFFFFF’	Either that application didn’t initialize properly and just exited its run loop or the table contained an invalid item reference (eg. invalid application, invalid task). Reporting ‘FFFFFFFF’ gives the ground an indication that something isn’t correct.	New
7475	HS3001	Upon receipt of a Execution Counter Table update indication, HS shall validate the Execution Counter Table.	Validate the table for gross errors.	LRO
7477	HS3001.1	If the Execution Counter Table fails validation, HS shall issue an event message.	Can verify gross errors in table. If errors exist, HS can still perform other monitoring.	New

7543		**5.4 Watchdog Management** HS manages the servicing of the watchdog timer. Every HS cycle it will check the HS maintained Watchdog Status Flag to determine if the watchdog should be serviced.		
7479	HS4000	During each HS execution cycle, HS shall check the status of the Update Watchdog Timer Flag.	The Watchdog Timer Flag is the mechanism used by all of the HS functions to indicate whether to service the watchdog timer or not.	LRO, SDO
7481	HS4000.1	If it is set to TRUE, HS shall service the Watchdog Timer.	Flag set to TRUE on initialization.	LRO, SDO
7483	HS4000.2	If it is set to FALSE, HS shall not service the Watchdog Timer.	Other applications can still service the watchdog.	LRO,SDO
7545		**5.5 Critical Event Monitoring** HS provides the ability to monitor event messages and take the table defined action. Note that events received prior to HS executing (based on the startup synch) will not be able to be monitored by HS.		
7485	HS5000	The HS Application shall compare each received event message with the events defined in the Critical Event Table for up to <PLATFORM_DEFINED> critical events.	HS subscribes to all events. HS uses a table to define the critical events. Note that in order to make an event unique, the application and the event ID are required.	LRO, SDO

7487	HS5000.1	If the event received is defined in the Critical Event Table, HS shall execute one of the following table-defined actions: a) Restart Application that generated the Event b) Perform cFE Processor Reset c) Delete the Application that generated the Event d) Send a Software Bus Message e) Perform No Action	The user defines the action that is appropriate if the critical event is received. Note that the delete is included in the case where an application might want to "get fancy" and stop running an application based on some event. Note that LRO allowed for RTSs to be started as one of the actions. CFS does not include this – can be done by LC using tlm and RTSs. Note that the Software Bus message can be any message, however, the intent it to allow the starting of an RTS (without tying HS to SC).	LRO, SDO (although deleting the specified Application was not present in either heritage mission)
7489	HS5000.1.1	If the action is to perform a cFE Processor Reset and the Number of cFE Processor Resets is less than the <PLATFORM_DEFINED> Max Number of cFE Processor Resets , HS shall a) Increment the Number of cFE Processor Resets b) Set the Watchdog servicing flag to False c) Command the cFE Processor Reset.	Note that the watchdog flag is set to false in the event that the cFE Processor reset can't be performed.	New
7491	HS5000.1.2	If the action is to perform a cFE Processor Reset and the Number of cFE Processor Resets is greater-than-or-equal-to the <PLATFORM_DEFINED> Max Number of cFE Processor Resets , HS shall send an event message.	Need to present an infinite reset loop.	New
7493	HS5000.2	If the Application defined in the Critical Event Counter Table is unknown, HS shall increment Critical Event Table Invalid/Unknown Apps counter	Informs ground that there is an entry in the table with an unknown application.	New

