Code 582
Flight Software Branch

CORE FLIGHT SYSTEM SCHEDULER BUILD 2.2.1.0

FLIGHT SOFTWARE BUILD VERIFICATON TEST REPORT

Flight Software Branch – Code 582

Version 1.0

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

SIGNATURES

Submitted by:

X WALTER MOLESKI

Digitally signed by WALTER MOLESKI
DN: c=US, o=U.S. Government, ou=NASA, ou=People, cn=WALTER MOLESKI, 0.9.2342.19200300.100.1.1=wmoleski
Date: 2017.07.05 12:37:45 -04'00'

Walt Moleski/Code 582
cFS Flight Software Tester

Approved by:

X SUSANNE STREGE

Digitally signed by SUSANNE STREGE
DN: c=US, o=U.S. Government, ou=NASA, ou=People, cn=SUSANNE STREGE, 0.9.2342.19200300.100.1.1=sstrege
Date: 2017.07.05 10:56:48 -04'00'

Susanne Strege/Code 582
cFS Flight Software Product Development Lead

PLAN UPDATE HISTORY

Version	Date	Description	Affected Pages
Draft	7/3/2017	Draft for review	All

TABLE OF CONTENTS

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

1 INTRODUCTION

1.1 DOCUMENT PURPOSE

This Test Report describes the test results from the Core Flight System (cFS) Scheduler (SCH) Flight Software (FSW) Test Team build 2.2.1.0 verification testing. It is used to verify that the SCH FSW has been tested in a manner that validates that it satisfies the functional and performance requirements defined within the cFS SCH Requirements Document. This Test Report summarizes the FSW test history, the build verification process, the build test configuration, and the test execution and results.

1.2 APPLICABLE DOCUMENTS

Unless otherwise stated, these documents refer to the latest version.

Parent Documents (Mission and FSW)

- 582-2008-036 cFS Scheduler Requirements Document, Version 1.1
- 582-2008-012 cFS Deployment Guide, Version 3.1

Reference Documents

All of the references below can be found on the Code 582 internal website at http://fsw.gsfc.nasa.gov/

- 582-2003-001 FSB FSW Test Plan Template
- 582-2004-001 FSB FSW Test Description Template
- 582-2004-002 FSB FSW Test Scenario Template
- 582-2004-003 FSB FSW Test Procedure Template
- 582-2004-004 FSB FSW Test Execution Summary Template
- 582-2004-005 FSB Test Product Peer Review Form
- 582-2000-002 FSB FSW Unit Test Standard

1.3 DOCUMENT ORGANIZATION

Section 1 of this document presents some introductory material.

Section 2 provides a flight software overview and context along with the test history and testing overview.

Section 3 describes the build verification process including procedure development and execution and test products produced.

Section 4 describes the build test configuration which includes an overview of the testbed and the requirements verification matrix.

Section 5 describes the test execution and results by subsystem.

Appendix A - provides the Requirements Traceability Matrix

Appendix B - provides the Command, Telemetry, and Events Verification Matrix

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

1.4 DEFINITIONS

There were 3 verifications methods used during build verification testing. They were:

- Demonstration: Show compliance with system requirement by exhibiting the required capability (e.g. by demonstrating interactive capability, display capability, print capability, etc.
- Inspection: Show compliance with a system requirement by visual verification of the software (e.g. verifying preparation for delivery, proper interfacing)
- Analysis: Perform detailed analysis of code, generated data (both intermediate data and final output data), etc., to determine compliance with system requirements.

The fields in the Requirements Verification Matrix in Section 4.3 are defined as follows:

- Requirements Tested Passed: Requirement was fully tested in a build test procedure and passed all tests.
- Requirements Tested Failed: Requirement was fully tested in a build test procedure and failed one or more aspect of the testing.
- Requirements Tested Partially: Requirement was tested partially in a build test procedure. To be fully tested, the partially tested requirement is either tested additionally in one or more other test procedures within the same build **and/or** other aspects of the requirement must be tested in a later build, due to capabilities not present in the current build
- Total Tested: Total number of requirements fully tested in a build test procedure. Includes total passed and total failed, but does **not** include requirements tested partially, **unless** (included as a separate entry) testing in multiple procedures within the same build constitutes total testing of a particular requirement. Total Requirements Tested is computed this way in order to avoid multiple counting of individual requirements that are tested partially in more than one procedure.
- Deferred: Number of requirements that were planned to be tested in current build, but were not tested due to some FSW capability or necessary system component not being present.
- Total: Total Requirements Tested + Number of Requirements Deferred

