



CFS SCH Requirements

Aug 1, 2011

ID		Summary		
2890		CFS SCH Requirements Document		
SR Contains:				
ID	ReqID	Text	Rationale	Heritage_Reference
2892		<h2>CFS Scheduler (SCH) Requirements</h2> <h3>1.0 Introduction</h3> <h4>1.1 Document Purpose</h4> <p>The Core Flight Software System (CFS) Scheduler (SCH) 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 Scheduler Application. This application is developed for re-use. For this reason, several nomenclatures are used in this document to identify configurations for a mission.</p> <p>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.</p> <h4>1.2 Document Scope</h4>		

The scope of this document is limited to the specification of requirements for the Scheduler 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 Scheduler context.

Section 3 documents the Scheduler system design decisions and constraints.

Section 4 contains the Scheduler 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 Scheduler Application Heritage Analysis 582-2007-030

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 SCHEDULER Application Context

The Scheduler (SCH) application is responsible for implementing a Time Division Multiplexer (TDM) scheduling mechanism for hosted system applications. SCH provides the ability to deterministically schedule application processing at specific times within a 1 second period, AND those scheduling opportunities are synchronized to the Spacecraft Clock.

The figure below shows major interfaces between the Scheduler task and other core Flight Executive (cFE) and Core Flight System (CFS) tasks. Note that although it isn't shown explicitly, all task-to-task communications are accomplished via the cFE Software Bus task.

Inputs to the Scheduler Application include:

- 1) Major Frame signal to drive the Scheduler Application
- 2) Minor frame signal,
- 3) Commands originating from outside sources such as the ground or Stored Command Processor.

Outputs from the Scheduler application include:

- 1) Schedule Commands which can include Housekeeping requests, wake-up calls etc.
- 2) Event Messages and
- 3) Scheduler Housekeeping telemetry

Two tables are used by the Scheduler App

- 1) The Schedule Definition Table (SDT) defines that major and minor frame activities that are to be performed.
- 2) The Message Definition Table (MDT) defines the messages that are sent as part of the activities.

An activity is defined in the SDT by: an Activity Type (currently only supports “Send Message Activity Type”), ActivityState (“Unused”, “Enabled”, “Disabled”), Activity Frequency (e.g. – number of seconds between activity executions), Activity Offset (e.g. – phase shift, in seconds, of first execution) and Activity Group/Multi-Group identifiers.

