

Code 582
Flight Software Branch

**CORE FLIGHT EXECUTIVE
BUILD 6.4.1.0**

**FLIGHT SOFTWARE BUILD VERIFICATION
TEST REPORT**

Flight Software Branch – Code 582

Version 1.0

SIGNATURES

Submitted by:



Walt Moleski/ 582
CFS Flight Software Tester

12/10/14

Date

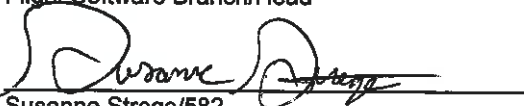
Approved by:



David McComas/582
Flight Software Branch/Head

12/10/14

Date



Susanne Stregge/582
cFE/CFS Flight Software Product Development Lead

12/10/14

Date

Core Flight Executive Flight Software Build Verification Test Report
Build 6.4.1.0

Core Flight Executive Flight Software Build Verification Test Report
Build 6.4.1.0

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	Document Purpose	1
1.2	Applicable Documents	1
1.3	Document Organization	1
1.4	Definitions	2
2	OVERVIEW	3
2.1	Flight Data System Context	3
2.2	Test History	3
2.3	Testing Overview	3
3	BUILD VERIFICATION TEST PREPARATION	5
3.1	Scenerio Development.....	5
3.2	Procedure Development and Execution.....	5
3.3	Test Products.....	5
4	BUILD VERIFICATION TEST EXECUTION	6
4.1	Testbed Overview	6
4.2	Requirements Verification Matrix	7
4.3	Requirements Partially Tested	7
4.4	Requirements Deferred.....	7
5	BUILD VERIFICATON TEST RESULTS.....	9
5.1	Software Bus Services (SB).....	9
5.1.1	Overall Assessment.....	9
5.1.2	Procedure Description	9
5.1.3	Analysis Requirements Verification	9
5.1.4	DCRs.....	10
5.1.5	Notes	10
5.2	Table Services (TBL)	10
5.2.1	Overall Assessment.....	10
5.2.2	Procedure Description	10
5.2.3	Analysis Requirements Verification	11
5.2.4	DCRs.....	11
5.2.5	Notes	11
5.3	DCRs verified	12
	APPENDIX A - RTTM.....	13
	APPENDIX B - COMMAND, TELEMETRY, AND EVENTS VERIFICATION MATRIX	14

1 INTRODUCTION

1.1 DOCUMENT PURPOSE

This Test Report describes the test results from the core Flight Executive (cFE) Flight Software (FSW) Test Team build 6.4.1.0 verification testing. It is used to verify that the cFE FSW has been tested in a manner that validates that it satisfies the functional and performance requirements defined within the cFE FSW Requirements Specification and all Discrepancy/Change Request (DCR) fixes and code updates assigned to build 6.4.1.0. 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)

- 582-2000-012 FSB Flight Software TestBed Requirements Guidelines

Reference Documents

All of the references below can be found on the Code 582 internal website at <https://fsb.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
- 582-2007-040 FSB Test Analysis Summary Template
- 582-2008-006 FSB Testbed Validation Description

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

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 DCR's. 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.

2 OVERVIEW

2.1 FLIGHT DATA SYSTEM CONTEXT

Build verification was performed using cFE in a single flight processor context, as depicted in Figure 2-1. The ground system interfaces with the flight Applications Command Ingest (CI) and Telemetry Output (TO) and not directly with the cFE. Spacecraft operators send Commands and Files to the cFE and receive Files, Events, and Telemetry from the cFE. Note that this context is relative to the cFE and does not show ground communications with other Applications. For example, a typical spacecraft has a Stored Command (SC) Application that receives stored command loads from the ground and sends stored command dumps to the ground.