In each software test section in Section 5 there is a table of DCRs. The state definitions are as follows:

- Opened: The DCR is currently being addressed
- Assigned: The DCR was accepted and the modification is being addressed
- InTest: The DCR was corrected and is currently in test
- Validated: The DCR was corrected and tested and have been validated, needs to have a CCB to close the DCR
- Closed: The DCR is closed and have been resolved and tested to satisfaction
- Closed with Defect: The DCR is closed and the defect is most likely assigned a differed DCR number associated with another subsystem.

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

2 OVERVIEW

2.1 FLIGHT DATA SYSTEM CONTEXT

Figure 2-1 illustrates the cFS system context. The cFE interfaces to five external systems: an Operating System (OS), a Hardware Platform (HP), an Operational Interface (OI), Applications (APP), and other cFE-based systems.

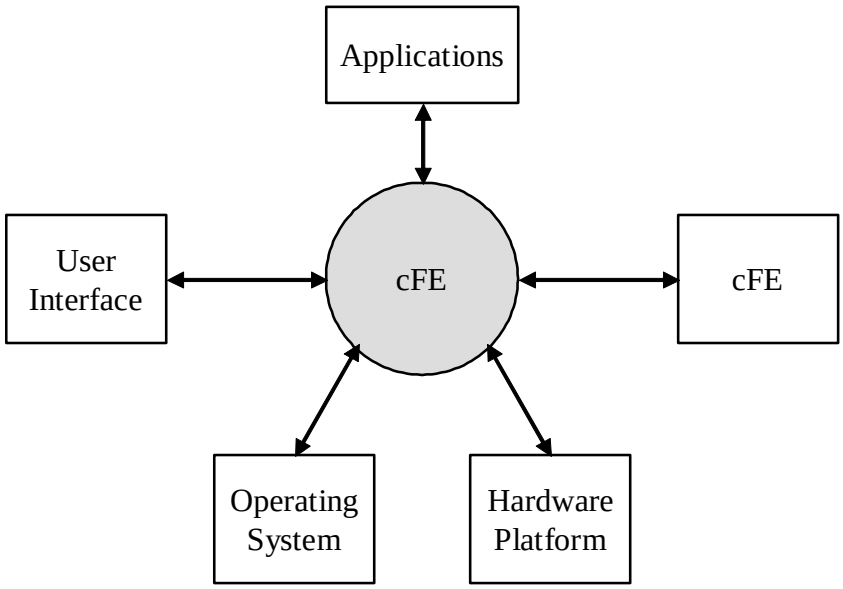

Figure 2-1 cFS System Context

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:

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

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"), Activity State ("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.