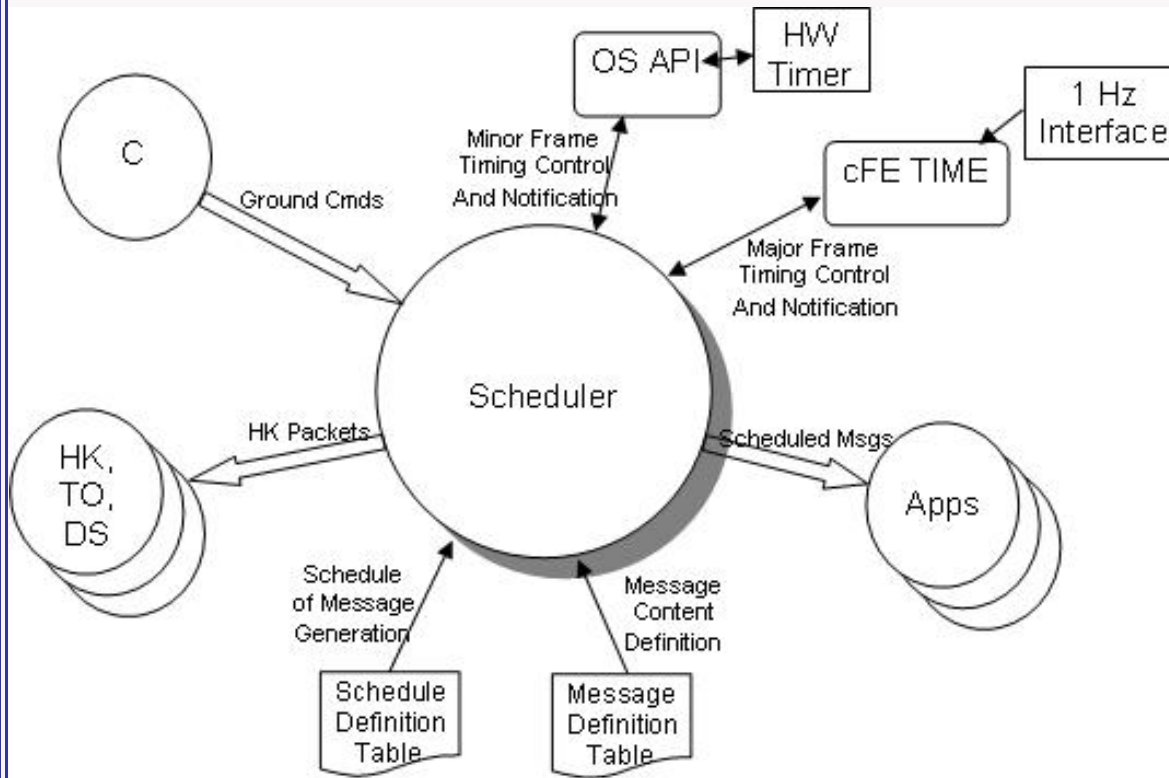


Figure 2.0 – CFS SCH Context

2.1 Assumptions

The following list summarizes the assumptions made by the CFS Scheduler Application:

- cFE API and OSAL are being used
- Major Frame signal available to drive and synchronize the schedule
- Minor Frame signal available to drive the minor frames (slots)

3.0 Design Specifications

The Scheduler 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 Scheduler Application Heritage Analysis document.

3.1 Design Constraints

Note that the Major Frame signal does not have to be a 1Hz signal. The cFE references a 1Hz tone signal that, as it turns out, does not have to be 1Hz. Confusion, however, may arise if the tone is something other than 1Hz since the cFE code and documentation make such references. The term Major Frame Signal is used for the Scheduler Application.

4.0 Subsystem Requirements

2990	CFS-400	The CFS shall support a table driven Time Division Multiplexed (TDM) Input/Output Scheme for scheduling system activities	Provides ability to deterministically schedule application processing at specific times	LRO, BAT
------	---------	---	---	----------

2992		<h2>5.0 Detailed Requirements</h2> <h3>5.1 Basic Requirements</h3> <p>The following requirements are basic requirements of Scheduler. Some of them are included here to avoid repeating these requirements for each applicable requirement.</p>		
2894	SCH1000	Upon receipt of a No-Op command, SCH shall increment the SCH Valid Command Counter and generate an event message.	Debug command to verify application is alive	LRO
2896	SCH1001	Upon receipt of a Reset command, SCH shall reset the following housekeeping variables to a value of zero: <ul style="list-style-type: none"> a) Valid Ground Command Counter b) Ground Command Rejected Counter e) Valid commands sent by the SCH APP f) Commands sent by SCH APP that were reported as erroneous by the Software Bus g) Number of Slots (minor frames) processed h) Number of Times Slots skipped i) Number of Times Multiple Slots processed j) Number of times that SCH woke up in the same slot as last time k) Number of bad entries found in the Scheduler table (indication of corrupted table) 	Important for testing and on-orbit flight operations in order to start with a "clean slate"	LRO, BAT
2898	SCH1002	For all SCH commands, if the length contained in the message header is not equal to the expected length, SCH shall reject the command and issue an event message.	Basic command verification in the event of SEU or memory corruption	LRO, BAT
2900	SCH1004	If SCH accepts any command as valid, SCH shall execute the command, increment the SCH Valid Command Counter and issue an event message	Operators require feedback on command execution	LRO, BAT
2902	SCH1005	If SCH rejects any command, SCH shall abort the command execution, increment the SCH Command Rejected Counter and issue an error event message	Operators require feedback on command execution	LRO, BAT

