***A Project Document of the***

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

ATC APIRI TPS v01.04

Test Procedure Specifications (TPS) for the Advanced Transportation Controller (ATC) Application Programming Interface Reference Implementation (APIRI)

**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](#_Toc456255173)

[2 TEST PROCEDURE SPECIFICATIONS 5](#_Toc456255174)

**[2.1](#_Toc456255175)** [Test Procedure Specification 1 - Auto-Execute Selected APIVS Script(s) 6](#_Toc456255175)

**[2.2](#_Toc456255176)** [Test Procedure Specification 2 - TOD Time Handling Functions 10](#_Toc456255176)

**[2.3](#_Toc456255177)** [Test Procedure Specification 3 - Front Panel Manager and ATC Configuration Menu 12](#_Toc456255177)

**[2.4](#_Toc456255178)** [Test Procedure Specification 4 - System Configuration Utilities 19](#_Toc456255178)

**[2.5](#_Toc456255179)** [Test Procedure Specification 5 - Intrinsic API Requirements 29](#_Toc456255179)

**[2.6](#_Toc456255180)** [Test Procedure Specification 6 – FIO Serial Ports and Status Counters 32](#_Toc456255180)

**[2.7](#_Toc456255181)** [Test Procedure Specification 7 – FPUI Display Presence and Size 35](#_Toc456255181)

**[2.8](#_Toc456255182)** [Test Procedure Specification 8 - FPUI Bell Activation and Application Termination 38](#_Toc456255182)

**[2.9](#_Toc456255183)** [Test Procedure Specification 9 - FPUI Display Graphics 42](#_Toc456255183)

**[2.10](#_Toc456255184)** [Test Procedure Specification 10 - FPUI Display Focus 47](#_Toc456255184)

**[2.11](#_Toc456255185)** [Test Procedure Specification 11 - System Configuration Menu Display 51](#_Toc456255185)

# INTRODUCTION

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

# TEST PROCEDURE SPECIFICATIONS

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

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

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIRI.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 APIRI 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 APIRI 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.

By default, all available test cases in the APIRI 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

1. After approximately ten (10) seconds, the LCD backlight should turn on and the display will indicate that the Validation Suite package is running. For this procedure, press the <YES> key on the keypad to select loopback mode.

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

**Loopback Mode [YES]/[NO]?**

1. Press the <YES> key on the keypad to start the test.

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

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

1. The display will indicate that the testing has begun.

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

**Running test session.**

**Testing APIRI.TCS.2010…**

1. 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 APIRI.TCS.2010… 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:24 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 - TOD Time Handling Functions

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIRI.TPS.4010.

### Purpose

This procedure validates the API’s time-handling functions by using them to asjust the controller’s clock and verifying the result.

### 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

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

All test cases executed by this procedure utilize the hardware environment as described in the APIRI 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.

#### **Start and Proceed**

1. Using test procedure APIRI.TPS.1001, execute XML script C4010\_in.xml. This procedure and XML script test the API TOD functionality using a root login. Note the result of the test.

**🗆** Pass

1. Establish a non-root login on the controller. Using a modified version of the runAPIVS script file and test procedure APIRI.TPS.1001, execute XML script C4010\_in.xml. Note the result of the test.

**🗆** Pass

#### 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

This procedure may be restarted at any point.

#### Wrap Up

Power down the ATC Controller.

If all procedure steps have passed then all included tests have passed.

#### Contingencies

None.

## Test Procedure Specification 3 - Front Panel Manager and ATC Configuration Menu

### Test Procedure Specification Identifier

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

### Purpose

This procedure validates certain API requirements for the ATC Front Panel Manager window and the ATC Configuration Menu. It is a step-by-step procedural ‘walkthrough’ of these elements of the API FPUI.

### Special Requirements

None.

### 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:

* an ATC Controller with a primary USB port capable of running startup scripts and an 8x40 character LCD display and associated keyboard

An ATC Controller with an 16x40 character LCD display and associated keyboard, as well as a primary USB port capable of running startup scripts, is also required for this procedure.

Runtime versions of the Front Panel Manager, ATC Configuration Menu and System Configuration Utilities are required to be installed and operational on the controllers.

#### **Start**

Turn the controller power ON (8x40 display version).

#### Proceed

1. After a short time, the controller’s LCD should display the Front Panel Manager (FPM) screen. If an application has been previously installed and selected as the default, the display may reflect the initial display for that application. If this is the case, press <**\***>-<**\***>-<**ESC**> to return to the FPM. The top two and bottom lines should appear exactly as shown. (APIR3.1.1.1[1])

**FRONT PANEL MANAGER**

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

**0 1**

**2 3**

**4 5**

**6 7**

**8 9**

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

**🗆** Pass

1. Press the <**DownArrow**> key three times and confirm that the display now appears as shown.

**FRONT PANEL MANAGER**

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

**6 7**

**8 9**

**A B**

**C D**

**E F**

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

**🗆** Pass

1. Individually press the <**0**>, <**ESC**>, <**YES**> and <**NO**> keys and confirm that the controller’s bell is activated for each keypress. (APIR3.1.1.1[21], APIR3.1.1.1[22], APIR3.2.1[38])