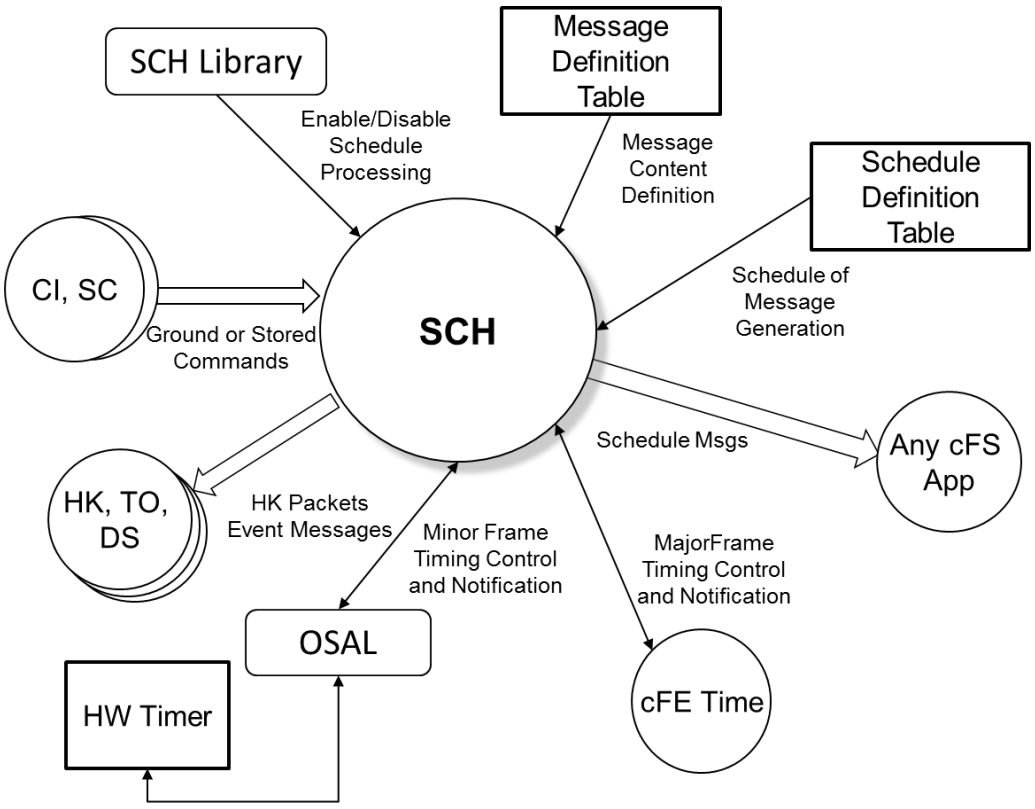

Figure 2-2 cFS SCH Context

2.2 TEST HISTORY

SCH 1.0.1.0 – Build Verification Testing completed 4/2/2009 by Ezinne Uzo-Okoro
SCH 2.0.0.0 – Build Verification Testing completed 9/3/2009 by Walt Moleski
SCH 2.1.2.0 – Build Verification Testing completed 8/16/2011 by Walt Moleski
SCH 2.2.0.0 – Build Verification Testing completed 8/7/2012 by Walt Moleski
SCH 2.2.1.0 – Build Verification Testing completed 7/3/2017 by Walt Moleski

2.3 TESTING OVERVIEW

The SCH application was tested during Build Verification testing using the following:

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

- 1 test application: tst_sch
- 4 test procedures: sch_gencmds, sch_timing_analysis, sch_functional_test, sch_lib_test
- Tests require use of the ASIST Ground Station.

The tst_sch test application is used to send schedule requests for the output of the CFE_TIME 1 Hz signal, and cFE and SCH's housekeeping data. This was useful when performing build verification testing since it provided great control over the sequence of steps. When deployed for a mission, the Scheduler Application would provide this request. In addition, the test application also provides the ability to force Scheduler to miss slots during processing. TST_SCH contains the following ground commands that are used by the SCH test procedures and/or contained in the Scheduler Message Definition Table:

- TST_SCH_NOOP
  This command increments the TST_SCH command counter.
- TST_SCH_ResetCtrs
  This command sets the TST_SCH command and error counters to a zero value.
- TST_SCH_MissSlots
  This command is used to force SCH to skip some slots while processing the minor frame. A slot skip is achieved by calling the CFE_Time_ExternalTone() function.
- TST_SCH_EnableSchProc
  This command is used to call the SCH_EnableProcessing() function contained in the SCH_LIB shared library. The call is only performed if the SCH_LIB_PRESENCE configuration parameter is set to 1 (one).
- TST_SCH_DisableSchProc
  This command is used to call the SCH_DisableProcessing() function contained in the SCH_LIB shared library. The call is only performed if the SCH_LIB_PRESENCE configuration parameter is set to 1 (one).
- TST_SCH_GetSchState
  This command is used to call the SCH_GetProcessingState() function contained in the SCH_LIB shared library. The call is only performed if the SCH_LIB_PRESENCE configuration parameter is set to 1 (one).

The 4 SCH test procedures do the following:

Procedure	**Description**
SCH_GenCmds	The purpose of this test is to verify the Scheduler (SCH) Application general commands function properly. The NOOP and Reset Counters commands will be tested as well as invalid commands to see if the SCH application handles these appropriately.
SCH_Timing_Analysis	The purpose of this test is to verify the Scheduler (SCH) Application responds correctly to changes in the schedule and message definition tables. It also verifies the timing of the SCH schedule and message definition table processing. This test also verifies that the SCH application handles anomalies appropriately.
SCH_Functional_Test	The purpose of this test is to verify the Scheduler (SCH) Application responds correctly and timely to definition changes in the schedule and message table in order to the enabling/disabling of individual activities. This test also verifies that the SCH application handles anomalies appropriately.
SCH_Lib_Test	The purpose of this test is to verify that the SCH application supports the schedule library functions properly. These functions allow an external application to control the enabling/disabling of the SCH application's activity processing.