2994		<h2>5.2 Scheduling of Activities</h2> <p>The following requirements relate to the scheduling of activities.</p>		
2904	SCH2000	Upon receipt of a Major Frame Signal, SCH shall start processing the Scheduler Definition Table (SDT) which contains the defined activities	Need a synchronization for the major frame in order to synchronize events.	LRO, BAT
2906	SCH2001	The Schedule Definition Table shall schedule activities with a minimum minor frame resolution of <PLATFORM_DEFINED, 10> milliseconds and a major frame resolution of <PLATFORM_DEFINED, 1> second.	The TDM scheduling paradigm for all heritage scheduler applications is based upon a 1 Second time period, divided into a designer determined collection of equally divided minor time frames within the 1 Sec period	LRO, BAT
2908	SCH2002	SCH shall schedule a maximum of <PLATFORM_DEFINED, 5> activities in each minor frame.	Provides a bounds for the Scheduler Definition Table for each mission's platform	LRO, BAT

2910	SCH2003	SCH shall identify the spacecraft Major Frame Signal as “noisy” when two Major Frame Signals occur consecutively that are separated from the previous Major Frame Signal by less than <PLATFORM_DEFINED, 990> milliseconds (running fast) and/or greater than <PLATFORM_DEFINED, 1010> milliseconds (running slow)	Major Frame Signal should retain a platform defined accuracy compared to the platform's minor frame clock source. Two errors are required for identification to cover the case when the clock is being synchronized.	None
2912	SCH2003.1	If the Major Frame Signal is running fast, SCH shall process the activities contained in the slots not processed prior to receiving the Major Frame Signal, up to the <PLATFORM_DEFINED> maximum number of activities allowed to be processed in a minor frame	Trying to catch up if the major fram signal arrives early. Process all of the activities that you can in a minor frame .	LRO
2914	SCH2003.2	Any remaning activites shall be scheduled in the subsequent minor frames until there are no unprocessed activities from the previous Major Frame	Continue to try to catch up by processing activities from the previous Major Frame until all of the activities have been executed.	LRO
2916	SCH2003.3	If the number of slots not processed prior to receiving the Major Frame Signal is greater than <PLATFORM_DEFINED> number of allowable “catch-up” slots, SCH shall: a) begin processing the first slot of the major frame b) skip the unprocessed slots c) incrementing the Number of Times Slots skipped counter	In this case, the Scheduler can't catch up and should just start processing the activities starting in the first slot (skipping unprocessed slots and activities in the previous Major Frame)	LRO
2918	SCH2003.4	If more than <PLATFORM_DEFINED> consecutive Major Frame Signals are noisy, SCH shall a) disable synchronizing to the Major Frame Signal and b) use the Minor Frame Signal for synchronization c) Set the SCH synchronization status to “flywheel”	Major frame is unreliable so need to rely on minor frame	LRO

2920	SCH2004	SCH shall schedule each activity's frequency as specified by that activity's table-defined major frame frequency (eg. every 5 seconds)	Need to provide the ability to schedule activities that occur in multiples of seconds (send packet X every 5 seconds)	LRO, BAT
2922	SCH2005	SCH shall schedule each activity's frequency as specified by that activity's table-defined major frame frequency offset (eg. every 5 seconds starting at second 2)	Want to be able to stagger activities in order to spread out the CPU loading	LRO, BAT
2924	SCH2006	Upon receipt of a SDT Load, SCH shall validate the entries of the table and if any entry is invalid, the table is declared invalid and rejected.	cFE Table Services provides an interface to modify the Scheduler App tables. Need to validate the table	LRO
2926	SCH2006.1	SCH shall identify any "Unused" SDT entry that contains data other than "Unused" as invalid	Entries are defined as "Enabled", "Disabled" or "Unused". Any entry labeled "Unused" that contains data indicates an error in the table.	LRO
2928	SCH2006.2	SCH shall identify any "Enabled" or "Disabled" SDT entry that contains a Frequency of zero as invalid	Every entry is required to have at least a Frequency of 1	LRO
2930	SCH2006.3	SCH shall identify any "Enabled" or "Disabled" SDT entry that contains a Frequency Offset greater than the entry's specified Frequency as invalid.	A Frequency Offset greater than the specified Frequency does not make sense.	LRO
2932	SCH2006.4	SCH shall identify any "Enabled" or "Disabled" SDT entry that contains a Message Index of zero or greater than the maximum number of Messages in the Message Definition Table as invalid.	A Message Index must have a chance of identifying a message definition in the Message Definition Table.	LRO