7495	HS5001	Upon receipt of an Enable Critical Event Monitor Command, HS shall a) Set the Enable Critical Event Monitoring to Enabled b) Begin processing the Critical Event Table.		New
7497	HS5002	Upon receipt of a Disable Critical Event Monitor Command, HS shall a) Set the Enable Critical Event Monitoring to Disabled b) Stop executing the Critical Event Table.	Useful for making table updates.	New
7499	HS5003	HS shall support up to <PLATFORM_DEFINED> critical events.		LRO
7501	HS5004	Upon receipt of a Critical Event Table update indication, HS shall validate the Critical Event Table by validating the action	Validate the table for gross errors. Only able to valid the action field (all others are runtime checks)	LRO
7503	HS5004.1	If the Critical Event Table fails validation, HS shall issue an event message.	Can verify gross errors in table. If errors exist, HS can still perform other monitoring (just not critical event monitoring).	New
7547		**5.6 CPU Management** Health and Safety is responsible for providing an CPU aliveness indicator.		
7505	HS6005	During each HS execution cycle, HS shall send a <PLATFORM_DEFINED> character(s) to the UART port every <PLATFORM_DEFINED> second(s).	Heartbeat reported to the UART gives an indication that the system is running.	SDO prints the 'nm' character set to the uart when the OS_DEBUG_LEVEL is set to 3 or higher. LRO and Swift print a dot to the terminal regardless of debug level
7507	HS6006	Upon receipt of an Enable CPU Aliveness Indicator Command, HS shall begin sending the <PLATFORM_DEFINED> heartbeat character(s) to the UART port.	Enable the heartbeat indicator.	New
7509	HS6007	Upon receipt of a Disable Aliveness Indicator Command, HS shall stop sending the <PLATFORM_DEFINED> heartbeat character (s) to the UART port.	Useful in flight since mission may not want messages being sent to the UART.	New

9710	HS6008	HS shall report peak CPU utilization during an interval of <PLATFORM_DEFINED> HS execution cycles over the last <PLATFORM_DEFINED> intervals.	Allows a mission to track CPU performance. Peak is the highest average number over the given time period	
9712	HS6009	HS shall report average CPU utilization over the last <PLATFORM_DEFINED> intervals of <PLATFORM_DEFINED> HS execution cycles each.	Allows a mission to track CPU performance.	
9714	HS6010	If the CPU utilization exceeds <PLATFORM_DEFINED> % for <PLATFORM_DEFINED> intervals and CPU Utilization Monitoring is enabled then HS shall issue an event message	Alerts the ground to CPU hogging. This event can be turned on or off	
9716	HS6011	Upon receipt of an Enable CPU utilization monitoring, HS shall begin monitoring CPU utilization.	Allows mission to turn CPU utilization on or off	
9718	HS6012	Upon receipt of an Disable CPU utilization monitoring, HS shall stop monitoring CPU utilization	Allows mission to turn CPU utilization on or off	
7549		**5.8 Miscellaneous Commanding**		
7511	HS7000	Upon receipt of a Reset Processor Resets Command, HS shall set the number of cFE Processor Resets commanded by HS to zero.	HS keeps track of the number of cFE Processor Resets it performs in order to avoid an infinite reset loop. Resetting this count allows HS to continue to perform resets up to the internally set Max.	New

7513	HS7001	Upon receipt of a Set Max Processor Resets Command, HS shall set the Maximum number of cFE Processor Resets commanded by HS to the Command-specified value.	Allows the ground to modify the default value specified in a configuration file without having to recompile. Primarily used in order to be consistent with cFE. This limit is different than the limit that the cFE maintains.	New
7553		**5.9 Status Reporting**		
7515	HS7100	HS shall generate a housekeeping message containing the following: a) Valid Ground Command Counter b) Ground Command Rejected Counter c) Critical Application Monitoring status (enable/disable) d) Critical Application Monitoring status per table entry (enable/disable) e) Number of CFE Processor resets (commanded by HS) f) Maximum number of CFE Processor resets g) Critical Event Monitoring status (enabled/disabled) h) Count of Monitored Event Messages i) CPU Aliveness Indicator (enabled/disabled) j) Execution Counter, for each table entry k) Number of Invalid/Unknown Apps contained in Critical Event Table. l) Peak CPU Utilization m) Average CPU utilization n) CPU Utilization Monitoring Enabled/Disabled	Housekeeping telemetry to indicate basic HS status.	LRO, SDO
7551		**5.10 Initialization Requirements** The following are the requirements associated with Health and Safety on an Application restart, cFE Processor Reset or a cFE Power-on Reset		
7517	HS8000	Upon cFE Power On Reset, HS shall initialize the following data to Zero (or the value specified for the item below) : a) Valid Ground Command Counter b) Ground Command Rejected Counter c) Monitor Critical Applications to <PLATFORM_DEFINED> d) Critical Application Monitoring status per Application Enabled e) Monitor Critical Events to <PLATFORM_DEFINED> f) CPU Aliveness Indicator to <PLATFORM_DEFINED> g) Watchdog Timer Flag set to TRUE h) Set the Watchdog Timer to <PLATFORM_DEFINED> value i) Maximum number of CFE Processor resets to <PLATFORM_DEFINED> value j) Number of cFE Processor Resets (commanded by HS) k) Number of Invalid/Unknown Apps contained in Critical Event Table l) Peak CPU Utilization	.	Derived