**🗆** Pass

1. Press <**\***>-<**\***>-<**NEXT**> and confirm that the ATC Configuration Menu is displayed. The menu items shown are the minimum allowed – additional items are acceptable. The top two and bottom lines should appear exactly as shown. (APIR3.1.1[13], APIR3.1.1.1[16], APIR3.1.1.1[17], APIR3.2.1[1], APIR3.2.1[2], APIR3.2.1[3], APIR3.2.1[7], APIR3.2.1[8], APIR3.2.1[9], APIR3.2.1[11], APIR3.2.1[12], APIR3.2.1[14], APIR3.2.1[16])

**ATC CONFIGURATION INFORMATION**

**SELECT ITEM [0-F]**

**0 System Time 1 Ethernet Port 1**

**2 Ethernet Port 2 3 System Services**

**4 Linux Info 5 API Info**

**6 Host EEPROM 7 Termination Test**

**8 9**

**[UP/DN ARROW] [FRONT PANEL- NEXT]**

**🗆** Pass

1. Confirm that the test display of the ATC Configuration Menu is normal text (not inverted or blinking) and that the backlight is off. (APIR3.2.1[13])

**🗆** Pass

1. Individually press the <**ESC**>, <**YES**> and <**NO**> keys and confirm that the controller’s bell is activated for each keypress. (APIR3.2.1[20])

**🗆** Pass

1. Press <**NEXT**> and confirm that the Front Panel Manager is displayed. Press <**NEXT**> again and confirm that the ATC Configuration Menu is displayed. (APIR3.2.1[19])

**🗆** Pass

1. Press the <**DownArrow**> key three times and confirm that the display now appears as shown. (APIR3.2.1[18])

**ATC CONFIGURATION INFORMATION**

**SELECT ITEM [0-F]**

**6 Host EEPROM 7**

**8 9**

**A B**

**C D**

**E F**

**[UP/DN ARROW] [FRONT PANEL- NEXT]**

**🗆** Pass

1. Press the <**UpArrow**> key three times. Press the <**0**> key to select the System Time Configuration Utility. The display should appear as shown. (APIR3.2.1[4], APIR3.2.1[5], APIR3.2.1[6], APIR3.2.1[15])

**SYSTEM TIME**

**DATE TIME TMZONE DST/Status**

**01/20/2016 17:26:08 -05:00 Enable/Inactv**

**CHANGE**

**DATE TIME TMZONE DST**

**01/20/2016 17:20:00 -05:00 Enable**

**[UP/DN ARROW] [APPLY-ENT] [QUIT-\*\*NEXT]**

**🗆** Pass

1. Using the <**LeftArrow**> and <**RightArrow**> keys, confirm that the cursor moves between the various time fields in the ‘Change’ section. Confirm that pressing the <**Left Arrow**> key when on the leftmost field (Month) has no effect. Confirm that pressing the <**Right Arrow**> key when on the rightmost field (DST Enable) has no effect. (APIR3.2.1[22], APIR3.2.1[23], APIR3.2.1[24], APIR3.2.1[25], APIR3.2.1[26])

**🗆** Pass

1. Move the cursor to the change Minute field. Set the Minute to 10 by pressing the <1>, <0> and <Ent> keys. Using the <+> and <-> keys, demonstrate that the value is incremented and decremented by those keys (respectively). Advance the minute field one minute ahead of the current minute and press the <**ENT**> key. Confirm that the utility remains in focus. Confirm that the minute in the current time has now advanced as well. Repeat this step five times, except instead of pressing the <**ENT**> key press the <**YES**>, <**UpArrow**>, <**DownArrow**>, <**LeftArrow**> and <**RightArrow**> keys to terminate the field. Confirm in each case that the minute in the current time has advanced, indicating that the new value was accepted. Repeat this step again, except instead of pressing the <**ENT**> key press the <**NO**> key. Confirm that the minute in the current time has reverted to it’s previous value. (APIR3.2.1[21], APIR3.2.1[27], APIR3.2.1[28], APIR3.2.1[29], APIR3.2.1[30], APIR3.2.1[31], APIR3.2.1[35])

**🗆** Pass

1. Using the numeric keypad <**0**>-<**9**>, enter a value into the change Minute field. Confirm that the value entered is right-justified within the field and that if the number of digits entered exceeds the width of the field that the excess leftmost digits are removed from the field. (APIR3.2.1[32], APIR3.2.1[33], APIR3.2.1[34])

**🗆** Pass

1. Press the <**NO**> key and confirm that the field value returns to what it was prior to pushing the first numeric key. (APIR3.2.1[31], APIR3.2.1[36])

**🗆** Pass

1. Press any key <**A**>-<**F**> and confirm that the controller’s bell sounds with each key press (as these are invalid keys for time entry fields). (APIR3.2.1[37])

**🗆** Pass

1. Using the numeric keypad <**0**>-<**9**>, enter a value into the change Minute field that is five minutes in advance of the current minute. Do NOT press any other keys.