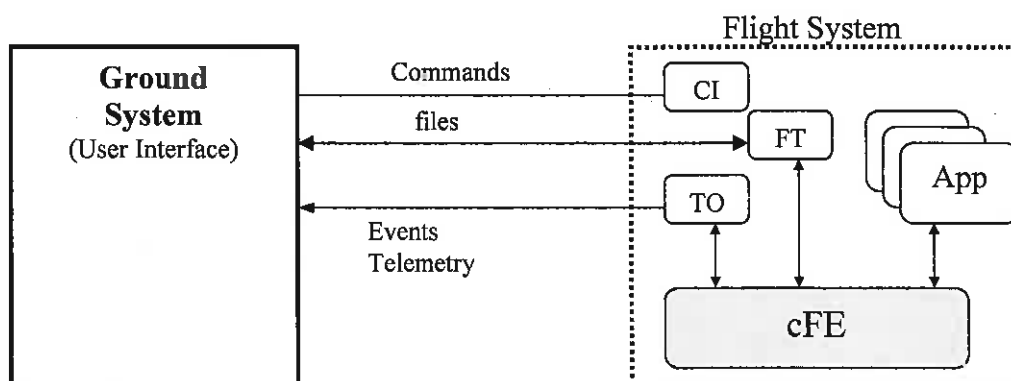


Figure 2-1 cFE Single Flight Processor Context

2.2 TEST HISTORY

cFE 3.3 – Build Verification Testing completed 9/2006 by Walt Moleski
cFE 4.0.0 – Regression Testing completed 12/2006 by Walt Moleski
cFE 4.0.0 – Build Verification Testing completed 3/2007 by Walt Moleski
cFE 4.0.1 – Build Verification Testing completed 4/2007 by Walt Moleski
cFE 4.1.0 – Build Verification Testing completed 7/6/2007 by Walt Moleski
cFE 4.2.0 - Build Verification Testing completed 8/16/2007 by Walt Moleski
cFE 4.2.1 - Build Verification Testing completed 9/24/2007 by Walt Moleski
cFE 5.0.0 - Build Verification Testing completed 11/7/2007 by Walt Moleski
cFE 5.2.0 - Build Verification Testing completed 10/6/2008 by Walt Moleski
cFE 6.0.0 – Build Verification Testing completed 8/18/2009 by Walt Moleski
cFE 6.1.1.0 – Build Verification Testing completed 11/30/2010 by Walt Moleski
cFE 6.2.2.0 – Build Verification Testing completed 10/3/2011 by Walt Moleski
cFE 6.3.1.0 – Build Verification Testing completed 2/21/12 by Walt Moleski
cFE 6.3.2.0 – Build Verification Testing completed 5/1/12 by Walt Moleski
cFE 6.4.0.0 – Build Verification Testing completed 9/24/14 by Walt Moleski
cFE 6.4.1.0 – Build Verification Testing completed 12/4/14 by Walt Moleski

2.3 TESTING OVERVIEW

There are 5 cFE core subsystems that are tested during Build Verification testing. There are a total of 22 test procedures that are executed. These test procedures are modified to test any new capabilities developed in a build as well as DCR fixes that were contained in a build.

For each build prior to cFE 6.0.0, a new test account was created for the testers to use. As of cFE 6.0.0, a single cfe_test account is used. This account runs the Advanced Spacecraft Integration and System Test (ASIST) software and is setup to contain all the files needed to test the cFE. 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.

The following utilities were used during testing:

Name	Description
\$sc_\$cpu_check_sb_msgcnt	Checks if the change in the message count per msg id is as expected.
\$sc_\$cpu_print_sb_pipes	Prints the status of all the test app pipes.
\$scx_\$cpu_print_all_pipes	Prints the SB routing table.
CFE_startup	Directive combines the "start_data_center", "open_tlm", and "open cmd <cpu>" ASIST startup commands.
CFE_shutdown	Directive combines the "close_data_center" and "exit" ASIST shutdown commands.
create_tbl_file_from_cvt	Procedure that creates a load file from the specified arguments and cvt
evs_app_unreg	Procedure that request the generation of one event message which is registered for filtering and one which is not.
evs_ctr_check	To verify application evt msg sent counter EVS msg sent counter and App bin filter ctr.
evs_fltinfo	To output evt msg filter info.
evs_gen_dis_ty	To request generation of event messages while all Evt Msg Tupes are DISABLED
evs_gen_evts	To request generation of evt msgs when requirement cEVS3103 is fully met
evs_gen_no_evts	To request generation of evt msgs while Event Message Generation is DISABLED
evs_mskd_evt	To request generation of evt msgs after change of binary filter mask from 0 to ffff (always filter) for the event message registered for filtering
evs_test_app_info	To provide test application information
FTP_file	To ftp a file to/from the FSW/GSW.
get_file_to_cvt	Procedure to write some specified FSW data to a file and then FTP the file from the FSW hardware to ASIST hardware and load file to the CVT.
get_tbl_to_cvt	Procedure that dumps the specified table from the processor and loads it into the cvt
load_start_app	Procedure to load and start a user application from the /s/opr/accounts/cfebx/apps/cpux directory.
load_table	Procedure that takes the specified file and transfers the file to the specified processor and then issues a TBL_LOAD command using the file.
tst_tbl_apps_start	Procedure that checks if the TST_TBL and TST_TBL2 applications are running and starts them if they are not.
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_tlmupdate	Procedure to wait for a specified telemetry point to update.
ut_tlmwait	Directive that waits for the specified telemetry condition to be met

3 BUILD VERIFICATION TEST PREPARATION

3.1 SCENERIO DEVELOPMENT

There were no new scenarios developed for build verification test 6.4.1.0. All scenarios are stored on the MKS server, in cfe-project test-and-ground directory within the test-review-packages subdirectory in the Scenarios folder.

