TBD Doc No

Software Test Procedures and Reports for the **Core Flight System SP0-VxWorks6.9 Platform Support Package – Version 1.5.1.0**

Engineering Directorate
Software, Robotics and Simulation Division

Availability:
NASA & NASA contractor employees as required

October 2021
Draft

National Aeronautics and
Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

Software Test Procedures and Reports for the **Core Flight System SP0-VxWorks6.9 Platform Support Package Version 1.5.1.0**

Prepared by:

_____

Tam Ngo/ER6 Date

Approved by:

_____

Mark Markovich/NT Date
Project S&MA Representative

_____

Dave Swartwout/ER6 Date
Project Manager

Approved for Release by:

_____

Pedro Martinez/ER1 Date
Division Approval Authority

National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

Verify this is the correct version before use

Change Record

Revision	*Date*	*Originator*	*Description*
N/A	Oct 2021	Tam Ngo	Initial draft

Table of Contents

List of Tables

1 INTRODUCTION

1.1 Purpose and Scope

This document defines the test plan, test procedures, and test reports for the "certifiable" version 1.5.1.0 of the Core Flight System SP0-VxWorks6.9 Platform Support Package (PSP) as class A, safety critical flight software component. The intent of the "certifiable" version of the SP0-VxWorks6.9 PSP is to provide the users with the necessary certification artifacts, such as test procedures, test suites and expected test results to perform a formal certification on their target platform with minimal changes.

While the verification plan for this software is documented within the cFS Certification Software Development Plan (SDP), this document provides additional test planning details. The test procedures in this document were defined and executed for verification and validation activities on the VxWorks 6.9/SP0-S platform.

1.2 Responsibility and Change Authority

This document is prepared in accordance with EA-WI-025, "GFE Flight Project Software and Firmware Development". The responsibility for the development of this document lies with the Engineering Directorate Software, Robotics, & Simulation Division (SR&SD), Spacecraft Software Engineering Branch/ER6. Change authority is the Software, Robotics and Simulation Division of the Johnson Space Center.

2 APPLICABLE AND REFERENCE DOCUMENTS

2.1 Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this VDD to the extent specified herein.

Table 2-1: Applicable Documents

Document Number	*Document Title*	*Revision / Release Date*
NPR 7150.2	NASA Software Engineering Procedural Requirements	Rev C / Aug 2019
EA-WI-025	GFE Flight Project Software and Firmware Development	Rev D / Sep 2013
GP-10021	cFS Certification Software Development Plan	Baseline / May 2020

2.2 Reference Documents

The following documents are reference documents utilized in the development of this VDD. These documents do not form a part of this VDD and are not controlled by their reference herein.

Table 2-2: Reference Documents

Document Number	*Document Title*	*Revision / Release Date*
TBD	SP0-VxWorks6.9 PSP Software Design Document	TBD
TBD	SP0-VxWorks6.9 PSP Version Description Document	TBD
TBD	SP0-VxWorks6.9 PSP Software Developer's Guide	TBD

2.3 Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence.

All specifications, standards, exhibits, drawings or other documents that are invoked as "applicable" in this specification are incorporated as cited. All documents referred to by an applicable document are considered to be for guidance and information only, with the exception of ICDs which will have their applicable documents considered to be incorporated as cited.

3 TEST PLAN

The Verification and Validation plan for the cFS Certification is contained within the cFS Certification Software Development Plan (SDP). However, a summary and additional test planning, procedures and detailed results for the SP0-VxWorks6.9 PSP are provided here for completeness.

3.1 Testing Activities

The following activities were performed against the code base of the SP0-VxWorks6.9 PSP:

a. Static code analysis, both internal and independent
b. Formal code inspection
c. Full code coverage unit testing
d. Functional testing
e. Integrated vertical validation testing

Items (a) & (b) have been completed by the cFS Certification project. Items (c) & (d) can be repeated by the project. And item (e) is expected to be performed by the project in the context of the overall software system the project is responsible for.

3.2 Requirement Traceability

The SP0-VxWorks6.9 PSP is implemented as a C library which provides a set of Application Programming Interfaces (APIs) that can be used by the cFE, the OSAL and any cFS application/library. They are essentially its "requirements". For purpose of certification, there will not be requirement traceabilities for libraries. And hence, this section is not applicable to the SP0-VxWorks6.9 PSP.

4 VERIFICATION AND VALIDATION PROCESS

4.1 Verification and Validation Management Responsibilities

The project is ultimately responsible for the Verification and Validation of the overall software system that the project is responsible for. The SP0-VxWorks6.9 PSP should be treated as a CSCI of that overall system. The project can integrate the SP0-VxWorks6.9 PSP's certifiable artifacts to conform to the project's overall verification and validation requirements or process.

4.2 Verification Methods

N/A

4.3 Validation Methods

N/A

4.4 Certification Process

Certification of cFS products shall be performed by the project in the context of the overall software system that the project is responsible for.

4.5 Acceptance Testing

Acceptance testing of cFS products shall be performed from the perspective of the overall software system that the project is responsible for.

