***A Project Document of the***

***ATC Application Programming Interface (API) Working Group***

ATC APIVS TCS v01.04

Test Case Specifications (TCS) for the Advanced Transportation Controller (ATC) Application Programming Interface Validation Suite (APIVS)

**July 14, 2016**

**In support of:** USDOT Contract # DTFH61-11-D-00052, Task Order # T-13-003

**For use by:** Siva Narla, Chief Engineer and ITS Standards Manager

Institute of Transportation Engineers

George Chen and Douglas Tarico, Co-Chairs

ATC API Working Group

Ralph W. Boaz, Project Manager and Systems Engineer

ATC API Reference Implementation Project

Members of the ATC API Working Group

Consulting Team for the ATC API RI Project

**Prepared by:** James Kinnard, Test Engineer

Adaptive Solutions, Inc.

Copyright 2015-2016 AASHTO/ITE/NEMA. All rights reserved.

**CHANGE HISTORY**

|  |  |
| --- | --- |
| **DATE** | **NOTE** |
| 10/7/15 | Initial Draft TCS v01.00 |
| 11/8/15 | TCS v01.01 |
| 12/1/15 | TCS v01.02 |
| 2/22/16 | TCS v01.03 (TRR) |
| 7/14/16 | TCS v01.04 (TRR2) |
|  |  |

**NOTICE**

**Joint NEMA, AASHTO and ITE Copyright and**

**Intelligent Transportation Systems (ITS) Working Group**

These materials are delivered "AS IS" without any warranties as to their use or performance.

AASHTO/ITE/NEMA AND THEIR SUPPLIERS DO NOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THESE MATERIALS. AASHTO/ITE/NEMA AND THEIR SUPPLIERS MAKE NO WARRANTIES, EXPRESSED OR IMPLIED, AS TO NON-INFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AASHTO, ITE, NEMA, OR THEIR SUPPLIERS BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY CLAIM OR FOR ANY CONSEQUENTIAL, INCIDENTAL, OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS ARISING FROM YOUR REPRODUCTION OR USE OF THESE MATERIALS, EVEN IF AN AASHTO, ITE, OR NEMA REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential, or special damages, or exclusion of implied warranties, so the above limitations may not apply to you.

Use of these materials does not constitute an endorsement or affiliation by or between AASHTO, ITE, or NEMA and you, your company, or your products and services.

If you are not willing to accept the foregoing restrictions, you should immediately return these materials.

ATC is a trademark of NEMA/AASHTO/ITE.

**CONTENTS**

[1 INTRODUCTION 5](#_Toc456255063)

[2 TEST CASE SPECIFICATIONS 6](#_Toc456255064)

**[2.1](#_Toc456255065)** [Common Elements Required by All Test Case Specifications 6](#_Toc456255065)

**[2.2](#_Toc456255066)** [Filename Conventions 7](#_Toc456255066)

**[2.3](#_Toc456255067)** [Test Case Specification 1 – FPUI Library C Functions Present 8](#_Toc456255067)

**[2.4](#_Toc456255068)** [Test Case Specification 2 – FPUI Library C Function Conforming Arguments 10](#_Toc456255068)

**[2.5](#_Toc456255069)** [Test Case Specification 3 – FPUI Library C Function Error and Argument Boundary Checking 11](#_Toc456255069)

**[2.6](#_Toc456255070)** [Test Case Specification 4 – FPUI Library Composite Testing 12](#_Toc456255070)

**[2.7](#_Toc456255071)** [Test Case Specification 5 – Front Panel Manager Window Testing 13](#_Toc456255071)

**[2.8](#_Toc456255072)** [Test Case Specification 6 – FIO Library C Functions Present 15](#_Toc456255072)

**[2.9](#_Toc456255073)** [Test Case Specification 7 – FIO Library C Function Conforming Arguments 16](#_Toc456255073)

**[2.10](#_Toc456255074)** [Test Case Specification 8 – FIO Library C Function Error and Argument Boundary Checking 17](#_Toc456255074)

**[2.11](#_Toc456255075)** [Test Case Specification 9 – FIO Library Composite Testing 18](#_Toc456255075)

**[2.12](#_Toc456255076)** [Test Case Specification 10 – TOD Library C Functions Present 19](#_Toc456255076)

**[2.13](#_Toc456255077)** [Test Case Specification 11 – TOD Library C Function Conforming Arguments 20](#_Toc456255077)

**[2.14](#_Toc456255078)** [Test Case Specification 12 – TOD Library C Function Error and Argument Boundary Checking 21](#_Toc456255078)