3.2 PROCEDURE DEVELOPMENT AND EXECUTION

This build test was completed by running 7 test procedures, 4 for Software Bus (SB), and 3 for Table Services (TBL). All test procedures were written using the Spacecraft Test and Operations Language (STOL). The naming convention for files output from these test procedures was: `scx_cpu<#>_<procedure name>_GMT.<ext>`.

In addition to the above tests that were executed on a Big Endian machine, the cFE 6.4.1.0 was compiled and started on a Little Endian machine to ensure that cFE 6.4.1.0 works. The test started the CFS Scheduler (SCH) and Stored Command (SC) applications and verified by inspection that the “patched” default tables for these applications loaded properly. These tables were updated to include use of the conversion macro that was supplied via DCR 22842 described below. Also, these “patched” files will be supplied outside of the cFE 6.4.1.0 release.

3.3 TEST PRODUCTS

Five 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 Standard Formatted Data Unit (SFDU) information (if applicable) contained in the full log.

A Test Report is developed by the tester after build testing is completed. The log files are stored on CFSLab1 in the \$WORK/test_logs/cFE6.4.1.0 folder. The data files generated are stored in the \$WORK/image folder. All test products are maintained on MKS in the cfe-project test-and-ground directory.

4 BUILD VERIFICATION TEST EXECUTION

4.1 TESTBED OVERVIEW

The cFE build verification testbed consists of two ASIST workstations running ASIST version 9.7k and three MPC750 CPU boards running VxWorks 6.4. CPU1 is primarily used for development testing. CPU2 is currently under development and is not being used. CPU3 is used for build verification testing. Figure 4-1 depicts the testbed.

M

Figure 4-1: cFE Build Verification Testbed

4.2 REQUIREMENTS VERIFICATION MATRIX

Subsystem	Requirements Tested Passed	Requirements Tested Failed	Requirements Tested Partially	Total Tested	Deferred	Total
Software Bus (SB)	35	0	0	35	0	35
Tables (TBL)	51	0	0	51	0	51

4.3 REQUIREMENTS PARTIALLY TESTED

No requirements were partially tested.

4.4 REQUIREMENTS DEFERRED

No requirements were deferred. However, only the Software Bus (SB and Table Services (TBL) tests were executed for cFE 6.4.1.0 since the changes did not affect the other subsystems.

5 BUILD VERIFICATON TEST RESULTS

5.1 SOFTWARE BUS SERVICES (SB)

5.1.1 Overall Assessment

During SB build verification testing

- 33 requirements were validated by demonstration
- 2 requirements were validated by analysis
- No new DCRs were generated
- 2 DCRs were verified

5.1.2 Procedure Description

Procedure	Description	Requirements tested
sb_enapipes	The purpose of this test is to verify that the flight software satisfies the requirements relating to enabling pipes.	cSB4000, cSB4003, cSB4004, cSB4005, cSB4007, cSB4007.1, cSB4300, cSB4301, cSB4302, cSB4303, cSB4304, cSB4305, cSB4305.5, cSB4305.6, cSB4306, cSB4307, cSB4308, cSB4309, cSB4701, cSB4704, cSB4705
sb_dispipes	The purpose of this test is to verify that the flight software satisfies the requirements relating to disabling pipes.	cSB4001, cSB4002, cSB4003, cSB4003.1, cSB4005, cSB4008, cSB4008.1, cSB4301, cSB4303, cSB4305.1, cSB4305.3, cSB4305.4, cSB4500, cSB4700, cSB4705, cSB4706
sb_cmds_err	The purpose of this test is to verify that the flight software will reject SB commands with bad data in the command fields.	cSB4004, cSB4005, cSB4305.6, cSB4701
sb_reset	The purpose of this test is to verify that the SB flight software handles a Power-On and Processor reset according to the requirements.	cSB4303, cSB4303.1, cSB4310, cSB4311, cSB4311.1, cSB4500, cSB4501

5.1.3 Analysis Requirements Verification

The following SB requirements were verified using analysis.

Requirement	Description	Status	Justification
cSB4300	The cFE shall provide a zero-copy message transfer mode for intra-processor communication.	Pass	Step 11.0 of the sb_enapipes procedure tests this requirement. The TST_SB application generates an event message that prints the pointer of the SB zero copy message being sent and also generates an event message when the zero copy message is received. The pointers were identical.

cSB4310	Upon receipt of Request the cFE shall free resources allocation for the specified Application	Pass	Step 7.2 of the SB_Reset test procedure sends a command to stop the TST_SB application. When this command executes, there are numerous events generated and contained in the log file indicating that the TST_SB resources were "freed".
---------	---	------	--

5.1.4 DCRs

No DCRs were generated during build testing.