5 SP0-VXWORKS6.9 PSP TEST PROCEDURES

5.1 Static code analysis

The static code analyzer tool, ***Understand***, was used to perform static code analysis on the SP0-VxWorks6.9 PSP code base against the JSC SSET Coding Standards for C programing language. Any deviation from the coding standard is documented in the cFS Certification SDP document. Since the SP0-VxWorks6.9 PSP source code is available, the project can opt to re-run static code analysis using the project's tool and coding standards for C.

5.2 Full code coverage unit testing

The SP0-VxWorks6.9 PSP code base comes with the source code for unit testing that can be built and run on an SP0-S platform running VxWorks 6.9 OS. All external interfaces are stubbed (including cFE and OSAL interfaces) so that unit tests can be executed as a stand-alone program. This is accomplished with the use of the cFS Unit Test Framework called ***ut_assert*** that is distributed as part of the OSAL code base.

The SP0-VxWorks6.9 PSP implementation is platform-dependent; hence, its unit tests are intended to be built and run on an SP0-S processor running VxWorks6.9 operating system. The Wind River's code coverage tools are used to obtain the code coverage.

5.2.1 Assumptions, Dependencies and Constraints

5.2.1.1 Assumptions

a. The application code base, in its entirety, resides as a sub-directory of the "psp" directory of a cFS workspace. See the recommended cFS workspace from the cFS repository at https://www.github.com/nasa/cfs.
b. There is a target build setup to include the SP0-VxWorks6.9 PSP. See the sample build setup in ***cmake/sample_defs*** directory and the sample top-level build file, ***cmake/Makefile.sample***, from the cFE repository at https://www.github.com/nasa/cfe.

5.2.1.2 Dependencies

a. cFS build system uses cmake, specifically cmake3.

5.2.1.3 Constraints

N/A.

5.2.2 Building the SP0-VxWorks6.9 PSP unit tests

The SP0-VxWorks6.9 PSP's unit tests and code coverage are built and run at the same time on the SP0 processor running VxWorks 6.9 OS.

It is assumed that

a. The SP0 is connected to the local network; and
b. The Wind River's Registry service must be able to communicate with the SP0; and
c. The location of the kernel image file, i.e., "***vxWorks***", is known.

The build steps are captured in a shell script, ***psp/fsw/sp0-vxworks6.9/unit_test/build_psp.sh***.

a. From the host computer, enable the Wind River environment.

   Ex: `$ sh /<VXWORKS_HOME>/wrenv.sh -p vxworks-6.9`

   where <VXWORKS_HOME> is the VxWorks installation directory on the host computer, or use the environment variable, $WIND_HOME, if available.

b. Execute the script, ***psp/fsw/sp0-vxworks6.9/unit_test/build_psp.sh***.

   The script will put the executable in ***psp/fsw/sp0-vxworks6.9/unit_test/payload*** directory.

5.2.3 Running the SP0-VxWorks6.9 PSP unit tests

The execution steps are captured in a shell script, ***psp/fsw/sp0-vxworks6.9/unit_test/run_psp.sh***.

a. Execute the script, ***psp/fsw/sp0-vxworks6.9/unit_test/run_psp.sh***, like below:

```
$ sh psp/fsw/unit_test/run_psp.sh <SP0-IP> <kernel-image-file>
```

where <SP0-IP> is the IP address of the SP0, and <kernel-image-file> is the full path to the kernel image.

Ex: `$sh psp/fsw/sp0-vxworks6.9/unit_test/run_psp.sh  192.110.22.10 ~/VxWorks6.9/prebuilt-kernels/VxWorks`

The script will establish a connection to the SP0 using the Wind River's Registry service, "***wtxregd***". Then the executable, as described in section 5.2.2, is loaded and executed on the SP0. A timer of 25 seconds is set, waiting for the unit test to complete. Once done, the results are obtained using the Wind River tool, "***coverageupload***", and get converted to HTML using the Wind River tool, "***coverageconvert***".

b. To view the code-coverage results in a browser,

The output HTML report can be viewed via a browser in the ***psp/fsw/sp0-vxworks6.9/unit_test/html*** directory.

5.2.4 Verifying the SP0-VxWorks6.9 PSP unit test results against the expected test results

Once built and run, the test results can be verified against the provided expected results. See Table 6-1 below for the file name and location of the expected unit test results and code coverage results.

5.3 Functional testing

For libraries, functional testing is done in place of verification testing.

5.3.1 Setting up the test rig

TBD.

5.3.2 Building the SP0-VxWorks6.9 PSP functional tests

TBD.

5.3.3 Running the SP0-VxWorks6.9 PSP functional tests

5.3.4 Verifying the SP0-VxWorks6.9 PSP functional test results against the expected test results

Once built and run, the test results can be verified against the provided expected results. See Table 6-1 below for the file name and location of the expected test results.

6 SP0-VXWORKS6.9 PSP CERTIFICATION ARTIFACTS

Table 6.1 lists the available certification artifacts for the SP0-VxWorks6.9 PSP.