**[2.15](#_Toc456255079)** [Test Case Specification 13 – TOD Library Composite Testing 22](#_Toc456255079)

**[2.16](#_Toc456255080)** [Test Case Specification 14 – Multiple and Concurrent Applications 23](#_Toc456255080)

**[2.17](#_Toc456255081)** [Test Case Specification 15 – APIVS Software Licensing Details 24](#_Toc456255081)

**[2.18](#_Toc456255082)** [Test Case Specification 16 – C Programming and Source Code Quality 25](#_Toc456255082)

**[2.19](#_Toc456255083)** [Test Case Specification 17 – XML Scripting, Execution and Logging 26](#_Toc456255083)

# INTRODUCTION

This document, *Test Case Specifications (TCS) for the Advanced Transportation Controller (ATC) Application Programming Interface Validation Suite (APIVS)*, provides the specific test cases necessary to fully test the required features of the API Validation Suite Engine (VSE) application and its associated files.

These test cases have been developed as part of the “Reference Implementation of ATC 5401 Application Programming Interface (API) Standard Version 2” project funded by the USDOT Contract Number DTFH61-11-D-00052, Work Order T-13003 (referred to as the APIRI project).

# TEST CASE SPECIFICATIONS

This section contains the individual Test Case Specifications referred to in the document *Test Plan for the*

*Advanced Transportation Controller (ATC) Application Programming Interface Validation Suite (APIVS).*

## Common Elements Required by All Test Case Specifications

All of the test cases included in this document utilize a single hardware and software platform which is common to all tests. The compliance output files produced by all test cases also have a consistent format which allows pass/fail status to be easily ascertained.

To reduce the size of this document and to ease future maintenance, these common elements are described in the following subsections. Test cases which utilize these common elements will not contain identical subsections in their individual descriptions. If a test case deviates from any of these common elements the appropriate detail will appear in an appropriately-named subsection within that test case.

#### Hardware

All test cases utilize the hardware environment as described in the APIVS Test Plan, specifically:

* an ATC Controller with a primary USB port capable of running startup scripts and a minimum 8x40 character LCD display and associated keyboards
* a Personal Computer (PC) with 1GB available hard drive storage and an available USB port
* a 1GB USB Flash Drive, formatted with a suitable FAT file system

#### Software

All test cases require the porting of APIRI and APIVS software packages to the Engine Board platform used by the ATC Controller under test. For more information regarding obtaining the source code and building the APIRI and APIVS software components, please refer to the *APIRI User Manual* and *APIVS User Manual* references in the APIRI Test Plan.

Prior to the execution of any test, the runtime APIVS package must be copied into the root directory of the USB Flash Drive. This package contains the executable VSE program and all configuration, script and data files necessary to execute all test cases using the associated test procedure(s).

#### Pass/Fail Criteria

Each test case execution produces an output compliance report (file) in XML format. This file contains an element (RunResult) toward the end of the file which indicates the overall completion status (PASS/FAIL) of the test.

<RunResult date="2015-10-01 12:33 PM EST" status="PASS" />

-OR-

<RunResult date="2015-10-01 12:33 PM EST" status="FAIL" />

If the test fails and the test execution was performed with detailed logging enabled, the file can be examined in more detail to determine the exact cause of the failure.

As this output XML file is textual, it can be viewed with a simple text editor such as Notepad. For a more structured view of the XML content in the file, there are many XML file viewing applications available, such as XML Notepad 2007, which can be downloaded via the Internet and installed on the test PC. There are also web sites available, such as <http://xmlgrid.net>, which provide customizable XML graphical views

## Filename Conventions

A naming convention has been established for all files associated with individual APIVS test cases:

***Cnnnn\_xxxyyy.zzz***

where,

***C*** indicates a file associated with an APIVS test case

***nnnn*** is the test case number (from the Test Case Specification Identifier)

***xxx*** is an identifier for the specific file content:

**in** APIVSXML test script (XML format)

**log** conformance report (XML format)

**key** keystroke to Front Panel input file (VSE flat file format)

**vd** Virtual Display compare file (VSE flat file format)

**cmd** SDLC command message file (VSE flat file format)

**rsp** SDLC response message file (VSE flat file format)

***yyy*** is an (optional) numeric identifier, generally for VSE flat files only

***zzz*** is the standard file type (txt or xml)

VSE configuration files, which can be shared between test cases, following the naming convention ***VS\_config\_nnn.txt***, where nnn is a number from 1-999 indicating the specific configuration file to be used.