5.1.5 Notes

There were no significant findings and/or anomalies reported during testing.

5.2 TABLE SERVICES (TBL)

5.2.1 Overall Assessment

During this build testing of the TB subsystem:

- 49 requirements were validated by demonstration
- 2 requirements were validated by analysis
- No new DCRs were generated during testing
- 2 DCRs were verified

5.2.2 Procedure Description

Procedure	Description	Requirements tested
tbl_func	The purpose of this test is to verify the functionality of the cFE Table Services commands.	cTBL6000, cTBL6000.5, cTBL6001, cTBL6002, cTBL6002.1, cTBL6002.2, cTBL6003, cTBL6003.1, cTBL6003.1.1, cTBL6003.1.2, cTBL6005, cTBL6005.1, cTBL6006, cTBL6011, cTBL6012, cTBL6012.1, cTBL6012.2, cTBL6012.3, cTBL6300, cTBL6300.1, cTBL6301, cTBL6302, cTBL6302.1, cTBL6302.2, cTBL6303, cTBL6304, cTBL6305, cTBL6305.1, cTBL6305.2, cTBL6306, cTBL6308, cTBL6308.1, cTBL6309, cTBL6310, cTBL6311, cTBL6311.1, cTBL6311.2, cTBL6312, cTBL6700, cTBL6701
tbl_cmding	The purpose of this test is to verify the Table Services commands.	cTBL6000, cTBL6000.1, cTBL6000.2, cTBL6000.3, cTBL6000.4, cTBL6001, cTBL6003, cTBL6007, cTBL6008, cTBL6009, cTBL6010, cTBL6011
tbl_reset	The purpose of this test is to verify that the cFE Table Services (TBL) software meets the requirements defined in the SRS for Power-On and Processor Resets	cTBL6500, cTBL6501, cTBL6501.1

5.2.3 Analysis Requirements Verification

The following TBL requirements were verified using analysis.

Requirement	Description	Status	Justification
cTBL6308.1	If a Table is locked when an update Request is made, an appropriate error code shall be returned to the calling Application and the update shall not occur.	Pass	The error code was printed in the uart window. However, the uart output was not captured. The error indicated that the table was locked and could not be updated. Once the lock was removed, the table was updated appropriately.
cTBL6311.1	Upon providing a calling Application with the addresses of a Tables' data, the cFE shall lock the contents of the Tables to prevent modification.	Pass	Step 18.2 of the TBL_func test procedure attempts to update a table that is shared by another application. The error message displayed indicating that the table did not have any working buffers available to perform the update.

5.2.4 DCRs

No new DCRs were generated.

5.2.5 Notes

There were no significant findings and/or anomalies reported during testing.

5.3 DCRS VERIFIED

The following DCRs were tested and verified during cFE 6.4.1.0 Build Verification testing.

DCR	Description	Test Method	Test Approach
22841	SB – CCSDS Command Secondary Header Redefined	Test Procedure Inspection	Ensure all the SB and TBL tests pass with this change. The Little Endian test was performed in a VM by verifying the SCH and SC default tables loaded on initialization.
22842	SB – Add Big Endian Conversion Macro to ccstds.h	Test Procedure Inspection	Ensure all the SB and TBL tests pass with this change. The Little Endian test was performed in a VM by verifying the SCH and SC default tables loaded on initialization.

APPENDIX A - RTTM

The cFE 6.4.1.0 RTTM can be found on the MKS server, in CFE-Repository test-and-ground directory Results folder.

APPENDIX B - COMMAND, TELEMETRY, AND EVENTS VERIFICATION MATRIX

Command	Test Procedure(s)	Notes/Comments
SB_NOOP	SB_EnablePipe	
SB_ResetCtrs	SB_DisablePipe	
SB_DumpStats	SB_DisablePipe	
SB_WriteRouting2File	SB_Reset, SB_DisablePipe, SB_EnablePipe	
SB_EnaRoute	SB_CmdsErr, SB_Reset, SB_DisablePipe, SB_EnablePipe	
SB_DisRoute	SB_CmdsErr, SB_DisablePipe, SB_EnablePipe	
SB_DumpNetwork	SB_DisablePipe	
SB_WritePipe2File	SB_EnablePipe	
SB_WriteMap2File	SB_DisablePipe	
SB_EnaSubRptg		
SB_DisSubRptg		
SB_SendPrevSubs		
TBL_NOOP	TBL_CMD	
TBL_ResetCtrs	TBL_CMD	
TBL_Load	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_Dump	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_Validate	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_Activate	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_WriteReg2File	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_TLMReg	TBL_Functionality	
TBL_DeleteCDS	TBL_Reset	
TBL_LoadAbort	TBL_CMD, TBL_Functionality	