**🗆** Pass

1. Press <**\***>-<**\***>-<**NEXT**> and confirm that the ATC Configuration Menu is displayed. (APIR3.2.1[39])

**🗆** Pass

1. Press the <**0**> key to (again) select the System Time Configuration Utility. Confirm that the last change made to the Minute field (which was not confirmed with the <ENT> key) was NOT accepted and that the current time is unchanged. (APIR3.2.1[40])
2. APIR3.2.1[44])

**🗆** Pass

1. Press <**\***>-<**\***>-<**NEXT**> and confirm that the ATC Configuration Menu is displayed.

**🗆** Pass

1. Press the <**7**> key to select the Termination Test Utility. Press <\*>-<\*>-<NEXT>, wait two seconds, and confirm that the display now appears as shown. (APIR3.2.1[41])

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**The Configuration Utility**

**Termination Test has not closed.**

**Do you wish to terminate it?**

**Press Yes or No**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**🗆** Pass

1. Press <**NO**> and confirm that the Termination Test utility is redisplayed. (APIR3.2.1[43])

**🗆** Pass

1. Press <\*>-<\*>-<NEXT>, wait two seconds, and confirm that the termination display now appears as before.

**🗆** Pass

1. Press <**YES**> and confirm that the ATC Configuration Menu is displayed. (APIR3.2.1[42])

**🗆** Pass

1. Repeat test steps 1-7 using the controller with the 16x40 display. Confirm that for steps 1, 4 and 7 that the first two and last text lines of the display are identical and are in the correct locations. (APIR3.1.1.1[18], APIR3.2.1[17])

**🗆** Pass

#### 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

This procedure may be restarted at any point.

#### Wrap Up

Power down the controller.

## Test Procedure Specification 4 - System Configuration Utilities

### Test Procedure Specification Identifier

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

### Purpose

This procedure validates certain API requirements for the System Configuration Utilities. It is a step-by-step procedural ‘walkthrough’ of these elements of the API FPUI.

### Special Requirements

This procedure requires the retrieval of certain Linux system information from an ATC Controller. It is intended to be run by a software developer or technician with at least a reasonable understanding of the Linux operating system and the method(s) for retrieving specific configuration information. An understanding of how to connect to the console on the ATC controller used for testing is also required.

### 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:

* an ATC Controller with a primary USB port capable of running startup scripts and a minimum 8x40 character LCD display and associated keyboard. The controller should also have a functional console port available (either via serial port or SSH Ethernet connection)
* a Personal Computer (PC) with 1GB available hard drive storage and an available USB port

An ATC Controller with an 16x40 character LCD display and associated keyboard, as well as a primary USB port capable of running startup scripts, is also required for this procedure.

Runtime versions of the Front Panel Manager, ATC Configuration Menu and System Configuration Utilities are required to be installed and operational on the controllers.

#### **Start**

Turn the controller power ON.

Using a serial or network (SSH) connection, make a console connection to the controller and retrieve the following information (if it is not already available):

* Current time (approximate)
* All network settings for both Ethernet ports
* A list of current system services
* Version numbers and memory details for installed Linux system
* Version numbers for API libraries

#### Proceed

1. After a short delay, the controller’s LCD should display the Front Panel Manager (FPM) screen. If an application has been previously installed and selected as the default, the display may reflect the initial display for that application. If this is the case, press <**\***>-<**\***>-<**ESC**> to return to the FPM. The top two and bottom lines should appear exactly as shown.

**FRONT PANEL MANAGER**

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

**0 1**

**2 3**

**4 5**

**6 7**

**8 9**

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

**🗆** Pass

1. Press <**NEXT**> and confirm that the ATC Configuration Menu is displayed. The top two and bottom lines should appear exactly as shown.

**ATC CONFIGURATION INFORMATION**

**SELECT ITEM [0-F]**

**0 System Time 1 Ethernet Port 1**

**2 Ethernet Port 2 3 System Services**

**4 Linux Info 5 API Info**

**6 Host EEPROM 7**

**8 9**

**[UP/DN ARROW] [FRONT PANEL- NEXT]**

**🗆** Pass

1. Press the <**0**> key to select the System Time Configuration Utility. The display should generally appear as shown. The top two and bottom lines should appear exactly as shown. (APIR3.2.2[1], APIR3.2.2[2], APIR3.2.2[3], APIR3.2.2[4], APIR3.2.2[8], APIR3.2.2[9])

**SYSTEM TIME**

**DATE TIME TMZONE DST/Status**

**01/20/2016 17:26:08 -05:00 Enable/Inactv**

**CHANGE**

**DATE TIME TMZONE DST**

**01/20/2016 17:20:00 -05:00 Enable**

**[UP/DN ARROW] [APPLY-ENT] [QUIT-\*\*NEXT]**

**🗆** Pass

1. Confirm that the system time displayed is correct and the the fields in the ‘Change’ section match the system time at the moment that the utility is first displayed. (APIR3.2.2[7])

**🗆** Pass

1. Confirm that all fields in the ‘Change’ section can be modified and that pressing <**ENT**> after all fields have been altered adjusts the system time to match the values entered. (APIR3.2.2[6])

**🗆** Pass

1. Press <**\***>-<**\***>-<**NEXT**> and confirm that the ATC Configuration Menu is displayed.

**🗆** Pass

1. Press the <**1**> key to select the Ethernet Configuration Utility (Port 1). The display should appear generally as shown. The exact values for the various settings will be different - this does not constitute a failure, however, the top and bottom lines should appear exactly as shown. (APIR3.2.3[1], APIR3.2.3[2], APIR3.2.3[3], APIR3.2.3[4])