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

The cFS Deployment Guide contains the instruction for how to set up both the cFS Flight and Ground test environment. The testers use a cFS Test Account for each build test. This account runs ASIST and is setup to contain all the files needed to test the application. These files are extracted from MKS, the source repository tool. Included in these files are test utilities. These utilities can be located in 2 places depending upon whether they are "local" or "global" utilities. The local utilities are extracted into the working prc directory ($WORK/prc). The global utilities are pointed to by ASIST in the global area defined on the test system. Additional tools utilized by the test procedures are located in the $TOOLS directory. It is assumed that test procedures and the ASIST telemetry database used for testing is built using procedure and database templates

The following utilities were used during testing:

Name	**Description**
cfe_startup	Directive combines the "start_data_center", "open_tlm", and "open cmd <cpu>" ASIST startup commands.
load_start_app	Procedure to load and start a user application from the $WORK/apps/cpux directory.
ut_pfindicate	Directive to print the pass fail status of a particular requirement number.
ut_runproc	Directive to formally run the procedure and capture the log file.
ut_sendcmd	Directive to send EVS commands Verifies command processed and command error counters.
ut_sendrawcmd	Send raw commands to the spacecraft. Verifies command processed and command error counters.
ut_setrequirements	A directive to set the status of the cFE requirements array.
ut_setupevents	Directive to look for multiple events and increment a value for each event to indicate receipt.
ut_tlmwait	Directive that waits for the specified telemetry condition to be met
ftp_file	To ftp a file to/from the FSW/GSW.
get_tbl_to_cvt	Directive that issues the CFE_TBL_Dump command and downloads the file to the ground and inserts it in the supplied table telemetry packet.
create_tbl_file_from_cvt	Directive that creates a CFE_TBL load file from the supplied table telemetry packet.
load_table	Directive that transfers the supplied file to the specified cpu and issues the CFE_TBL_Load command.

2.4 VERSION INFORMATION

Item	Version
SCH Requirements	1.1
SCH Application	2.2.1.0
TST_SCH Application	2.2.0.0
cFE	6.5.0.0
OSAL	4.2.0.0
ASIST	20.2
VxWorks	6.9

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

3 BUILD VERIFICATION TEST PREPARATION

3.1 SCENARIO DEVELOPMENT

No new scenarios developed for build verification test 2.2.1.0. All scenarios are stored on the MKS server, in cFS-Repository SCH test-and-ground directory within the test-review-packages subdirectory in the Scenarios folder. It should be noted that as SCH requirement evolve these scenarios are not updated to reflect any changes made.

3.2 PROCEDURE DEVELOPMENT AND EXECUTION

This build test was completed by running 4 test procedures. All test procedures were written using the STOL scripting language. The naming convention for files created by the test procedures was: scx_cpu<#>_<procedure name.<ext>.

3.3 TEST PRODUCTS

Four log files were generated for every procedure that was run. They are defined as follows:

- Logs with the .loge extension list all events sent by the flight software
- Logs with the .logr extension list all requirements that passed validation by demonstration
- Logs with the .logp extension lists all prints that are generated by the test procedure
- Logs with the .logf extension lists everything from the other logs along with the steps in the test procedure
- Logs with the .logs extension lists the SFDU information (if applicable) contained in the full log.

A test summary report is developed in MKS for each procedure by the tester after build testing is completed. All test products are maintained on MKS in the cFS-Repository SCH test-and-ground directory.

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

4 BUILD VERIFICATION TEST EXECUTION

4.1 TESTBED OVERVIEW

SCH FSW testing took place in the cFS FSW Development and Test Facility. A high level view of the cFS FSW Test Bed is shown in Figure 4-1. This facility is located in GSFC Building 23, Room N410. This facility consists of two ASIST workstations running ASIST version 9.7g and three MPC750 CPU boards running VxWorks 6.4. CPU1 is primarily used for development testing while CPU2 and CPU3 are used for build verification testing.