Telemetry	Test Procedure(s)	Notes/Comments
SB_CMDPC	SB_DisablePipe; SB_Reset	
SB_CMDEC	SB_DisablePipe; SB_Reset	
SB_NoSubEC	SB_DisablePipe; SB_EnablePipe; SB_Reset	
SB_MsgSndEC	SB_DisablePipe; SB_EnablePipe	
SB_MsgRecEC	SB_DisablePipe; SB_Reset	
SB_InternalEC		
SB_NewPipeEC	SB_DisablePipe	
SB_SubscrEC	SB_Reset	
SB_DupSubCnt	SB_Reset	

SB_PipeOvrEC	SB_DisablePipe	
SB_MsgLimEC	SB_DisablePipe	
SB_MemPoolHdl		
SB_MemInUse		
SB_UnmarkedMem		
SB_Stat.SB_SMMIDIU	SB_DisablePipe	
SB_Stat.SB_SMPMIDIU	SB_DisablePipe	
SB_Stat.SB_SMMMIDALW	SB_DisablePipe	
SB_Stat.SB_SMPIU	SB_DisablePipe	
SB_Stat.SB_SMPPIU	SB_DisablePipe	
SB_Stat.SB_SMMPALW	SB_DisablePipe	
SB_Stat.SB_SMBMIU	SB_DisablePipe	
SB_Stat.SB_SMPBMIU	SB_DisablePipe	
SB_Stat.SB_SMMBMALW	SB_DisablePipe	
SB_Stat.SB_SMSIU	SB_DisablePipe	
SB_Stat.SB_SMPSIU	SB_DisablePipe	
SB_Stat.SB_SMMSALW	SB_DisablePipe	
SB_Stat.SB_SMSBBIU	SB_DisablePipe	
SB_Stat.SB_SMPBBIU	SB_DisablePipe	
SB_Stat.SB_SMMPDALW	SB_DisablePipe	
SB_Stat.SB_SMPDS.SB_PDPIPEID	SB_DisablePipe	
SB_Stat.SB_SMPDS.SB_PDDEPTH	SB_DisablePipe	
SB_Stat.SB_SMPDS.SB_PDINUSE	SB_DisablePipe	
SB_Stat.SB_SMPDS.SB_PDPKINUSE	SB_DisablePipe	
TBL_CMDPC	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_CMDEC	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_NumTables	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_NumUpdatesPend		No real way to test this
TBL_ValCompltdCtr	TBL_CMD	
TBL_LastValCRC	TBL_Functionality	
TBL_LastValS	TBL_Reset, TBL_Functionality	
TBL_LastValBuf	TBL_CMD, TBL_Functionality	
TBL_LastValTblName	TBL_Functionality	
TBL_ValSuccessCtr	TBL_CMD	
TBL_ValFailedCtr	TBL_CMD	
TBL_ValReqCtr	TBL_CMD	
TBL_NumFreeShrBuf	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_MemPoolHdl		
TBL_LastUpdTime.TBL_Seconds	TBL_CMD, TBL_Functionality	
TBL_LastUpdTime.TBL_SubSeconds	TBL_CMD, TBL_Functionality	
TBL_LastUpdTblName	TBL_CMD, TBL_Functionality	
TBL_LastFileLoaded	TBL_CMD, TBL_Functionality	
TBL_LastFileDumped	TBL_CMD, TBL_Functionality	

TBL_Size	TBL_Functionality, TBL_Reset	
TBL_CRC		
TBL_ActBufAdd	TBL_Functionality	
TBL_IActBufAdd	TBL_Functionality	
TBL_ValFuncPtr	TBL_Functionality	
TBL_TimeLastUpd.TBL_TLUSecnds	TBL_Functionality	
TBL_TimeLastUpd.TBL_TLUSubSecnds	TBL_Functionality	
TBL_FILECSECONDS	TBL_Functionality	
TBL_FILECSUBSECONDS	TBL_Functionality	
TBL_LoadedOnce	TBL_Functionality	
TBL_UpdatePending	TBL_Functionality	
TBL_DumpOnly	TBL_Reset, TBL_Functionality	
TBL_DblBuffered	TBL_Functionality	
TBL_Name	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_LastFileUpd	TBL_CMD, TBL_Reset, TBL_Functionality	
TBL_OwnerApp	TBL_Functionality	
TBL_CritFlag	TBL_Functionality	