**ETHERNET CONFIGURATION**

**ETHERNET PORT 1 (64:55:63:00:05:55)**

**Port Mode: static**

**IP Address: 169.254. 1.100**

**Subnet Mask: 255.255. 0. 0**

**Default Gateway: 0. 0. 0. 0**

**DNS Server: 127. 0. 0. 1**

**[UP/DN ARROW] [APPLY-ENT] [QUIT-\*\*NEXT]**

**🗆** Pass

1. Confirm that the display matches the controller’s network settings for Ethernet Port 1. (APIR3.2.3[10])

**🗆** Pass

1. Using the <**Arrow**> keys and the numeric keypad <**0**>-<**9**>, confirm that it is possible to modify one or more of the configuration settings. Press the <**ENT**> key to apply the changes. Note the specific settings that were changed and the new settings used. (APIR3.2.3[5], APIR3.2.3[6])

**🗆** Pass

1. Press the <**DownArrow**> key and confirm that the display now appears as shown. Again, the exact values for the various settings will be different - this does not constitute a failure, however, the top and bottom lines should appear exactly as shown. (APIR3.2.3[8],)

**ETHERNET CONFIGURATION**

**ETHERNET PORT 1 (64:55:63:00:05:55)**

**Host Name: ATC1**

**Packets Sent GD: BD:**

**Packets Rcvd GD: BD:**

**[UP/DN ARROW] [APPLY-ENT] [QUIT-\*\*NEXT]**

**🗆** Pass

1. Confirm that the numofpackets fields change once per second if network traffic is occurring. Confirm that the numofpackets fields cannot be modified. (APIR3.2.3[7], APIR3.2.3[8], APIR3.2.3[9])

**🗆** Pass

1. Repeat Steps 6 through 10, except that at Step 7 the <**2**> key should be pressed to select the Ethernet Configuration Utility (Port 2). Confirm that the display matches the controller’s network settings for Ethernet Port 2.

**🗆** Pass

1. Press <**\***>-<**\***>-<**NEXT**> and confirm that the ATC Configuration Menu is displayed.

**🗆** Pass

1. Press the <**3**> key to select the System Services Utility. The display should appear generally as shown. The exact values for the various settings will be unique - this does not constitute a failure, however, the top two and bottom lines should appear exactly as shown. (APIR3.2.4[1], APIR3.2.4[2], APIR3.2.4[3])

**ENABLE/DISABLE SYSTEM SERVICES**

**SERVICE STATUS CHANGE**

***ServiceName0 ENABLED ENABLED***

***ServiceName1 ENABLED ENABLED ServiceName2 ENABLED ENABLED ServiceName3 ENABLED ENABLED***

***ServiceName4 ENABLED ENABLED* [UP/DN ARROW] [APPLY–ENT] [QUIT-\*\*NEXT]**

**🗆** Pass

1. Confirm that the display can be scrolled using the < **UpArrow** > and <**DownArrow**> keys. Confirm that the service names are limited to 22 characters. (APIR3.2.4[5], APIR3.2.4[7])

**🗆** Pass

1. Confirm that the services displayed match the controller’s Linux configuration, including their condition (Enabled/Disabled). Confirm that the ‘STATUS’ field can not be modified. (APIR3.2.4[6], APIR3.2.4[8], APIR3.2.4[9], APIR3.2.4[10])

**🗆** Pass

1. Confirm that the ‘CHANGE’ field can be modified to be either “Enabled” or “Disabled”. (APIR3.2.4[11], APIR3.2.4[12])

**🗆** Pass

1. Press <**\***>-<**\***>-<**NEXT**> and confirm that the ATC Configuration Menu is displayed.

**🗆** Pass

1. Press the <**4**> key to select the Linux Information Utility. The display should appear generally as shown. The exact values for the various settings will be unique - this does not constitute a failure, however, the top and bottom lines should appear exactly as shown. (APIR3.2.5[1], APIR3.2.5[2], APIR3.2.5[3])

**LINUX INFORMATION**

**Linux Release: *release***

**Linux Version: *version***

**Machine Hardware Type*: type***

**Memory Total: *####MB* Free: *####MB***

**Filesystem Total: *####MB* Free: *####MB***

**Load Average: *loadavg***

**[UP/DN ARROW] [QUIT-\*\*NEXT]**

**🗆** Pass

1. Press the <**DownArrow**> key and confirm that the display now appears as shown. Confirm that the display can be scrolled using the < **UpArrow** > and <**DownArrow**> keys. (APIR3.2.5[4])

**LINUX INFORMATION**

**Uptime: *#### days ## hours ## mins***

**[UP/DN ARROW] [QUIT-\*\*NEXT]**

**🗆** Pass

1. Confirm that the values displayed match the controller’s Linux configuration. (APIR3.2.5[5], APIR3.2.5[6])

**🗆** Pass

1. Press <**\***>-<**\***>-<**NEXT**> and confirm that the ATC Configuration Menu is displayed.

**🗆** Pass

1. Press the <**5**> key to select the API Information Utility. The display should appear generally as shown. The top and bottom lines should appear exactly as shown. (APIR3.2.5[8], APIR3.2.5[9], APIR3.2.5[10])