2934	SCH2006.5	SCH shall identify any "Enabled" or "Disabled" SDT entry that contains an undefined Activity Type as invalid.	SCH currently only supports the Send Message Activity Type	.
2936	SCH2007	Upon receipt of a Message Definition Table Load, SCH shall validate the entries of the table and if any table entry is invalid, reject the table	cFE Table Services provides an interface to modify the Scheduler App tables. Need to validate the table	LRO
2938	SCH2007.1	SCH shall identify any "Unused" MDT entry that contains data other than "Unused" as invalid	Table entries should be self consistent	LRO
2940	SCH2007.2	SCH shall identify any MDT entry that has a message definition with a Message ID that is greater than <MISSION_DEFINED> maximum Message ID as invalid	Messages whose Message ID are out of range are invalid	LRO
2942	SCH2007.3	SCH shall identify any MDT entry that has a message definition with a Message ID that is less than <MISSION_DEFINED> minimum Message ID as invalid	Messages whose Message ID are out of range are invalid	LRO
2944	SCH2007.4	SCH shall identify any MDT entry that has a message length greater than <PLATFORM_DEFINED> maximum message length as invalid	Messages of an unacceptable length are invalid	LRO
2946	SCH2007.5	SCH shall identify any MDT entry that has a message length that is an odd number of bytes as invalid	SCH requires all messages to be an even number of bytes in length	LRO
2996		<h2>5.3 Activity Commands and Groups</h2> <p>Below are the requirements associated with the commands that can be sent as part of an activity.</p>		
2948	SCH3000	For each "Enabled" activity defined in the SDT, SCH shall send the specified message defined in the Message Definition Table (MDT) via a Software Bus message.	The definition of the action part of the activity is defined in the MDT.	.

2950	SCH3001	SCH shall associate each activity with a single Group and zero to 24 Multi-Groups.	Grouping activities allows related activities to be enabled/disabled simultaneously	.
2952	SCH3001.1	SCH shall support a maximum of 255 Groups	Need to limit # of groups to a byte range so that Groups and Multi-Groups can be combined into a single 32-bit integer (bit mask)	LRO
2954	SCH3002	SCH shall support a maximum of <PLATFORM_DEFINED, 255> messages in the Message Definition Table	Need to limit for table sizing purposes	LRO
2956	SCH3003	SCH shall allow the user to combine one or more groups of activities in the Schedule Definition Table into one or more Multi-Groups.	NOTE: Multi-groups can be thought of as super groups of the individual groups or just another possible group classification for the activities.	.
2958	SCH3003.1	SCH shall support a maximum of 24 Multi-Groups.	Need to limit for implementation reasons. The maximum of 24 was chosen by the intended implementation of the requirement via a bit mask that is limited to 24 bits.	.

2998		<h2>5.4 Scheduler App Commands</h2> <p>The Scheduler Application is driven by the definitions in the supported tables and does not require much in the way of ground command support. There are a few commands that are required of the Scheduler Application in order to control the enabling/disabling of individual activities and the enabling/disabling of groups of commands from the ground (or stored command).</p>		
2960	SCH4000	Upon receipt of an Enable Activity Command, SCH shall enable the command-specified activity defined in the Schedule Definition Table	Provides the ability to enable an individual activity within a minor frame	LRO, BAT
2962	SCH4000.1	If an "Unused" Activity definition is specified by the command, the command shall be rejected	"Unused" Activities should not be able to be "Enabled"	LRO, BAT
2964	SCH4000.2	If the command specified activity identifier is out of range, the command shall be rejected	Command Parameter is out of range	LRO
2966	SCH4001	Upon receipt of a Disable Activity Command, SCH shall disable further processing of the command-specified activity defined in the Schedule Definition Table	Provides the ability to disable an individual activity within a minor frame	LRO, BAT
2968	SCH4001.1	If an "Unused" Activity definition is specified by the command, the command shall be rejected	"Unused" Activities should not be able to be "Disabled"	LRO, BAT
2970	SCH4001.2	If the command specified activity identifier is out of range, the command shall be rejected	Command Parameter is out of range	LRO
2972	SCH4002	Upon receipt of an Enable Group Command, SCH shall enable all the activities associated with the command-specified Group and/or Multi-Group	Provides the operators with the ability to enable groups of activities. An activity Group is a logical categorization of associated command activities	LRO