File Telemetry	Test Procedure(s)	Notes/Comments
RF.TBL_Size	TBL_Functionality	
RF.TBL_SysTime.TBL_ST_Seconds	TBL_CMD, TBL_Functionality	
RF.TBL_SysTime.TBL_ST_Subseconds	TBL_CMD, TBL_Functionality	
RF.TBL_NumUsers	TBL_Functionality	
RF.TBL_LoadBufferID	TBL_CMD, TBL_Reset, TBL_Functionality	
RF.TBL_FileCreateSeconds		
RF.TBL_FileCreateSubseconds		
RF.TBL_RegCRC		
RF.TBL_ValFuncPresent	TBL_Functionality	
RF.TBL_LoadedOnce	TBL_Functionality	
RF.TBL_UpdatePndng	TBL_Functionality	
RF.TBL_DumpOnly	TBL_Reset, TBL_Functionality	
RF.TBL_DblBuffered	TBL_Functionality	
RF.TBL_Name	TBL_CMD, TBL_Reset, TBL_Functionality	
RF.TBL_LastFileUpd	TBL_CMD, TBL_Reset, TBL_Functionality	
RF.TBL_OwnerAppName		
RF.TBL_Critical	TBL_Functionality	
SB_RouteEntry.SB_MsgId	SB_DisablePipe; SB_EnablePipe; SB_Reset	
SB_RouteEntry.SB_PipeId	SB_DisablePipe; SB_EnablePipe; SB_Reset	
SB_RouteEntry.SB_State	SB_EnablePipe; SB_Reset	

SB_RouteEntry.SB_MsgCnt	SB_DisablePipe; SB_EnablePipe;	
SB_RouteEntry.SB_AppName	SB_Reset	
SB_RouteEntry.SB_PipeName	SB_DisablePipe; SB_EnablePipe; SB_Reset	
PE.SBPF_InUse		
PE.SBPF_PipeID		
PE.SBPF_PipeName	SB_EnablePipe	
PE.SBPF_AppName		
PE.SBPF_TaskId		
PE.SBPF_SysQId		
PE.SBPF_LastSender		
PE.SBPF_Qdepth		
PE.SBPF_SendErrs		
PE.SBPF_Buffer		
SB_MsgMapEntry.SB_MM_MID		
SB_MsgMapEntry.SB_MM_INDEX		

Id	Event Message	Test Procedure(s)	Notes/Comments
1	CFE_SB_INIT_EID		
2	CFE_SB_CR_PIPE_BAD_ARG_EID	SB_DisablePipe;	
3	CFE_SB_MAX_PIPES_MET_EID	SB_DisablePipe; SB_EnablePipe; SB_Reset;	
4	CFE_SB_CR_PIPE_ERR_EID	SB_DisablePipe; SB_EnablePipe; SB_Reset	
5	CFE_SB_PIPE_ADDED_EID	ES_AppCtrl; ES_Logging; ES_Reset; EVS_BinFiltr; EVS_Cmds; EVS_EvtGen; EVS_Reset; SB_DisablePipe; SB_EnablePipe; SB_Reset; TBL_Cmd; TBL_Functionality; TBL_Reset;	
6	CFE_SB_SUB_ARG_ERR_EID		
7	CFE_SB_DUP_SUBSCRIP_EID	ES_Logging; SB_Reset; TBL_Functionality;	
8	CFE_SB_MAX_MSGS_MET_EID	SB_DisablePipe;	
9	CFE_SB_MAX_DESTS_MET_EID	ES_AppCtrl; SB_DisablePipe; SB_EnablePipe; SB_Reset;	
10	CFE_SB_SUBSCRIPTION_RCVD_EID	ES_AppCtrl; ES_Logging; ES_Reset; EVS_BinFiltr; EVS_Cmds; EVS_EvtGen; EVS_Reset; SB_DisablePipe; SB_EnablePipe; SB_Reset; TBL_Cmd; TBL_Functionality; TBL_Reset;	
11	CFE_SB_UNSUB_ARG_ERR_EID		
12	CFE_SB_UNSUB_NO_SUBS_EID	SB_Reset	
13	CFE_SB_SEND_BAD_ARG_EID		
14	CFE_SB_SEND_NO_SUBS_EID	ES_AppCtrl; SB_EnablePipe; SB_CmdsErr; SB_EnablePipe; SB_Reset	