## Test Case Specification 1 – FPUI Library C Functions Present

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1110.

### Objective and Test Items

This test case validates that each FPUI function defined in Section 4.1 of the ATC 5401 Standard is present in the API software (APIVS SRS Section 3.21.1).

This test case also confirms VS Requirement 3.7: that the APIVS software package is designed to run on an ATC 5201 Standard conforming controller.

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1110\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_null.txt** parameter load file (null data)

**Cxxxx\_key0.txt** keystroke file (Key ‘0’)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1110\_log.xml** conformance report (XML format)

### Environmental Needs

#### Hardware

This test case utilizes the standard APIVS test hardware configuration as detailed in Section 2.1.

#### Software

This test case utilizes the standard APIVS test software as detailed in Section 2.1.

#### Other

None.

### Special Procedural Requirements

None.

### Intercase Dependencies

None.

### Additional Pass/Fail Criteria

None.

## Test Case Specification 2 – FPUI Library C Function Conforming Arguments

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1120.

### Objective and Test Items

This test case validates that each FPUI function has the correct function arguments as defined in Section 4.1 of the ATC 5401 Standard (APIVS SRS Section 3.21.2).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1120\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_null.txt** parameter load file (null data)

**Cxxxx\_key0.txt** keystroke file (Key ‘0’)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1120\_log.xml** conformance report (XML format)

## Test Case Specification 3 – FPUI Library C Function Error and Argument Boundary Checking

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1130.

### Objective and Test Items

This test case validates that each FPUI function returns the correct error codes for the error conditions defined in Section 4.1 of the ATC 5401 Standard (APIVS SRS Section 3.21.3).

This test case also validates that the boundaries of the arguments to the FPUI functions operate as defined in Section 4.1 of the ATC 5401 Standard (APIVS SRS Section 3.21.4).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1130\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_null.txt** parameter load file (null data)

**Cxxxx\_key0.txt** keystroke file (Key ‘0’)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1130\_log.xml** conformance report (XML format)

## Test Case Specification 4 – FPUI Library Composite Testing

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1150.

### Objective and Test Items

This test case validates that each FPUI function operates correctly under typical operating conditions with other API functions using at least one composite test (APIVS SRS Section 3.21.5).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1150\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_key0.txt** keystroke file (Key ‘0’)

**Cxxxx\_key1.txt** keystroke file (Key ‘1’)

**Cxxxx\_vd\_clr.txt** Virtual Display compare file (clear display)

**C1150\_vd.txt** Virtual Display compare file

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1150\_log.xml** conformance report (XML format)

## Test Case Specification 5 – Front Panel Manager Window Testing

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1160.

### Objective and Test Items

This test case validates that the Front Panel Manager Window operates per the requirements established in Section 3.1.1.1 of the ATC 5401 Standard (APIVS SRS Section 3.21.6).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1160\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_keyStar.txt** keystroke file (Key ‘\*’)

