***A Project Document of the***

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

ATC APIVS TPS v01.04

Test Procedure Specifications (TPS) 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 TPS v01.00 |
| 11/8/15 | TPS v01.01 |
| 12/1/15 | TPS v01.02 |
| 2/22/16 | TPS v01.03 (TRR) |
| 7/14/16 | TPS 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](#_Toc456255098)

[2 TEST PROCEDURE SPECIFICATIONS 5](#_Toc456255099)

**[2.1](#_Toc456255100)** [Test Procedure Specification 1 - Auto-Execute Selected APIVS Script(s) 6](#_Toc456255100)

**[2.2](#_Toc456255101)** [Test Procedure Specification 2 – Multiple and Concurrent Applications 9](#_Toc456255101)

**[2.3](#_Toc456255102)** [Test Procedure Specification 3 - APIVS Software Licensing Details 12](#_Toc456255102)

**[2.4](#_Toc456255103)** [Test Procedure Specification 4 - C Programming and Source Code Quality 13](#_Toc456255103)

**[2.5](#_Toc456255104)** [Test Procedure Specification 5 - XML Scripting, Execution and Logging 15](#_Toc456255104)

# INTRODUCTION

This document, *Test Procedure Specifications (TPS) for the Advanced Transportation Controller (ATC) Application Programming Interface Validation Suite (APIVS)*, provides the detailed step-by-step procedures necessary to test specific features of the API Validation Suite Engine (VSE) as identified in APIVS Test Cases.

# TEST PROCEDURE SPECIFICATIONS

The following Test Procedure Specifications are defined for use by all APIVS Test Case Specifications (TCS) found in document *Test Case Specifications for the Advanced Transportation Controller (ATC) Application Programming Interface Validation Suite (APIVS)*.

## Test Procedure Specification 1 - Auto-Execute Selected APIVS Script(s)

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIVS.TPS.1001.

### Purpose

This procedure runs the Validation Suite Engine (VSE) using the source test script and runtime options as associated with one or more specific Test Case Specifications. This execution will run from beginning to end with only limited human intervention, except as may be otherwise specified in the specific Test Case Specification.

This procedure may be used with any APIVS Test Case Specification unless otherwise indicated.

### Special Requirements

This procedure requires the editing of text files and the movement of files between a host computer Hard Disk Drive and a USB Flash Drive and is intended to be run by an operator with a reasonable technical knowledge of PC file systems and the tools available for the editing of files and the moving of files between devices.

### Procedure Steps

#### Log

All necessary logging required for the proper validation of this procedure is performed automatically by the VSE. No additional or manual logging is required.

#### Setup

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

* an ATC 5201 Standard-conforming traffic controller (VS Requirement 3.7) with a primary USB port capable of running startup scripts and a minimum 8x40 character LCD display and associated keyboard
* 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

Prior to the first execution of any test on the supplied USB Flash Drive, the runtime APIVS package must be copied into the root directory of the drive (see *APIVS User Manual*). This package contains the executable VSE program and all configuration, script and data files necessary to execute all test cases using this test procedure.

By default, all available test cases in the APIVS distribution are executed by this procedure. To select one (or more) specific test cases *only* for execution, the shell script file ***runAPIVS*** in the root of the USB Flash Drive should be edited to select the specific test cases for testing. Options in this same file also permit selection of the conformance report logging level (1-3) as well as the test iteration options. See the file comments in ***runAPIVS*** for specific instructions.

#### **Start**

To start the procedure, insert the prepared USB Flash Drive into the ATC Controller’s primary USB port and turn the controller power ON.

#### Proceed

After approximately ten (10) seconds, the LCD backlight should turn on and the display will prompt to start the test.

**ATC 5401 API Validation Suite v1.0**

**Begin Test [YES]/[NO]?**

Press the <YES> key on the keypad to start the test. The display will indicate that the testing has begun.