2974	SCH4002.1	If no Group or Multi-Group is specified then the command shall be rejected	Command does not make sense unless a Group is identified	LRO
2976	SCH4003	Upon receipt of a Disable Group Command, SCH shall disable further processing of all activities associated with the command-specified Group and/or Multi-Group.	Provides the operators with the ability to disable groups of activities. An activity Group is a logical categorization of associated command activities	LRO
2978	SCH4003.1	If no Group or Multi-Group is specified then the command shall be rejected	Command does not make sense unless a Group is identified	LRO
2980	SCH4004	Upon receipt of a SCH Diagnostic Command, SCH shall produce a Diagnostic Message containing, at a minimum, the Activity State for all Activities.	With the ability to enable/disable activities, the operator needs a method of determining their runtime states.	.
2982	SCH4005	Upon receipt of an Enable Major Frame Synchronization Command, SCH shall synchronize its Scheduler Definition Table (SDT) containing defined activities to the next Major Frame Signal	If SCH has to switch to using the Minor Frame Signal due to a noisy Major Frame Signal, the ground needs a way to switch to using the Major Frame as the Synchronization	.
3000		5.5 Status Reporting		

2984	SCH8000	SCH shall generate a housekeeping message containing the following: a) Valid Ground Command Counter b) Ground Command Rejected Counter e) Valid commands sent by the SCH APP f) Commands sent by SCH APP that were reported as erroneous by the Software Bus g) Number of Slots processed h) Number of Times Slots skipped i) Number of Times Multiple Slots processed j) Number of times that SCH woke up in the same slot as last time k) Number of bad entries found in the Scheduler table (indication of corrupted table) l) Synchronization status (flywheeling)	Housekeeping telemetry to indicate basic SCH status	LRO, BAT
3002		<h2>5.6 Initialization Requirements</h2> <p>The following are the requirements associated with Scheduler on an Application reset, cFE Processor Reset or a cFE Power-on Reset</p>		
2986	SCH9000	Upon any Initialization of the SCH Application (cFE Power On, cFE Processor Reset or SCH Application Reset), SCH shall initialize the following data to Zero: a) Valid Ground Command Counter b) Ground Command Rejected Counter e) Valid commands sent by the SCH APP f) Commands sent by SCH APP that were reported as erroneous by the Software Bus g) Number of Slots processed h) Number of Times Slots skipped i) Number of Times Multiple Slots processed j) Number of times that SCH woke up in the same slot as last time k) Number of bad entries found in the Scheduler table (indication of corrupted table) l) Synchronization status (not flywheeling)	No information is preserved across a cFE Processor reset or SCH Application Reset.	Derived
2988	SCH9001	Upon any Initialization, the SCH Application shall inhibit processing of the Schedule Definition Table until the cFE indicates that all of the applications have started.	Processor and Power-On Resets result in an initial race condition as Applications start up. The SCH needs to hold off processing until the system has stabilized. It will begin processing when cFE's Executive Services indicates that all applications have started	.

3004

Appendix A Acronyms and Abbreviations

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

- **Activity** - performed each scheduled activity period identified within the SCH Schedule Definition Table (SDT). Each entry in the SDT is an activity (eg. send the Memory Manager HK request packet every 4 seconds starting at second 2)
- **File** – A logical group of data, but more flexible than tables. May reside in memory or on physical media like a hard disk. Can grow or shrink in size, can be deleted or created at any time. Can be grouped in hierarchical directories if the file system supports it.
- **Group** – A logical collection of Activities that can be Enabled/Disabled simultaneously.
- **Major Frame** - A time period which is synchronized to a Major Frame Signal time source
- **MDT** – Message Definition Table. Table that contains the messages (commands) that are to be sent based on the SDT.
- **Minor Frame** – collection of equally divided minor time frames within a major frame
- **Multi-Group** – A logical collection of Activities and/or Groups that can be Enabled/Disabled simultaneously.
- **Operating System Abstraction Layer (OSAL)** – Set of functions supplied as part of the cFE that isolate the calling application from operating system dependencies.

- **Scheduler (SCH)** – Scheduler Application

- **SDT** – Schedule Definition Table. Table that contains the definition of the activities to be performed

- **Slot** – Minor Frame

- **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.