**Cxxxx\_keyESC.txt** keystroke file (Key ‘<Esc>’)

**Cxxxx\_keyDN.txt** keystroke file (Key ‘<Down>’)

**Cxxxx\_key0.txt** keystroke file (Key ‘0’)

**Cxxxx\_key1.txt** keystroke file (Key ‘1’)

**Cxxxx\_key2.txt** keystroke file (Key ‘2’)

**Cxxxx\_key3.txt** keystroke file (Key ‘3’)

**Cxxxx\_key4.txt** keystroke file (Key ‘4’)

**Cxxxx\_key5.txt** keystroke file (Key ‘5’)

**Cxxxx\_key6.txt** keystroke file (Key ‘6’)

**Cxxxx\_key7.txt** keystroke file (Key ‘7’)

**Cxxxx\_key8.txt** keystroke file (Key ‘8’)

**Cxxxx\_key9.txt** keystroke file (Key ‘9’)

**Cxxxx\_keyA.txt** keystroke file (Key ‘A’)

**Cxxxx\_keyB.txt** keystroke file (Key ‘B’)

**Cxxxx\_keyC.txt** keystroke file (Key ‘C’)

**Cxxxx\_keyD.txt** keystroke file (Key ‘D’)

**Cxxxx\_keyE.txt** keystroke file (Key ‘E’)

**Cxxxx\_keyF.txt** keystroke file (Key ‘F’)

**C1160\_vd\_fpm\_clr.txt** Virtual Display compare file (clear display)

**C1160\_vd\_fpm\_1.txt** Virtual Display compare file (display 1)

**C1160\_vd\_fpm\_2.txt** Virtual Display compare file (display 2)

**C1160\_vd\_00.txt** Virtual Display compare file (App ‘0’)

**C1160\_vd\_01.txt** Virtual Display compare file (App ‘1’)

**C1160\_vd\_02.txt** Virtual Display compare file (App ‘2’)

**C1160\_vd\_03.txt** Virtual Display compare file (App ‘3’)

**C1160\_vd\_04.txt** Virtual Display compare file (App ‘4’)

**C1160\_vd\_05.txt** Virtual Display compare file (App ‘5’)

**C1160\_vd\_06.txt** Virtual Display compare file (App ‘6’)

**C1160\_vd\_07.txt** Virtual Display compare file (App ‘7’)

**C1160\_vd\_08.txt** Virtual Display compare file (App ‘8’)

**C1160\_vd\_09.txt** Virtual Display compare file (App ‘9’)

**C1160\_vd\_10.txt** Virtual Display compare file (App ‘A’)

**C1160\_vd\_11.txt** Virtual Display compare file (App ‘B’)

**C1160\_vd\_12.txt** Virtual Display compare file (App ‘C’)

**C1160\_vd\_13.txt** Virtual Display compare file (App ‘D’)

**C1160\_vd\_14.txt** Virtual Display compare file (App ‘E’)

**C1160\_vd\_15.txt** Virtual Display compare file (App ‘F’)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1160\_log.xml** conformance report (XML format)

## Test Case Specification 6 – FIO Library C Functions Present

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1310.

### Objective and Test Items

This test case validates that each FIO function defined in Section 4.1 of the ATC 5401 Standard is present in the API software (APIVS SRS Section 3.22.1).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1310\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_null.txt** parameter load file (null data)

**Cxxxx\_rsp177.txt** FIO Response Message data (Frame 177)

**Cxxxx\_rsp181a.txt** FIO Response Message data (Frame 181)

**Cxxxx\_rsp183.txt** FIO Response Message data (Frame 183)

**Cxxxx\_cmd60.txt** FIO Command Message data (Frame 60)

**Cxxxx\_inputs\_null.txt** parameter load file (null input data)

**Cxxxx\_outputs\_null.txt** parameter load file (null output data)

**Cxxxx\_channels\_null.txt** parameter load file (null channel data)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1310\_log.xml** conformance report (XML format)

## Test Case Specification 7 – FIO Library C Function Conforming Arguments

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1320.

### Objective and Test Items

This test case validates that each FIO function has the correct arguments as defined in Section 4.1 of the ATC 5401 Standard (APIVS SRS Section 3.22.2).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1320\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_null.txt** parameter load file (null data)

**Cxxxx\_rsp177.txt** FIO Response Message data (Frame 177)

**Cxxxx\_rsp181a.txt** FIO Response Message data (Frame 181)

**Cxxxx\_rsp183.txt** FIO Response Message data (Frame 183)

**Cxxxx\_cmd60.txt** FIO Command Message data (Frame 60)

**Cxxxx\_inputs\_null.txt** parameter load file (null input data)

**Cxxxx\_outputs\_null.txt** parameter load file (null output data)

**Cxxxx\_channels\_null.txt** parameter load file (null channel data)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1320\_log.xml** conformance report (XML format)

## Test Case Specification 8 – FIO Library C Function Error and Argument Boundary Checking

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1330.

### Objective and Test Items

This test case validates that each FIO function returns the correct error codes for the error conditions defined in Section 4.1 of the ATC 5401 Standard (APIVS SRS Section 3.22.3).