**API INFORMATION**

**FIO API Version: *version\_text***

**FIO API LKM Version: *version\_text***

**FPUI API Version: *version\_text***

**FPUI API LKM Version: *version\_text***

**TOD API Version: *version\_text***

**[UP/DN ARROW] [QUIT-\*\*NEXT]**

**🗆** Pass

1. Confirm that the versions displayed match the controller’s API configuration. (APIR3.2.5[7])

**🗆** Pass

1. Press <**\***>-<**\***>-<**NEXT**> and confirm that the ATC Configuration Menu is displayed.

**🗆** Pass

1. Press the <**6**> key to select the Host EEPROM Information Utility. The display should appear generally as shown. Confirm that pressing the <**UpArrow**> and <**DownArrow**> keys allows scrolling within the display of all of the EEPROM information. The top and bottom lines should appear exactly as shown. (APIR3.2.6[1], APIR3.2.6[2], APIR3.2.6[3], APIR3.2.6[5], APIR3.2.6[7])

**HOST EEPROM INFORMATION**

**Host EEPROM Version: *2***

**Host EEPROM Size (bytes): *150***

**#Modules: *4***

**<module\_information>**

**<module\_information>**

**<module\_information>**

**[UP/DN ARROW] [QUIT-\*\*NEXT]**

**🗆** Pass

1. Confirm that the values for the "Ethernet Switch/Router Mac Addresses," "Host Board Serial Ports Used" and the "Agency Reserved" fields are displayed as hexadecimal pairs. (APIR3.2.6[6])

**🗆** Pass

1. Press <**\***>-<**\***>-<**NEXT**> and confirm that the ATC Configuration Menu is displayed.

**🗆** Pass

1. Using a console connection to the controller, confirm that the changes to the network settings for Ethernet Ports 1 and 2 which were made in Steps 10 and 11 were successfully applied

**🗆** Pass

1. Repeat all test steps using the controller with the 16x40 display. Confirm that for steps x, x, x, x, x and x that the first and last text lines of the display are identical and are in the correct locations. (APIR3.2.2[4], APIR3.2.3[4], APIR3.2.4[4], APIR3.2.5[3], APIR3.2.6[4])

**🗆** Pass

#### 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

This procedure may be restarted at any point.

#### Wrap Up

Power down the controller.

## Test Procedure Specification 5 - Intrinsic API Requirements

### Test Procedure Specification Identifier

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

### Purpose

This procedure validates a number of requirements which must be confirmed by an inspection of the relevant source code files.

### Special Requirements

This procedure requires the examination of certain API distribution C-language source code files. It is intended to be run by a software developer with specific knowledge of the API distribution source code files and structure.

### 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:

* an ATC Controller with a primary USB port capable of running startup scripts and a minimum 8x40 character LCD display and associated keyboard. The controller should also have a functional console port available (either via serial port or SSH Ethernet connection)
* 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.

It is expected that the files necessary for each procedural step inspection during formal testing will have been previously identified and the specific locations within those files to be inspected will also have been noted.

#### **Start and Proceed**

1. Confirm by inspection that the ATC Controller satisfies the hardware requirements as defined in the ATC Standard. (APIR3.4[1]).

**🗆** Pass

1. Inspect every ATC/API vendor’s controller products to verify that the operational look and feel of user interfaces developed for the API have consistent window titling conventions, scrolling methods, menu styles and selection methods.. (APIR3.5.2[1]).

**🗆** Pass

1. Select a C source code file at random from the APIRI distribution. Confirm by inspection that it meets the following requirements:
2. API function calls are specified using the C programming language as described by “ISO/IEC 9899:1999” commonly referred to as the C99 Standard. (APIR3.4[2]).

**🗆** Pass

1. If API functions have a similar operation to existing Linux functions, they shall have a similar name and argument style to those functions to the extent possible without causing compilation issues.. (APIR3.5.2[2]).

**🗆** Pass

1. API function calls are lower case. (APIR3.5.2[3]).

**🗆** Pass

1. API functions shall use the Linux “errno” error notification mechanism if an error indication is expected for a function.. (APIR3.5.2[4]).

**🗆** Pass

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

misc\_test\_C6030 {

#Check for ELF format (APIR3.5.2[5])

elfmagic="7f454C46"

FILES="/usr/lib/libfpui.so

/usr/lib/libfio.so

/usr/lib/libtod.so"

for f in $FILES

do

if [ $(od -An -N4 -tx4 <$f) == $elfmagic ]; then

echo "$(date -u):$f is ELF format file" >>/tmp/C6030\_log.txt

else

echo "$(date -u):$f is not ELF format file" >>/tmp/C6030\_log.txt

fi

done

mv /tmp/C6030\_log.txt ./

sync

}

This code produces an output file, C6030\_log.txt, which indicates whether or not the API libraries are loadable as an ELF library. Open and examine this file. Confirm that the API libraries are loadable as an ELF library (PASS) or not (FAIL). (APIR3.5.2[5])

**🗆** Pass

1. Inspect every single APIRI source code file in it’s entirety and verify that the API software only references operating system commands and features that are available in the Linux environment defined in the ATC Board Support Package (see ATC Controller Standard, Section 2.2.5, Annex A and Annex B).. (APIR3.6[1]).

**🗆** Pass

## Test Procedure Specification 6 – FIO Serial Ports and Status Counters

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIRI.TPS.6040.

### Purpose

This procedure validates a number of software-related requirements which do not have ‘exposure’ outside the internal workings of the API and so must be confirmed by an inspection of the relevant source code files.