Table 6-1: Certification Artifacts

Testing Type	*Artifact Description*	*Artifact Location*
	Unit test source code and build file	psp/fsw/sp0-vxworks6.9/unit_test/*

Unit testing with code coverage	Unit test expected results	psp/fsw/sp0-vxworks6.9/unit_test/psp_sp0-vxworks6.9_ut_results_expected.log
	Code coverage expected results	psp/fsw/sp0-vxworks6.9/unit_test/psp_sp0-vxworks6.9_ut_coverage_expected.zip
	Functional tests	psp/fsw/sp0-vxworks6.9/functional_test/*
	Functional test expected results	psp/fsw/sp0-vxworks6.9/functional_test/tte_lib_ft_results_expected.log

7 APPENDICES

7.1 ABBREVIATIONS AND ACRONYMS

Term	*Definition*
API	Application Programming Interface
CBCS	Computer Based Control System
CDS	cFE Critical Data Storage
cFE	Core Flight Executive
cFS	Core Flight System
ES	cFE Executive Services
EVS	cFE Event Services
GFE	Government Furnished Equipment
GSFC	Goddard Space Flight Center
ICD	Interface Control Document
ISR	Interrupt Service Routine
JSC	Johnson Space Center
LRO	Lunar Reconnaissance Orbiter
MDT	SCH_TT Message Definition Table
Msg	Message
PMP	Project Management Plan
SB	cFE Software Bus
SBNG	Software Bus Network for Gateway cFS application
SCH_TT	Time-Triggered Ethernet Scheduler cFS application
SDD	Software Design Document
SDG	Software Developer's Guide
SDP	Software Development Plan
SDT	SCH_TT Schedule Definition Table
SRS	Software Requirements Specification
SR&SD	JSC Engineering Directorate Software, Robotics & Simulation Division
SSET	JSC Spacecraft Software Engineering Team
STP	Software Test Plan
SUG	Software User's Guide
TTE	Time-Triggered Ethernet
TTE_LIB	Time-Triggered Ethernet cFS Library
TTE_MGR	Time-Triggered Ethernet Manager cFS application
VDD	Version Description Document

7.2 Definition of Terms

Terms	*Definition*
V&V Terms	
Certification	The audit process by which the body of evidence that results from the verification activities presented are provided to the appropriate certifying authority to indicate all requirements are met.
Deviation	Written authorization issued "before the fact" to develop a product that departs from established requirements.
HSI1	Hardware/software integration (HSI) that is performed prior to PDR. This testing establishes confidence that the hardware and software design concepts are adequate to meet functional interfaces.
HSI2	Hardware/software integration that is performed prior to CDR on engineering unit or DVTU hardware. This testing establishes confidence that the hardware and software detailed designs meets requirements.
Validation	The process that ensures a system meets the customer/sponsor's expectations for intended use. Unique validation activities may not be required if validation is satisfied through verification or acceptance testing activities.
Verification	A formal process, using the method of test, analysis, inspection or demonstration, to confirm that a system and its hardware and software components satisfy all specified performance and operational requirements.
Waiver	Written authorization to temporarily accept an item that departs from a particular performance or design requirement of a specification, drawing, or other contract document. The authorization is granted for a specific number of items and/or a specific period of time. The item(s) is/are considered suitable for use "as is" for a specified period of time or quantity of items, until reworked by approved method..
Types of V&V Methods	
Test	A method of verification wherein formal project requirements (performance, environment, etc.) are verified by measurement or functional test during or after the controlled application of functional and/or environmental stimuli. These measurements may require the use of laboratory equipment, recorded data, procedures, test support items, or specialized software.
Analysis	A verification method utilizing techniques and tools such as math models, prior test data, simulations, analytical assessments, etc. Verification by similarity is acceptable if the subject article is similar or identical in design, manufacturer, manufacturing process, and quality control to another article that has been previously verified to equivalent or more stringent criteria.
Inspection	A method of verification of physical characteristics that determines compliance without the use of special laboratory equipment, procedures, test

	support items, or services. Inspection uses standard methods such as visual gauges, etc. to verify compliance with design requirements.
Demonstration	A qualitative method of verification that evaluates the properties of the subject end item. Demonstration is used with or without special test equipment or instrumentation to verify required characteristics such as operational performance, human engineering features, service and access features, transportability, and displayed data.
Testing Levels	
Development Testing	
Qualification Testing	
Acceptance Testing	
System Level Testing	
End-to-End Testing	
Types of Test Articles	
Prototype Unit	The breadboard, generic component or developmental assembly of hardware and software that roughly performs the basic functions of the engineering unit but is not fully functional equivalent. This unit is used for proof of concept testing of the preliminary design.
Engineering Unit	The hardware, firmware, and software unit that is functionally equivalent to the qualification unit, but not necessarily form and fit equivalent. This unit is used for proof of concept testing of the detailed design. It may be used for software verification credit after CDR with quality controls as defined in the Software Development Plan.
Design Verification Test Unit (DVTU)	The hardware, firmware, and software unit which is form, fit and functional equivalent to the flight unit, but may not be manufactured using the exact flight parts. This unit is used for design proof of concept.
Qualification Unit	
Flight Unit	
Proto-flight Unit	
Ground Support Equipment	

8 NOTES

None.