This test case also validates that that the boundaries of the arguments to the FIO functions operate as defined in Section 4.1 of the ATC 5401 Standard (APIVS SRS Section 3.22.4).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1330\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_null.txt** parameter load file (null data)

**Cxxxx\_rsp177.txt** FIO Response Message data (Frame 177)

**Cxxxx\_rsp181a.txt** FIO Response Message data (Frame 181)

**Cxxxx\_rsp183.txt** FIO Response Message data (Frame 183)

**Cxxxx\_cmd60.txt** FIO Command Message data (Frame 60)

**Cxxxx\_inputs\_null.txt** parameter load file (null input data)

**Cxxxx\_outputs\_null.txt** parameter load file (null output data)

**Cxxxx\_channels\_null.txt** parameter load file (null channel data)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1330\_log.xml** conformance report (XML format)

## Test Case Specification 9 – FIO Library Composite Testing

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1350.

### Objective and Test Items

This test case validates that each FIO function operates correctly under typical operating conditions with other API functions using at least one composite test (APIVS SRS Section 3.22.5).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1350\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_rsp177.txt** FIO Response Message data (Frame 177)

**Cxxxx\_rsp181a.txt** FIO Response Message data (Frame 181)

**Cxxxx\_rsp183.txt** FIO Response Message data (Frame 183)

**Cxxxx\_cmd49a.txt** FIO Command Message data (Frame 49)

**Cxxxx\_cmd49b.txt** FIO Command Message data (Frame 49)

**Cxxxx\_cmd53.txt** FIO Command Message data (Frame 53)

**Cxxxx\_cmd55a.txt** FIO Command Message data (Frame 55)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1350\_log.xml** conformance report (XML format)

## Test Case Specification 10 – TOD Library C Functions Present

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1410.

### Objective and Test Items

This test case validates that each TOD function defined in Section 4.1 of the ATC 5401 Standard is present in the API software (APIVS SRS Section 3.23.1).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1410\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_null.txt** parameter load file (null data)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1410\_log.xml** conformance report (XML format)

## Test Case Specification 11 – TOD Library C Function Conforming Arguments

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1420.

### Objective and Test Items

This test case validates that each TOD function has the correct arguments as defined in Section 4.1 of the ATC 5401 Standard (APIVS SRS Section 3.23.2).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1420\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_null.txt** parameter load file (null data)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1420\_log.xml** conformance report (XML format)

## Test Case Specification 12 – TOD Library C Function Error and Argument Boundary Checking

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1430.

### Objective and Test Items

This test case validates that each TOD function returns the correct error codes for the error conditions defined in Section 4.1 of the ATC 5401 Standard (APIVS SRS Section 3.23.3).

This test case also validates that the boundaries of the arguments to the TOD functions operate as defined in Section 4.1 of the ATC 5401 Standard (APIVS SRS Section 3.23.4).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1430\_in.xml** APIVSXML test script (XML format)

**Cxxxx\_null.txt** parameter load file (null data)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1430\_log.xml** conformance report (XML format)

## Test Case Specification 13 – TOD Library Composite Testing

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1450.

### Objective and Test Items

This test case validates that each TOD function operates correctly under typical operating conditions with other API functions using at least one composite test (APIVS SRS Section 3.23.5).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1450\_in.xml** APIVSXML test script (XML format)

**VS\_config\_1.txt** VSE configuration file (specified on VSE command line)

### Output Specifications

This test case produces the following file(s) as output:

File Description

**C1450\_log.xml** conformance report (XML format)

## Test Case Specification 14 – Multiple and Concurrent Applications

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.1510.

### Objective and Test Items

This test case validates that multiple application programs, running concurrently, can exercise the Front Panel Manager Window, the Field I/O Manager functions and the Time of Day functions simultaneously (APIVS SRS Section 3.24).

### Input Specifications

This test case requires the following file(s) as input:

File Description

**runVS1510** Linux shell script

### Output Specifications

This test case produces no output.

## Test Case Specification 15 – APIVS Software Licensing Details

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.6010.

### Objective and Test Items

This test case validates the following APIVS software licensing requirements:

[3.1] All APIVS software shall be free of licensing fees and available for free download.

[3.2] All APIVS software source code shall be available under open source licensing terms.

[3.3] All APIVS software source code shall be available under open source licensing terms approved by ITE.

[3.4] All APIVS software shall be available under open source licensing terms which allow unrestricted use.