**ATC 5401 API Validation Suite v1.0**

**Running test session.**

**Testing APIVS.TCS.1101…**

As the test runs, the name of the current test case being tested will be indicated on the display. As the testing of each test case is completed, a PASS/FAIL status will be displayed along with a count of the total number of test cases which have passed and failed during this test session.

**ATC 5401 API Validation Suite v1.0**

**Running test session.**

**Testing APIVS.TCS.1101… PASS**

**Test cases passed:1 failed:0**

#### Measure

No measurements are necessary during the execution of this procedure.

#### Shutdown

If unexpected events occur which interrupt the execution of this procedure, turn the power to the ATC Controller to OFF and wait 30 seconds before restoring power to the controller to restart the procedure.

#### Restart

There are no available restart points for this procedure. If unexpected events occur which interrupt the execution of this procedure it must be restarted from the beginning. Turn the power to the ATC Controller to OFF and wait 30 seconds before restoring power to the controller to restart the procedure.

#### Stop

When testing is complete, the ATC Front Panel will produce three ‘beeps’ and the LCD screen will indicate completion.

**ATC 5401 API Validation Suite v1.0**

**Running test session.**

**Session complete.**

**Please remove USB drive and reboot.**

**Test cases passed:10 failed:0**

#### Wrap Up

Power down the ATC Controller, remove the USB Flash Drive from the controller and insert the drive into the PC’s USB port.

If all test cases passed (as indicated on the LCD display) then no examination of conformance reports is required and all included tests have passed.

If any failures are indicated, locate and examine the output XML file(s) (conformance report) on the USB Flash Drive to identify the cause of the failure.

#### Contingencies

None.

## Test Procedure Specification 2 – Multiple and Concurrent Applications

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIVS.TPS.1510.

### Purpose

This procedure confirms 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).

### Special Requirements

This procedure requires the retrieval of files from an online code repository (GitHub) and an examination of C-language source code files. It is intended to be run by an operator with a reasonable technical knowledge of PC file systems and Internet website access and navigation.

### Procedure Steps

#### Log

The pass/fail status of each procedure step should be logged. For procedure steps which fail, a description of the cause of the failure as well as acceptable remediation step(s) (if available) should also be noted on the log.

#### Setup

This procedure 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 keyboard
* 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

Prior to the first execution of any test on the supplied USB Flash Drive, the runtime APIVS package must be copied into the root directory of the drive (see *APIVS User Manual*). This package contains the executable VSE program and all configuration, script and data files necessary to execute all test cases using this test procedure.

The shell script file ***startup*** in the root of the USB Flash Drive should be edited to select script file ***runVS1510*** for execution.

#### Start

To start the procedure, insert the prepared USB Flash Drive into the ATC Controller’s primary USB port and turn the controller power ON.

#### Proceed

1. After approximately ten (10) seconds, the LCD backlight should turn on and the display will indicate that the Front Panel Manager is running.
2. After a short delay, the controller’s LCD should display the three test applications.

**FRONT PANEL MANAGER**

**SELECT WINDOW [0-F] SET DEFAULT \*[0-F]**

**0 ATCAPI-TEST1 1 ATCAPI-TEST2**

**2 ATCAPI-TEST3 3**

**4 5**

**6 7**

**8 9**

**[UP/DN ARROW] [CONFIG INFO- NEXT]**

**🗆** Pass

1. Let the test run for ten (10) minutes.
2. Press the <**0**> key to select the first application. The display should appear as shown. Confirm that all tests have passed.

**ATCAPI\_TEST1**

**All tests passed.**

**🗆** Pass

1. Press <**\***>-<**\***>-<**ESC**> to return to the FPM.
2. Press the <**1**> key to select the second application. The display should appear as shown. Confirm that all tests have passed.

**ATCAPI\_TEST2**

**All tests passed.**

**🗆** Pass

1. Press <**\***>-<**\***>-<**ESC**> to return to the FPM.
2. Press the <**2**> key to select the third application. The display should appear as shown. Confirm that all tests have passed.