Id	Event Message	Test Procedure(s)	Notes/Comments
15	CFE_SB_MSG_TOO_BIG_EID	SB_CmdsErr; SB_DisablePipe; SB_EnablePipe	
16	CFE_SB_GET_BUF_ERR_EID		
17	CFE_SB_MSGID_LIM_ERR_EID	ES_AppCtrl; ES_Logging; SB_DisablePipe; SB_EnablePipe	
18	CFE_SB_RCV_BAD_ARG_EID	SB_DisablePipe; SB_Reset;	
19	CFE_SB_BAD_PIPEID_EID	SB_EnablePipe;	
20	CFE_SB_DEST_BLK_ERR_EID		
21	CFE_SB_SEND_INV_MSGID_EID		
22	CFE_SB_SUBSCRIPTION_RPT_EID		
24	CFE_SB_UNSUBSCRIPTION_RPT_EID		
25	CFE_SB_Q_FULL_ERR_EID	SB_DisablePipe	
26	CFE_SB_Q_WR_ERR_EID		
27	CFE_SB_Q_RD_ERR_EID		
28	CFE_SB_CMD0_RCVD_EID	EVS_BinFiltr; EVS_Cmds; EVS_EvtGen; EVS_Reset; SB_CmdsErr; SB_EnablePipe;	
29	CFE_SB_CMD1_RCVD_EID	SB_DisablePipe	
30	CFE_SB_LSTSNDER_ERR1_EID		
31	CFE_SB_LSTSNDER_ERR2_EID		
32	CFE_SB_SND_STATS_EID	SB_DisablePipe; SB_EnablePipe	
33	CFE_SB_ENBL_RTE1_EID	SB_CmdsErr; SB_EnablePipe	
34	CFE_SB_ENBL_RTE2_EID	SB_DisablePipe; SB_EnablePipe	
35	CFE_SB_ENBL_RTE3_EID	SB_CmdsErr; SB_Reset;	
36	CFE_SB_DSBL_RTE1_EID	SB_CmdsErr; SB_EnablePipe	
37	CFE_SB_DSBL_RTE2_EID	SB_DisablePipe; SB_EnablePipe	
38	CFE_SB_DSBL_RTE3_EID	SB_CmdsErr	
39	CFE_SB_SND_RTG_EID	SB_DisablePipe; SB_EnablePipe; SB_Reset;	
40	CFE_SB_SND_RTG_ERR1_EID	SB_DisablePipe; SB_EnablePipe	
41	CFE_SB_GLS_INV_CALLER_EID		
42	CFE_SB_BAD_CMD_CODE_EID	EVS_Cmds; EVS_EvtGen; SB_CmdsErr;	
43	CFE_SB_BAD_MSGID_EID		
44	CFE_SB_FULL_SUB_PKT_EID		
45	CFE_SB_PART_SUB_PKT_EID		
46	CFE_SB_DEL_PIPE_ERR1_EID		
47	CFE_SB_PIPE_DELETED_EID	ES_AppCtrl; ES_Logging; SB_EnablePipe; SB_Reset; TBL_Functionality; TBL_Reset;	
48	CFE_SB_SUBSCRIPTION_REMOVED_EID	ES_AppCtrl; ES_Logging; SB_EnablePipe; SB_Reset; TBL_Functionality; TBL_Reset;	
49	CFE_SB_FILEWRITE_ERR_EID		
50	CFE_SB_SUB_INV_PIPE_EID	SB_Reset;	

Id	Event Message	Test Procedure(s)	Notes/Comments
51	CFE_SB_SUB_INV_CALLER_EID		
52	CFE_SB_UNSUB_INV_PIPE_EID		
53	CFE_SB_UNSUB_INV_CALLER_EID		
54	CFE_SB_DEL_PIPE_ERR2_EID		
1	CFE_TBL_INIT_INF_EID		
10	CFE_TBL_NOOP_INF_EID	EVS_BinFiltr; EVS_Cmds; EVS_EvtGen; EVS_Reset; TBL_Cmd	
11	CFE_TBL_RESET_INF_EID	TBL_Cmd	
12	CFE_TBL_FILE_LOADED_INF_EID	TBL_Cmd; TBL_Functionality; TBL_Reset	
13	CFE_TBL_OVERWRITE_DUMP_INF_EID	TBL_Cmd; TBL_Functionality; TBL_Reset	
14	CFE_TBL_WRITE_DUMP_INF_EID	TBL_Cmd; TBL_Functionality; TBL_Reset	
15	CFE_TBL_OVERWRITE_REG_DUMP_INF_EID	TBL_Cmd; TBL_Functionality; TBL_Reset	
16	CFE_TBL_VAL_REQ_MADE_INF_EID	TBL_Cmd; TBL_Functionality; TBL_Reset	
17	CFE_TBL_LOAD_PEND_REQ_INF_EID	TBL_Cmd; TBL_Functionality; TBL_Reset	
18	CFE_TBL_TLM_REG_CMD_INF_EID	TBL_Functionality	
21	CFE_TBL_LOAD_ABORT_INF_EID	TBL_Cmd; TBL_Functionality	
22	CFE_TBL_WRITE_REG_DUMP_INF_EID	TBL_Cmd; TBL_Functionality; TBL_Reset	
23	CFE_TBL_ASSUMED_VALID_INF_EID	TBL_Functionality	
35	CFE_TBL_LOAD_SUCCESS_INF_EID	ES_AppCtrl; TBL_Cmd; TBL_Functionality; TBL_Reset;	
36	CFE_TBL_VALIDATION_INF_EID	TBL_Cmd; TBL_Functionality; TBL_Reset	
37	CFE_TBL_UPDATE_SUCCESS_INF_EID	TBL_Cmd; TBL_Functionality; TBL_Reset	
38	CFE_TBL_CDS_DELETED_INFO_EID	TBL_Reset;	
50	CFE_TBL_MID_ERR_EID		
51	CFE_TBL_CC1_ERR_EID	EVS_EvtGen; EVS_Cmds; TBL_Cmd;	
52	CFE_TBL_LEN_ERR_EID		
53	CFE_TBL_FILE_ACCESS_ERR_EID	TBL_Cmd; TBL_Functionality	
54	CFE_TBL_FILE_STD_HDR_ERR_EID		
55	CFE_TBL_FILE_TBL_HDR_ERR_EID		
56	CFE_TBL_FAIL_HK_SEND_ERR_EID		
57	CFE_TBL_NO_SUCH_TABLE_ERR_EID	TBL_Functionality; TBL_Reset	
58	CFE_TBL_FILE_TYPE_ERR_EID		
59	CFE_TBL_FILE_SUBTYPE_ERR_EID		
60	CFE_TBL_NO_WORK_BUFFERS_ERR_EID	TBL_Functionality	
61	CFE_TBL_INTERNAL_ERROR_ERR_EID		
62	CFE_TBL_CREATING_DUMP_FILE_ERR_EID	TBL_Functionality	
63	CFE_TBL_WRITE_CFE_HDR_ERR_EID		
64	CFE_TBL_WRITE_TBL_HDR_ERR_EID		
65	CFE_TBL_WRITE_TBL_IMG_ERR_EID		
66	CFE_TBL_NO_INACTIVE_BUFFER_ERR_EID	TBL_Functionality	
67	CFE_TBL_TOO_MANY_VALIDATIONS_ERR_EID		