[3.5] All APIVS software source code shall be available under open source licensing terms (Gnu Public License) which require modifications and derived works to be distributed under the same terms.

[3.6] All APIVS software source code shall be available under the terms of the Gnu Public License (GPL).

### Input Specifications

This test case requires one or more randomly selected source files from the APIVS Software distribution as input.

### Output Specifications

This test case produces no output.

### Environmental Needs

#### Hardware

This test case utilizes the standard APIVS test hardware configuration as detailed in Section 2.1.

#### Software

This test case requires a copy of the APIVS source code from the online repository.

#### Other

This test case requires Internet access for the Test PC in order to retrieve the APIVS Software distribution from the online repository (GitHub) at <https://github.com/apiriadmin/APIVS>.

### Additional Pass/Fail Criteria

All requirements in Section 2.17.2 must be satisfied for this test case to pass.

## Test Case Specification 16 – C Programming and Source Code Quality

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.6020.

### Objective and Test Items

This test case validates the following APIVS requirements related to C programming and source code quality:

[3.8] All APIVS software programs are written in the C Programming Language and are designed to be compatible with the uClibc C library.

[3.9] All APIVS source code is written to follow the styling practices of the Linux kernel and GNU standard library source code.

### Input Specifications

This test case requires one or more randomly selected source files from the APIVS Software distribution as input. It also requires an examination of file C6020\_log.txt to confirm that the VSE is compatible with the uClibc library.

### Output Specifications

This test case produces no output.

### Environmental Needs

#### Hardware

This test case utilizes the standard APIVS test hardware configuration as detailed in Section 2.1.

#### Software

This test case requires a copy of the APIVS source code from the online repository.

#### Other

This test case requires access to the APIVS Software distribution from the online repository (GitHub) at <https://github.com/apiriadmin/APIVS>.

### Additional Pass/Fail Criteria

All requirements in Section 2.18.2 must be satisfied for this test case to pass.

## Test Case Specification 17 – XML Scripting, Execution and Logging

### Test Case Specification Identifier

The identifier for this Test Case Specification is APIVS.TCS.6030.

### Objective and Test Items

This test case validates the following APIVS requirements related to the VSE (Validation Suite Engine) execution, including details regarding the XML input file, the VSE execution options, and the format and content of the XML output conformance log file.

[3.10] The APIVS software allows validation tests to be configured using XML files. Details of the APIVSXML language are found in *API Validation Suite APIVSXML Specification.*

[3.11] XML TCS files may be established and interpreted by the APIVS software at run-time without requiring recompilation.

[3.12] The VSE executable may be configured to run through all tests supplied in the distribution by use of the command line switch enabling this option.

[3.13] The VSE executable may be configured to run one or more individual tests from the full set of tests supplied in the distribution.

[3.14] The VSE executable may be configured to run each test load once or continuously.

[3.15] The VSE executable returns a value of 0 to indicate a pass or conformance condition for the test run.

[3.16] The VSE executable returns a value of -1 to indicate a failure or non-conformance condition for the test run.

[3.17] When configured for verbose output, the APIVS software returns a detailed conformance report.

[3.18]

[3.19]

[3.20] The XML format of the conformance report allows interactive viewing of test results at increasing levels of detail.

### Input Specifications

This test case requires the following file(s) as input:

File Description

**C1110\_in.xml** APIVSXML test script (XML format)

**C1110\_log\_L1.xml** conformance report Level 1 (XML format)

**C1110\_log\_L2.xml** conformance report Level 2 (XML format)

**C1110\_log\_L3.xml** conformance report Level 3 (XML format)

**runAPIVS** Linux script file (VSE execution control)

### Output Specifications

This test case produces no output.

### Environmental Needs

#### Hardware

This test case utilizes the standard APIVS test hardware configuration as detailed in Section 2.1.

#### Software

This test case utilizes the standard APIVS test software as detailed in Section 2.1.

This test case requires the examination of XML files using a suitable text editor. A good choice for this task is NotePad++ (<https://notepad-plus-plus.org>), which provides context-sensitive formatting for XML files.

#### Other

This test case requires Internet access for the Test PC in order to access a suitable online multi-level XML file viewer (recommended viewer is <http://xmlgrid.net>).

### Additional Pass/Fail Criteria

All requirements in Section 2.19.2 must be satisfied for this test case to pass.