### Special Requirements

This procedure requires the examination of certain API distribution C-language source code files. It is intended to be run by a software developer with specific knowledge of the API distribution source code files and structure.

### 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 the source file fiomsg.c using the following link:

<https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.c#L1010-L1056>

Confirm by inspection that the following requirements are satisfied:

* 1. the API expects to have exclusive access to the serial ports used for FIO devices (APIR3.1.2[1])
  2. the supported FIO communications ports are SP3, SP5 and SP8 (APIR3.1.2[2])
  3. the only supported communications modes on those ports is SDLC at either 153.6bkps or 614.4kbps (APIR3.1.2[3])
  4. that BIU and MMU FIO devices only operate at 153.6bkps and that all other devices operate at 614.4kbps (APIR3.1.2[7])

**🗆** Pass

1. Open the source files fiomsg.h, fioapi.h, fiomsg.c, and fioman.c using the following links:

[https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.h#L225](https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.h" \l "L225)

[https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fioapi/fioapi.h#L252-L253](https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fioapi/fioapi.h" \l "L252-L253)

[https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fioman.c#L2920](https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fioman.c" \l "L2920)

[https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.c#L747-L748](https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.c" \l "L747-L748)

[https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.c#L764-L765](https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.c" \l "L764-L765)

Confirm by inspection that the following requirements are satisfied:

* 1. All counters contained in the Field I/O Device status information shall be four byte unsigned values each with a maximum value of 4,294,967,295 (APIR3.1.2[103])
  2. The counters shall be frozen when they reach the maximum value to prevent rollover (APIR3.1.2[104[)

**🗆** Pass

1. Open the source files fiomsg.h, fioapi.h, and fiomsg.c using the following links:

[https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.h#L225](https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.h" \l "L225)

[https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fioapi/fioapi.h#L255](https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fioapi/fioapi.h" \l "L255)

[https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.c#L762-L763](https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.c" \l "L762-L763)

Confirm by inspection that the following requirements are satisfied:

* 1. The response frame sequence number shall be a four byte unsigned value and rollover after the maximum value (APIR3.1.2[106[)

**🗆** Pass

1. Open the source file fioman.h using the following link:

<https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fioman.h#L101-L104>

Confirm by inspection that the following requirements are satisfied:

* 1. To set the state of an output point and control dimming, the API shall use separate arrays for control of the Load Switch + and Load Switch – (see Section 3.3.1.4.1.5 of the TS 2 Standard). (APIR3.1.2[42[)

**🗆** Pass

1. Open the source file fiomsg.c using the following link:

<https://github.com/apiriadmin/APIRI/blob/2e5416b2972451858eb53b2196d7d67779a5a997/fio/src/fiodriver/fiomsg.c#L199-L340>

Confirm by inspection that the following requirements are satisfied:

* 1. The timing for the command/response cycle of the frames shall be defined by the “Handshaking” algorithm in Section 3.3.1.5.3 of the NEMA TS 2 Standard. (APIR3.1.2[73[)

**🗆** Pass

## Test Procedure Specification 7 – FPUI Display Presence and Size

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIRI.TPS.7010.

### Purpose

This procedure validates certain API requirements related to the window display size and the detection of the presense (or absence) of a connected display panel. As suitable controller for performing panel removal and size changes is not available, these requirements must be confirmed by an inspection of the relevant source code files.

### Special Requirements

This procedure requires the examination of certain API distribution C-language source code files. It is intended to be run by a software developer with specific knowledge of the API distribution source code files and structure.

### 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 the source file ViewportManager.c using the following link:

<https://github.com/apiriadmin/APIRI/blob/117fc0f14e6dc5871f1a4b090d6c571ab9c2622a/fpu/FrontPanelSystem/FrontPanelManager/ViewportManager.c#L151>

Confirm by inspection that the following requirements are satisfied:

* 1. the display size of the windows shall have a minimum size of 4 lines x 40 characters and a maximum size of 24 lines x 80 characters. (APIR3.1.1[4])

**🗆** Pass

1. Open the source file ViewportManager.c using the following link:

<https://github.com/apiriadmin/APIRI/blob/117fc0f14e6dc5871f1a4b090d6c571ab9c2622a/fpu/FrontPanelSystem/FrontPanelManager/ViewportManager.c#L143-L145>

Confirm by inspection that the following requirements are satisfied:

* 1. If no physical display exists, the API shall operate as if it has a display with a size of 8 lines x 40 characters. (APIR3.1.1[5])

**🗆** Pass

1. Open the source file libfpui.c using the following link:

<https://github.com/apiriadmin/APIRI/blob/117fc0f14e6dc5871f1a4b090d6c571ab9c2622a/fpu/FrontPanelSystem/libfpui/libfpui.c#L113-L133>

Confirm by inspection that the following requirements are satisfied:

* 1. the API shall provide a mechanism to allow application programs to detect the presence or absence of a Front Panel. (APIR3.1.1.2[38])

**🗆** Pass

1. Open the source file ViewportManager.c using the following link:

<https://github.com/apiriadmin/APIRI/blob/117fc0f14e6dc5871f1a4b090d6c571ab9c2622a/fpu/FrontPanelSystem/FrontPanelManager/ViewportManager.c#L402-L410>

Confirm by inspection that the following requirements are satisfied:

* 1. the API shall recognize the presence or absence of a Front Panel in 5 seconds. (APIR3.1.1.2[39])

**🗆** Pass

1. Open the source file ViewportManager.c using the following link:

<https://github.com/apiriadmin/APIRI/blob/117fc0f14e6dc5871f1a4b090d6c571ab9c2622a/fpu/FrontPanelSystem/FrontPanelManager/ViewportManager.c#L402-L415>

Confirm by inspection that the following requirements are satisfied:

1. the API shall provide an asynchronous notification to alert application programs of a change in the presence or absence of a Front Panel. (APIR3.1.1.2[40])

**🗆** Pass

1. Open the source file ViewportManager.c using the following link:

<https://github.com/apiriadmin/APIRI/blob/117fc0f14e6dc5871f1a4b090d6c571ab9c2622a/fpu/FrontPanelSystem/FrontPanelManager/ViewportManager.c#L459-L466>

Confirm by inspection that the following requirements are satisfied:

1. the API shall provide an asynchronous notification to alert all application programs when their associated windows change size. (APIR3.1.1.2[41])

**🗆** Pass

## Test Procedure Specification 8 - FPUI Bell Activation and Application Termination

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIRI.TPS.7020.

### Purpose

This procedure tests the controller’s bell activation requirements. It runs the Validation Suite Engine (VSE) using the associated Test Case Specification’s XML test script to create two FPUI ‘applications’ which aid in conducting the test. One application will ring the controller’s bell periodically while the other will not. Only when the bell-ringing application is in focus should the bell be heard.

This procedure also verifies that an FPUI API connection is closed when an application terminates without explicitly closing the connection.

### Special Requirements

This script runs in non-loopback mode so that the applications have access to the controller’s physical front panel.

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

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

All test cases executed by this procedure utilize the hardware environment as described in the APIRI 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 ***runVSnlb*** in the root of the USB Flash Drive should be edited to select this procedure’s specific test case (C7020) alone for testing. See the file comments in ***runVSnlb*** 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

1. After approximately ten (10) seconds, the LCD backlight should turn on and the display will indicate that the Validation Suite package is running. For this procedure, press the <NO> key on the keypad to select non-loopback mode.

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

**Loopback Mode [YES]/[NO]?**

1. Press the <YES> key on the keypad to start the test.

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

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

1. After a brief delay the display should show the Front Panel Manager (FPM). After another brief delay, the names of the two FPUI applications should appear in the FPM window.

**FRONT PANEL MANAGER**

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

**0 C7020\_00 1 C7020\_01**

**2 3**

**4 5**

**6 7**

**8 9**

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

**🗆** Pass

1. Press the <**0**> key to select the first application. The display should appear as shown. Confirm that the controller’s bell is silent. APIR3.1.1.2[25] (The application that is ringing the bell, C7020\_01, does NOT have focus here, so the bell is silent).

**C7020\_00**

**Bell is OFF**

**🗆** 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 the controller’s bell is active approximately every two seconds. APIR3.1.1.2[24]

**C7020\_01**

**Bell is ON**

**🗆** Pass

1. Press <**\***>-<**\***>-<**ESC**> to return to the FPM.
2. Approximately 120 seconds after the test begins, both applications will abnormally terminate (without explicitly closing their connection to the FPUI API). If they are successfully deregistered from the API, their registered names will be removed from the FPM display. Confirm that this is the case. APIR3.1.2[134]

**FRONT PANEL MANAGER**

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

**0 1**

**2 3**

**4 5**

**6 7**

**8 9**

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

**🗆** Pass

#### 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.

#### Wrap Up

Power down the ATC Controller and remove the USB Flash Drive from the controller.

If all procedure steps have passed then no examination of the conformance report is required and all included tests have passed.

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

#### Contingencies

None.

## Test Procedure Specification 9 - FPUI Display Graphics

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIRI.TPS.7030.

### Purpose

This procedure tests the API’s handling of graphical display commands. It runs the Validation Suite Engine (VSE) using the associated Test Case Specification’s XML test script to create two FPUI ‘applications’ which aid in conducting the test. Each application will update a small graphical area on the display a short time after receiving display focus.

### Special Requirements

This script runs in non-loopback mode so that the applications have access to the controller’s physical front panel.

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

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

All test cases executed by this procedure utilize the hardware environment as described in the APIRI 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 ***runVSnlb*** in the root of the USB Flash Drive should be edited to select this procedure’s specific test case (C7030) alone for testing. See the file comments in ***runVSnlb*** 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

1. After approximately ten (10) seconds, the LCD backlight should turn on and the display will indicate that the Validation Suite package is running. For this procedure, press the <NO> key on the keypad to select non-loopback mode.

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

**Loopback Mode [YES]/[NO]?**

1. Press the <YES> key on the keypad to start the test.

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

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

1. After a brief delay the display should show the Front Panel Manager (FPM). After another brief delay, the names of the two FPUI applications should appear in the FPM window.

**FRONT PANEL MANAGER**

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

**0 C7030\_00 1 C7030\_01**

**2 3**

**4 5**

**6 7**

**8 9**

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

**🗆** Pass

1. Press the <**0**> key to select the first application. The display should appear as shown.

**C7030\_00**

**🗆** Pass

1. After approximately five seconds, a graphical icon should appear on the display.

**C7030\_00**

**0**

**🗆** 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.

**C7030\_01**

**🗆** Pass

1. After approximately five seconds, a graphical icon should appear on the display.

**C7030\_01**

**1**

**🗆** Pass

1. Press <**\***>-<**\***>-<**ESC**> to return to the FPM.
2. Press the <**0**> key to select the first application again. The display should appear as shown.