**ATCAPI\_TEST3**

**All tests passed.**

**🗆** Pass

1. Using the following links into the source code repository, conform by code inspection that the program makes use of fpui, fio and tod libraries.

fio\_register(): https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/apps/atcapi-test.c#L60

fpui\_open(): https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/apps/atcapi-test.c#L108

tod\_get(): https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/apps/atcapi-test.c#L122

**🗆** Pass

## Test Procedure Specification 3 - APIVS Software Licensing Details

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIVS.TPS.6010.

### Purpose

This procedure confirms that the APIVS Software conforms to the licensing requirements found in the APIVS SDD.

### Special Requirements

This procedure requires the retrieval of files from an online code repository (GitHub) and an examination of C-language source code files. It is intended to be run by an operator with a reasonable technical knowledge of PC file systems and Internet website access and navigation.

### Procedure Steps

#### Log

The pass/fail status of each procedure step should be logged. For procedure steps which fail, a description of the cause of the failure as well as acceptable remediation step(s) (if available) should also be noted on the log.

#### Setup

This test procedure utilizes a subset of the hardware environment as described in the APIVS Test Plan, specifically:

* a Personal Computer (PC) with 1GB available hard drive storage and an available USB port

This test procedure 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>.

#### Start and Proceed

1. Download a copy of the APIVS software package from the online repository at <https://github.com/apiriadmin/APIVS>. See *Advanced Transportation Controller (ATC) Application Programming Interface Validation Suite (APIVS) Software User Manual* for more information on how to perform this retrieval.
2. Confirm that download did not require payment of access or licensing fees.

**🗆** Pass

1. Open a C-language source or header file from the distribution. Confirm that the licensing information in the comment header establishes the unrestricted use of the software under GPL (Gnu Public License) open-source licensing terms which have been approved by ITE.

**🗆** Pass

## Test Procedure Specification 4 - C Programming and Source Code Quality

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIVS.TPS.6020.

### Purpose

This procedure validates the APIVS requirements related to C programming and source code quality.

### Special Requirements

This procedure requires the examination of C-language source code files and an evaluation of the quality of those file(s). It is intended to be run by a software developer with expertise and experience with the style and quality of source code in the Linux kernel and GNU standard library source code, as well as compatibility requirements of the uClibc library.

### Procedure Steps

#### Log

The pass/fail status of each procedure step should be logged. For procedure steps which fail, a description of the cause of the failure as well as acceptable remediation step(s) (if available) should also be noted on the log.

#### Setup

This test procedure utilizes a subset of the hardware environment as described in the APIVS Test Plan, specifically:

* a Personal Computer (PC) with 1GB available hard drive storage and an available USB port

This test procedure requires access to the APIVS Software distribution from the online repository (GitHub) at <https://github.com/apiriadmin/APIVS>. The software can have been previously downloaded.

#### Start and Proceed

1. Open one or more source or header file(s) from the distribution. Confirm that the code is written in the C language and that the general style and quality of the code follows those of the Linux kernel and the GNU standard library.

**🗆** Pass

1. The following script ‘code’ is executed by the runAPIVS script file for every auto-executed XML script.

misc\_test\_C6020 {

#Check for uClibc load object identifier (APIVS[3.8])

LD\_TRACE\_LOADED\_OBJECTS=1 vse | grep -q "ld-uClibc"

if [ $? != 0 ]

then

echo "$(date -u):vse is not compatible with uClibc" >/tmp/C6020\_log.txt

else

echo "$(date -u):vse is compatible with uClibc" >/tmp/C6020\_log.txt

fi

mv /tmp/C6020\_log.txt ./

sync

}

This code produces an output file, C6020\_log.txt, which indicates whether or not the VSE is compatible with uClibc. Open and examine this file. Confirm that the VSE is compatible with the use of the uClibc library (PASS) or not (FAIL)..