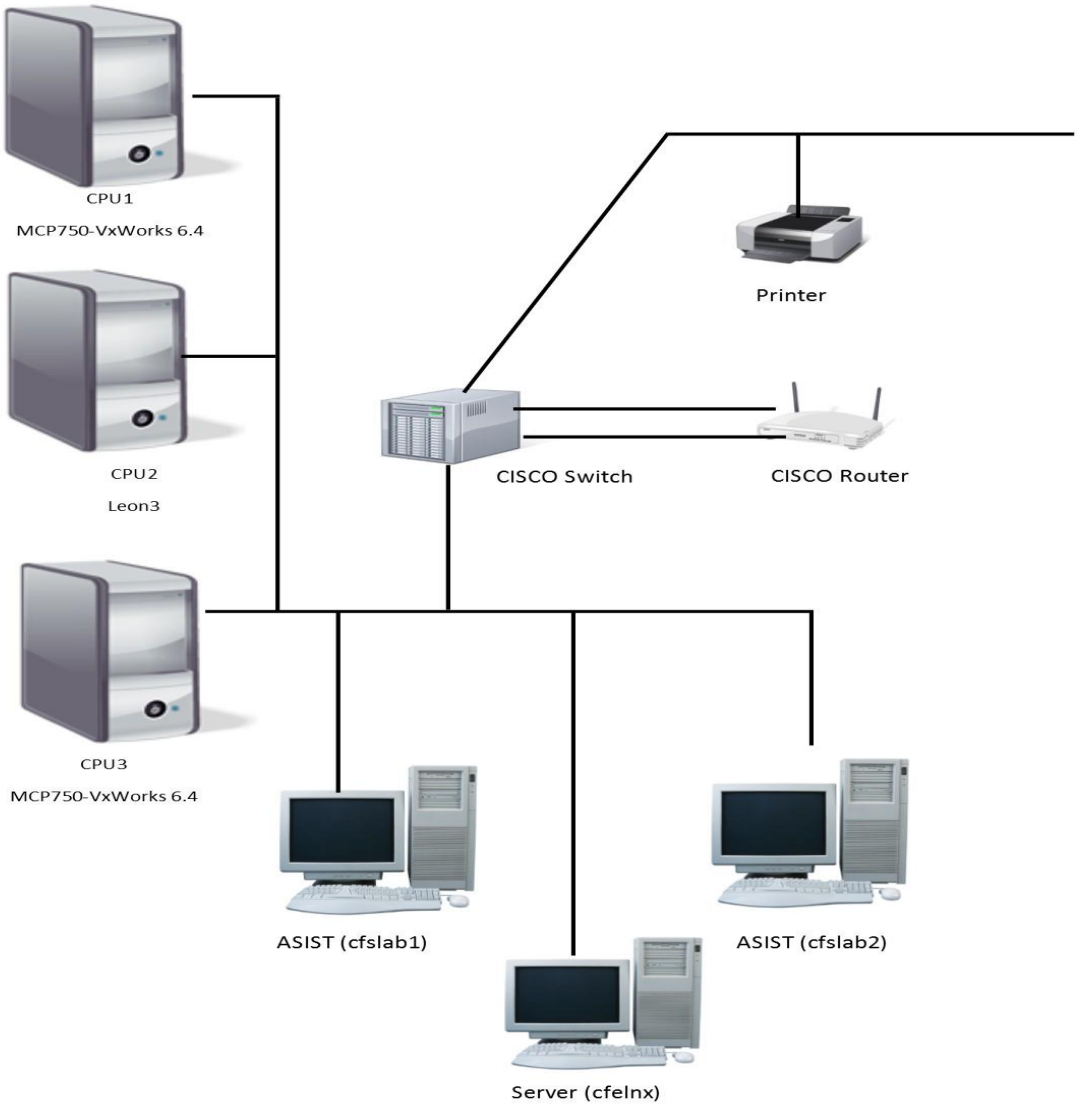

Figure 4-1 cFS FSW Development and Testing Facility

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

4.2 REQUIREMENTS VERIFICATION MATRIX

	Scheduler (SCH)
Requirements Tested Passed	48
Requirements Tested Failed	0
Requirements Tested Partially	0
Total Tested	48
Deferred	0
Total	48

4.3 REQUIREMENTS PARTIALLY TESTED

No requirements were partially tested.

4.4 REQUIREMENTS/FUNCTIONALITY DEFERRED

No requirements were deferred.

4.5 REQUIREMENTS/FUNCTIONALITY DEFERRED FOR MISSION TESTING

No functionality was deferred to mission testing.