**C7030\_00**

**🗆** Pass

1. After approximately five seconds, a graphical icon should appear on the display.

**C7030\_00**

**0**

**🗆** Pass

#### 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.

#### Wrap Up

Power down the ATC Controller and remove the USB Flash Drive from the controller.

If all procedure steps have passed then no examination of the conformance report is required and all included tests have passed.

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

#### Contingencies

None.

## Test Procedure Specification 10 - FPUI Display Focus

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIRI.TPS.7040.

### Purpose

This procedure tests the API requirements regarding an application’s request for display focus. It runs the Validation Suite Engine (VSE) using the associated Test Case Specification’s XML test script to create three FPUI ‘applications’ which aid in conducting the test. Two applications which are not originally in focus will request it and the user will be required to observe the ‘requests’ and to give focus to those applications and validate the requirements.

### Special Requirements

This script runs in non-loopback mode so that the applications have access to the controller’s physical front panel.

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

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

All test cases executed by this procedure utilize the hardware environment as described in the APIRI 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 ***runVSnlb*** in the root of the USB Flash Drive should be edited to select this procedure’s specific test case (C7040) alone for testing. See the file comments in ***runVSnlb*** 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

1. After approximately ten (10) seconds, the LCD backlight should turn on and the display will indicate that the Validation Suite package is running. For this procedure, press the <NO> key on the keypad to select non-loopback mode.

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

**Loopback Mode [YES]/[NO]?**

1. Press the <YES> key on the keypad to start the test.

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

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

1. After a brief delay the display should show the Front Panel Manager (FPM). After another brief delay, the names of the three FPUI applications should appear in the FPM window.

**FRONT PANEL MANAGER**

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

**0 C7040\_00 1 C7040\_01**

**2 C7040\_02 3**

**4 5**

**6 7**

**8 9**

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

**🗆** Pass

1. After a brief delay, confirm that the display backlight is flashing and that the names of applications C7040\_01 and C7040\_02 in the FPM window are blinking.

**🗆** Pass

1. Press the <**0**> key to select the first application. The display should appear as shown.

**C7040\_00**

**🗆** Pass

1. Press <**\***>-<**\***>-<**ESC**> to return to the FPM.
2. Confirm that the display backlight is still flashing and that the names of applications C7040\_01 and C7040\_02 in the FPM window are still blinking.

**🗆** Pass

1. Press the <**1**> key to select the second application. The display should appear as shown.

**C7040\_01**

**🗆** Pass

1. Press <**\***>-<**\***>-<**ESC**> to return to the FPM.
2. Confirm that the display backlight is still flashing and that the name of application C7040\_02 in the FPM window is still blinking. C7040\_01 should no longer be blinking.

**🗆** Pass

1. Press the <**2**> key to select the third application. The display should appear as shown.

**C7040\_02**

**🗆** Pass

1. Press <**\***>-<**\***>-<**ESC**> to return to the FPM.
2. Confirm that the display backlight is no longer flashing and that none of the application names in the FPM window are blinking.

**🗆** Pass

#### 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.

#### Wrap Up

Power down the ATC Controller and remove the USB Flash Drive from the controller.

If all procedure steps have passed then no examination of the conformance report is required and all included tests have passed.

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

#### Contingencies

None.

## Test Procedure Specification 11 - System Configuration Menu Display

### Test Procedure Specification Identifier

The identifier for this Test Procedure Specification is APIRI.TPS.7050.

### Purpose

This procedure validates the use of the configuration text file used by the System Configuration Utility to create the System Configuration Menu.

### Special Requirements

This procedure requires an understanding of the method within the APIVS software for how the changes to the configuration file are applied to the configuration menu. It is intended to be run by a software developer who has experience with the APIVS Software distribution.

### 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

All tests executed by this procedure utilize the hardware environment as described in the APIRI 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.

#### Start and Proceed

1. Locate and open the file **ATCConfigurationMenu.txt** with a suitable text editor such as NotePad++. Confirm that the file contains configuration line items of the form:

configitemname, executablepathname

**🗆** Pass

1. Edit the file to add 12 new configuration items of the following form.

SCTestApp1, SCTestApp

SCTestApp2, SCTestApp

SCTestApp3, SCTestApp

SCTestApp4, SCTestApp

SCTestApp5, SCTestApp

SCTestApp6, SCTestApp

SCTestApp7, SCTestApp

SCTestApp8, SCTestApp

SCTestApp9, SCTestApp

SCTestApp10, SCTestApp

SCTestApp11, SCTestApp

SCTestApp12, SCTestApp

**🗆** Pass

1. Locate and open the file **SConfig.c** with a suitable text editor such as NotePad++. Confirm by code inspection that the file contains API function calls fpui\_open\_configuration\_window() and fpui\_close\_configuration\_window() for opening and closing (respectively) the ATC Configuration Window:

**🗆** Pass

1. Restart the controller. Navigate to the System Configuration Menu and confirm that the 12 new test applications added to the configuration file are present in the menu:

**🗆** Pass