**🗆** Pass

## Test Procedure Specification 5 - XML Scripting, Execution and Logging

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIVS.TPS.6030.

### Purpose

This procedure validates the requirements related to the APIVS 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.

### Special Requirements

This procedure requires the examination of certain test files. It is intended to be run by a software developer or technician with at least a basic understanding of Extensible Markup Language (XML) files and Linux shell and scripting language.

### Procedure Steps

#### Log

The pass/fail status of each procedure step should be logged. For procedure steps which fail, a description of the cause of the failure as well as acceptable remediation step(s) (if available) should also be noted on the log.

#### Setup

This test procedure utilizes a subset of the hardware environment as described in the APIVS Test Plan, specifically:

* a Personal Computer (PC) with 1GB available hard drive storage and an available USB port

Prior to the start of this test procedure, the following files from the runtime APIVS package must be copied onto the PC hard drive at an accessible location.

File Description

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

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

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

This test procedure 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.

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

#### Start and Proceed

1. Open the file **C1110\_in.xml** with a suitable text editor such as NotePad++. Confirm that the file contains elements of the XML language. (VS [3.10])

**🗆** Pass

1. Open the file **runAPIVS** with a suitable editor. Confirm that this script file provides the following capabilities:
2. . (VS [3.12], [3.13])

**🗆** Pass

1. that XML script files are not being compiled prior to VSE execution. (VS [3.11])

**🗆** Pass

1. that each test may be run once or continuously. (VS [3.14])

**🗆** Pass

1. that the VSE executable returns a value of 0 to indicate success or -1 to indicate failed execution of each individual test script. (VS [3.15], [3.16])

**🗆** Pass

1. Open the file **C1110\_log\_L3.xml** with a suitable text editor such as NotePad++. Confirm that the file contains elements of the XML language corresponding to a detailed conformance report. Confirm that this report corresponds to the execution of the input script file **C1110\_in.xml.** (VS [3.17])

**🗆** Pass

1. Using a suitable web browser, navigate to the web site [xmlgrid.net](http://xmlgrid.net).
2. Select the “Open File” button, followed by the “Choose File” button and select the file **C1110\_log\_L3.xml** for upload. Select “Submit”. Once the upload is complete, confirm that the file is considered ‘Well-Formed XML’ by the website.

**🗆** Pass

1. Use the website to interactively examine the structure of the XML conformance log. (VS [3.18])

**🗆** Pass

1. Open the file **C1110\_log\_L2.xml** with a suitable text editor such as NotePad++. Confirm that the file contains elements of the XML language corresponding to a less-detailed conformance report. Confirm that this report corresponds to the execution of the input script file **C1110\_in.xml.**

**🗆** Pass

1. Using a suitable web browser, navigate to the web site [xmlgrid.net](http://xmlgrid.net).
2. Select the “Open File” button, followed by the “Choose File” button and select the file **C1110\_log\_L2.xml** for upload. Select “Submit”. Once the upload is complete, confirm that the file is considered ‘Well-Formed XML’ by the website.

**🗆** Pass

1. Use the website to interactively examine the structure of the XML conformance log. (VS [3.19])

**🗆** Pass

1. Open the file **C1110\_log\_L1.xml** with a suitable text editor such as NotePad++. Confirm that the file contains elements of the XML language corresponding to a pass/failconformance report. Confirm that this report corresponds to the execution of the input script file **C1110\_in.xml.**

**🗆** Pass

1. Using a suitable web browser, navigate to the web site [xmlgrid.net](http://xmlgrid.net).
2. Select the “Open File” button, followed by the “Choose File” button and select the file **C1110\_log\_L1.xml** for upload. Select “Submit”. Once the upload is complete, confirm that the file is considered ‘Well-Formed XML’ by the website.

**🗆** Pass

1. Use the website to interactively examine the structure of the XML conformance log. (VS [3.20])

**🗆** Pass