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

5 BUILD VERFICIATON TEST RESULTS

5.1 OVERALL ASSESSMENT

During this build test of the SCH Application the software behaved as expected with the exception of the OS Timer Accuracy, which prompted the Scheduler Application to send slots at a minimum of 16.667 ms (10ms is expected). This however, is an artifact of the test environment and is not caused by the SCH Application. Below is a summary of the results:

- 47 requirements passed via demonstration
- 1 requirement was validated by analysis.
- 4 DCRs were verified.

5.2 PROCEDURE DESCRIPTION

Procedure	Description	Requirements tested
SCH_GenCmds	The purpose of this test is to verify the Scheduler (SCH) general commands function properly. The NOOP and Reset Counters commands will be tested as well as invalid commands to see if the SCH application handles these appropriately.	SCH1000; SCH1001; SCH1002; SCH1004; SCH1005; SCH8000; SCH9000; SCH9001
SCH_Timing_Analysis	The purpose of this test is to verify the Scheduler (SCH) Application responds correctly to changes in the schedule and message definition tables. It also verifies the timing of the SCH schedule and message definition table processing. This test also verifies that the SCH application handles anomalies appropriately.	SCH1002; SCH1004; SCH1005; SCH2000; SCH2001; SCH2002; SCH3000; SCH3001; SCH3001.1; SCH3002; SCH3003; SCH3003.1; SCH4000; SCH4000.1; SCH4000.2; SCH4001; SCH4001.1; SCH4001.2; SCH4002; SCH4002.1; SCH4003; SCH4003.1; SCH4004; SCH4005; SCH8000; SCH9000; SCH9001
SCH_Functional_Test	The purpose of this test is to verify the Scheduler (SCH) Application responds correctly and timely to definition changes in the schedule and message table in order to the enabling/disabling of individual activities. This test also verifies that the SCH application handles anomalies appropriately.	SCH2000; SCH2002; SCH2003; SCH2003.1; SCH2003.2; SCH2003.3; SCH2003.4; SCH2004; SCH2005; SCH2006; SCH2006.1; SCH2006.2; SCH2006.3; SCH2006.4; SCH2006.5; SCH2007; SCH2007.1; SCH2007.2; SCH2007.3; SCH2007.4; SCH2007.5; SCH8000; SCH9000; SCH9001
SCH_Lib_Test	The purpose of this test is to verify that the SCH application supports the schedule library functions properly. These functions allow an external application to control the enabling/disabling of the SCH application's activity processing.	SCH8000; SCH9000; SCH9001

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

5.3 ANALYSIS REQUIREMENTS VERIFICATION

The following requirements were verified using analysis:

Requirement	Description	Status	Justification
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>	Pass	The Timing Analysis test procedure scheduled a large amount of NOOP commands in Step 2.1. The resolution can be determined by viewing the log file.

5.4 DCRS

No new DCRs were generated during SCH 2.2.1.0 testing

5.4.1 DCRs Verified

The following DCRs were verified during testing:

DCR	Description	Test Method	Test Approach
4157	Update commands in sch_platform_cfg.h to fix comments	Inspection	Verified the stated fix was made to the file submitted with the DCR.
4174	SCH tables have CCSDS primary packet header that must be Big Endian on Little Endian platforms	Inspection	The sch_def_msgtbl.c file contained the stated macro call on CCSDS primary headers.
4248	SCH – Scheduler task source module sch_custom.c has local definitions of PSP functions.	Inspection	The stated PSP functions were removed from the files submitted with this DCR.
145925	SCH – CFE_EVS_SendEvent Format Warnings	Demonstration	The compile process produced not warnings for the SCH application.

5.4.2 Outstanding DCRs/Trac Tickets

The following outstanding DCRS are written against the Scheduler application:

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

DCR/Trac #	Description
#21	SCH cmake table integration. The cFE 6.5 release adds better support for building table objects from CMakeLists files. This ticket contains an update to the SCH app to use this feature to build its table files in addition to the normal SCH build.
#35	SCH occasionally hangs when CFS is stopped. When the CFS process receives a SIGINT (ctrl-c), the SCH_MinorFrameCallback is called by the pthread_cancel. Often this will cause that callback to hang on a lock.
#45	SCH: Remove dependencies on cfe_platform_cfg
#47	Inaccuracies in Scheduler Minor Frame Slot Timing/Processing
#84	SCH Does Not Protect Scheduler Table Slots from Being Commanded. SCH currently allows any slot within the Scheduler Table to be commanded to an enable/disable state.
#85	SCH Minor Frame Interrupt Processed Before Receipt of Startup Sync. The default custom.c file includes the create timer function in the early initialization function. On NICER this caused the minor frame interrupt to be processed before the SCH application has received the startup sync.