Id	Event Message	Test Procedure(s)	Notes/Comments
68	CFE_TBL_WRITE_TBL_REG_ERR_EID		
69	CFE_TBL_LOAD_ABORT_ERR_EID		
70	CFE_TBL_ACTIVATE_ERR_EID	TBL_Cmd; TBL_Functionality	
71	CFE_TBL_FILE_INCOMPLETE_ERR_EID		
72	CFE_TBL_LOAD_EXCEEDS_SIZE_ERR_EID	TBL_Cmd; TBL_Functionality	
73	CFE_TBL_ZERO_LENGTH_LOAD_ERR_EID		
74	CFE_TBL_PARTIAL_LOAD_ERR_EID		
75	CFE_TBL_FILE_TOO_BIG_ERR_EID	TBL_Cmd	
76	CFE_TBL_TOO_MANY_DUMPS_ERR_EID		
77	CFE_TBL_DUMP_PENDING_ERR_EID		
78	CFE_TBL_ACTIVATE_DUMP_ONLY_ERR_EID	TBL_Functionality	
79	CFE_TBL_LOADING_A_DUMP_ONLY_ERR_EID	TBL_Functionality	
80	CFE_TBL_ILLEGAL_BUFF_PARAM_ERR_EID	TBL_Functionality;	
81	CFE_TBL_UNVALIDATED_ERR_EID	TBL_Functionality	
82	CFE_TBL_IN_REGISTRY_ERR_EID	TBL_Reset	
83	CFE_TBL_NOT_CRITICAL_TBL_ERR_EID		
84	CFE_TBL_NOT_IN_CRIT_REG_ERR_EID	TBL_Reset	
85	CFE_TBL_CDS_NOT_FOUND_ERR_EID		
86	CFE_TBL_CDS_DELETE_ERR_EID		
87	CFE_TBL_CDS_OWNER_ACTIVE_ERR_EID	TBL_Reset	
88	CFE_TBL_LOADING_PENDING_ERR_EID		
89	CFE_TBL_FAIL_NOTIFY_SEND_ERR_EID		
90	CFE_TBL_REGISTER_ERR_EID	TBL_Functionality; TBL_Reset	
91	CFE_TBL_SHARE_ERR_EID		
92	CFE_TBL_UNREGISTER_ERR_EID		
93	CFE_TBL_LOAD_ERR_EID	TBL_Functionality	
94	CFE_TBL_LOAD_TYPE_ERR_EID		
95	CFE_TBL_UPDATE_ERR_EID		
96	CFE_TBL_VALIDATION_ERR_EID	TBL_Cmd; TBL_Functionality;	
97	CFE_TBL_SPACECRAFT_ID_ERR_EID	TBL_Validate	
98	CFE_TBL_PROCESSOR_ID_ERR_EID	TBL_Validate	