		m) Average CPU utilization n) CPU Utilization Monitoring Enabled/Disabled to <PLATFORM_DEFINED>		
7519	HS8001	Upon cFE Processor Reset or HS Application Restart, HS preserves the following: a) Number of cFE Processor Resets (commanded by HS) b) Maximum number of cFE Processor Resets.	Need to preserve the cFE Processor reset default and current number across a reset.	New
7521	HS8002	Upon any Initialization, HS shall subscribe to all event messages.	HS needs to subscribe to all event messages in support of Critical Event Processing.	LRO, SDO
7523	HS8003	Upon any Initialization, HS shall load the Critical Application Table.	Need default values.	LRO
7525	HS8003.1	If the Critical Application Table fails validation, HS shall issue an event message and disable Critical Application Monitoring.	Can verify gross errors in table. If errors exist, HS can still perform other monitoring (just not critical application monitoring).	New
7527	HS8004	Upon any Initialization, HS shall load the Critical Event Table.	Need default values.	LRO
7529	HS8004.1	If the Critical Event Table fails validation, HS shall issue an event message and disable Critical Event Monitoring.	Can verify gross errors in table. If errors exist, HS can still perform other monitoring (just not critical event monitoring).	New
7531	HS8005	Upon any Initialization, HS shall load the Execution Counter Table.	Need default values.	New

7533	HS8005.1	If the Execution Counter Table fails validation, HS shall: a) Issue an event message b) Report 0x‘FFFFFFFF’ for all <PLATFORM_DEFINED> items in the table.	If the file does not exist to populate the Execution Counter Table (or fails validation), HS should continue to execute other functions and report an identifiable number (i.e. x’ffffffff’) for entries using the configuration parameter.	New
7535	HS8006	Upon any initialization, HS shall wait until the cFE startup synch has been received indicating all Applications have started.	The cFE provides indication that all applications that it was told to start have started. Having HS wait on that signal avoids HS reporting things like critical app check-in failure when in reality the app just has not been started yet.	New
7537	HS8006.1	If the startup-synch is not received in <PLATFORM_DEFINED> seconds, HS shall begin processing.	If HS doesn’t get the startup synch within some timeout value, HS should start executing.	New

7555

Appendix A Acronyms and Abbreviations

This appendix contains the list of abbreviations and acronyms for the CFS Health and Safety Application used in this document. These abbreviations and acronyms adhere to FSB standard usage.

- **Health and Safety –** Application responsible for monitoring critical applications, monitoring event messages, managing the watchdog, monitoring and reporting CPU utilization and reporting execution counters.
- **Critical Application –** An application that is defined by a user to be “critical” in that it must be executing. If it is not executing then HS takes the action defined in the Critical Application Monitoring Table.
- **Critical Application Monitoring Table –** Contains entries for the applications that should be monitored and the actions that should be taken in the event that the application does not check-in.
- **Critical Event –** Event message that is defined by a user to be “critical” in that an action needs to be taken. The Critical Events are defined in the Critical Event Table.
- **Critical Event Table –** Table which defines which events need to be monitored and what actions to take if received.
- **Operating System Abstraction Layer (OSAL)** – Set of functions supplied as part of the cFE that isolate the calling application from operating system dependencies.
- **HS** – Health and Safety Application.
- **Table** – A section of memory partitioned for a logical group of data. Usually fixed in size, at a fixed address, and created on system startup. Can be modified, but not normally deleted.