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

#86	Consider Moving Group Macro Definitions to a Header File. Moving these macro definitions to a header file would allow ground system command definitions to include the actual group definitions vs. redefining them which can be error prone.
#87	Should the Scheduler Table's Types be Configurable? MMS Made the SCH table critical. The current CFS SCH options do not allow the Scheduler's table's types to be configurable.
#88	SCH Timer Configuration Issues. The SCH "Minor Frame MET Sync logic" is very dependent on the accuracy of the timer being used. There currently is no way for user's to configure the "Minor Frame MET Sync logic" to work with a variety of different timers.
4121	SCH - Add Trick Simulation Support (JSC Request)

5.5 NOTES

The SCH_Lib_Test procedure was executed twice to test the default state of the SCH Library being present and a second time with the SCH Library not present. The second test required the SCH application to be recompiled with the configuration parameter SCH_LIB_PRESENCE set to 0 (zero).

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

APPENDIX A - RTTM

The SCH Build 2.2.1.0 RTTM can be found on the MKS server cFS-Repository and in the release package in the SCH test-and-ground directory results folder.

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

APPENDIX B - COMMAND, TELEMETRY, AND EVENTS VERIFICATION MATRIX

Command	Test Procedure(s)	Notes/Comments
SCH_NOOP	sch_gencmds	
SCH_RESETCTRS	sch_funnctional_test; sch_gencmds	
SCH_ENABLEENTRY	sch_funnctional_test; sch_timing_analysis	
SCH_DISABLEENTRY	sch_funnctional_test; sch_timing_analysis	
SCH_DISABLEGROUP	sch_funnctional_test; sch_lib_test; sch_timing_analysis	
SCH_ENABLEGROUP	sch_funnctional_test; sch_lib_test; sch_timing_analysis	
SCH_ENABLESYNC	sch_funnctional_test; sch_timing_analysis	
SCH_SENDDIAG	sch_timing_analysis	

Telemetry	Test Procedure(s)	Notes/Comments
SCH_CMDPC	sch_gencmds; sch_functional_test; sch_timing_analysis	
SCH_CMDEC	sch_gencmds; sch_functional_test; sch_timing_analysis	
SCH_SYNCTOMET	sch_gencmds; sch_functional_test; sch_timing_analysis	
SCH_MAJORFRAMESOURCE	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_ACTSUCCESSCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_ACTFAILURECTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_SLOTPROCCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_SKIPSLOTCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_MULTSLOTCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_SAMESLOTCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

Telemetry	Test Procedure(s)	Notes/Comments
SCH_BADTBLDATACTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_TBLPASSVERIFYCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_TBLFAILVERIFYCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_TBLPROCCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_VALIDMFCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_MISSMFCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_UNEXPCTDMFCTR	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_MINORSINCETONE	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_NEXTSLOT	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_LASTSYNCMETSLOT	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_IGNOREMF	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_UNEXPCTDMAJORFRAME	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
SCH_ENTRYSTATES[]		
SCH_MSGIDS[]		

Table Telemetry	Test Procedure(s)	Notes/Comments
SCH_DefaultMessageTable[].MessageBuffer[]	sch_badmdttables; sch_mdtloadfile	
SCH_DefaultScheduleTable[].EnableState	sch_badsdttbls; sch_sdt2; sch_sdt3; sch_sdtloadfile;	
SCH_DefaultScheduleTable[].SE_Type	sch_badsdttbls; sch_functional_test; sch_sdt2; sch_sdt3; sch_sdtloadfile;	
SCH_DefaultScheduleTable[].SE_Frequency	sch_badsdttbls; sch_functional_test; sch_sdt2; sch_sdt3; sch_sdtloadfile;	

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

Table Telemetry	Test Procedure(s)	Notes/Comments
SCH_DefaultScheduleTable[].Remainder	sch_badsdttbls; sch_sdt2; sch_sdt3; sch_sdtloadfile;	
SCH_DefaultScheduleTable[].MessageIndex	sch_badsdttbls; sch_sdt2; sch_sdt3; sch_sdtloadfile;	
SCH_DefaultScheduleTable[].GroupData	sch_badsdttbls; sch_sdt2; sch_sdt3; sch_sdtloadfile;	

	Event Message Ids	Test Procedure(s)	Notes/Comments
1	SCH_INITSTATS_INF_EID	sch_gencmds; sch_functional_test; sch_lib_test; sch_timing_analysis	
2	SCH_APP_EXIT_EID		
3	SCH_CR_PIPE_ERR_EID		
4	SCH_SUB_HK_REQ_ERR_EID		
5	SCH_SUB_GND_CMD_ERR_EID		
7	SCH_SDT_REG_ERR_EID		
8	SCH_MDT_REG_ERR_EID		
9	SCH_SDT_LOAD_ERR_EID		
10	SCH_MDT_LOAD_ERR_EID		
11	SCH_ACQ_PTR_ERR_EID		
12	SCH_MINOR_FRAME_TIMER_CREATE_ERR_EID		
13	SCH_MINOR_FRAME_TIMER_ACC_WARN_EID	sch_gencmds; sch_functional_test; sch_lib_test; sch_timing_analysis	
14	SCH_MAJOR_FRAME_SUB_ERR_EID		
15	SCH_SEM_CREATE_ERR_EID		
16	SCH_SAME_SLOT_EID	sch_functional_test; sch_gencmds; sch_lib_test;	
17	SCH_SKIPPED_SLOTS_EID	sch_functional_test;	
18	SCH_MULTI_SLOTS_EID	sch_functional_test;	
19	SCH_CORRUPTION_EID		
20	SCH_PACKET_SEND_EID		
21	SCH_NOISY_MAJOR_FRAME_ERR_EID	sch_functional_test; sch_lib_test;	
30	SCH_SCHEDULE_TBL_ERR_EID	sch_functional_test;	
31	SCH_SCHEDULE_TABLE_EID	sch_functional_test; sch_timing_analysis	
32	SCH_MESSAGE_TBL_ERR_EID	sch_functional_test;	
33	SCH_MESSAGE_TABLE_EID	sch_functional_test;	
40	SCH_NOOP_CMD_EID	sch_gencmds; sch_lib_test; sch_functional_test; sch_timing_analysis	
41	SCH_RESET_CMD_EID	sch_gencmds; sch_functional_test;	
42	SCH_ENABLE_CMD_EID	sch_functional_test; sch_timing_analysis	

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/

	Event Message Ids	**Test Procedure(s)**	**Notes/Comments**
43	SCH_DISABLE_CMD_EID	sch_functional_test; sch_timing_analysis	
44	SCH_ENA_GRP_CMD_EID	sch_functional_test; sch_lib_test; sch_timing_analysis	
45	SCH_DIS_GRP_CMD_EID	sch_functional_test; sch_lib_test; sch_timing_analysis	
46	SCH_ENA_SYNC_CMD_EID	sch_functional_test; sch_timing_analysis	
47	SCH_SEND_DIAG_CMD_EID	sch_timing_analysis	
50	SCH_ENABLE_CMD_ARG_ERR_EID	sch_timing_analysis	
51	SCH_ENABLE_CMD_ENTRY_ERR_EID	sch_timing_analysis	
52	SCH_DISABLE_CMD_ARG_ERR_EID	sch_timing_analysis	
53	SCH_DISABLE_CMD_ENTRY_ERR_EID	sch_timing_analysis	
54	SCH_ENA_GRP_CMD_ERR_EID	sch_timing_analysis	
55	SCH_ENA_GRP_NOT_FOUND_ERR_EID	sch_timing_analysis	
56	SCH_DIS_GRP_CMD_ERR_EID	sch_timing_analysis	
57	SCH_DIS_GRP_NOT_FOUND_ERR_EID	sch_timing_analysis	
58	SCH_CC_ERR_EID	sch_gencmds;	
59	SCH_MD_ERR_EID		
60	SCH_CMD_LEN_ERR_EID	sch_gencmds; sch_timing_analysis	

Printed copies of this document are for REFERENCE PURPOSES ONLY! The only controlled copy of this document is located on-line at http://fsw.gsfc.nasa.gov/internal/StandardsCCB/