A Project Document of the
ATC Application Programming Interface Working Group

ATC APIVS SDD WTWB v01.03

Application Programming Interface (API) Validation Suite Software Design Description (SDD) v01.01 Walkthrough Workbook (WTWB)

May 22, 2015

WTWB in support of:	USDOT Contract # DTFH61-11-D-00052, Task Order # T-13-003
For use by:	Members of the ATC API Working Group
	Consulting Team for the ATC API Reference Implementation Project
Prepared by:	Ralph W. Boaz, Pillar Consulting, Inc.

CHANGE HISTORY

DATE	NOTE
01/02/15	Initial Draft v01.00
01/05/15	Draft v01.01. Updated per first day of walkthrough.
02/02/15	Draft v01.02. Developer draft.
05/22/15	Draft v01.03. Updated with responses to issues from the API WG walkthroughs.

NOTICE

Joint NEMA, AASHTO and ITE Copyright and
Advanced Transportation Controller (ATC)
Application Programming Interface (API) 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

1.1 Purpose 5
1.2 Scope 5

2 REFERENCES 5

3 DESIGN ANALYSIS 6

4 GENERAL NOTES AND ACTIONS FROM THE WALKTHROUGH 6

5 WALKTHROUGH MATRIX 7

1 INTRODUCTION

1.1 Purpose

This document is a walkthrough workbook (WTWB) to facilitate the formal review of the Advanced Transportation Controller (ATC) Application Programming Interface Validation Suite (APIVS) Software Design Document (SDD). It is a tool to help reviewers to verify that the design of the software to be developed will meet the requirements of the ATC 5401 Standard. It provides traceability from the requirements in the ATC APIVS Software Requirements Specification (SRS) to the design elements in the SDD to ensure that each requirement is completely addressed. This WTWB has been developed for:

a) The consulting team contracted to develop the software described; and

b) The consultants, manufacturers, and public transportation professionals who participate in the API Working Group (WG) who provide domain expertise, quality assurance, testing assistance and ultimately the maintenance of the software; and

1.2 Scope

The focus of this WTWB is on the APIRI SDD document. In order to perform the review, ready access to the ATC APIVS SRS may be necessary to provide context for the requirements listed below. It is possible that anomalies in the requirements of the APIVS SRS may be discovered. If an anomaly is found, it will be noted for a later discussion and the review of the SDD will continue.

2 REFERENCES

Institute of Electrical and Electronics Engineers, *IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions*. IEEE, 1998.
http://standards.ieee.org/index.html

Institute of Electrical and Electronics Engineers, *IEEE Std 10287-1997, IEEE Standard for Software Design Reviews*. IEEE, 1997.
http://standards.ieee.org/index.html

Institute of Transportation Engineers, *ATC 5201 Advanced Transportation Controller (ATC) Standard Version 06*. ATC Joint Committee, 30 July 2012.
http://www.ite.org/standards/index.asp

Institute of Transportation Engineers, *ATC 5401 Application Programming Interface (API) Standard for the Advanced Transportation Controller (ATC) v02*. ATC Joint Committee, 15 September 2013.
http://www.ite.org/standards/index.asp

Institute of Transportation Engineers, *Advanced Transportation Controller (ATC) Application Programming Interface (API) Validation Suite (APIVS) Software Requirements Specification (SRS) v02.03*. ATC Joint Committee, 20 September 2014. http://www.ite.org/standards/index.asp

Institute of Transportation Engineers, *Software Design Description (SDD) for the Advanced Transportation Controller (ATC) Application Programming Interface (API) Validation Suite v01.00.* ATC Joint Committee, 4 December 2014. http://www.ite.org/standards/index.asp

3 DESIGN ANALYSIS

In accordance with IEEE Std 1016-1998, the APIRVS SDD identifies various "design entities" which are elements of the design which are structurally and functionally distinct from other elements. They are separately named and referenced. The design attributes of these entities are organized into several "design views" to reveal aspects of the APIVS software design. The design views are as follows:

- Decomposition Description – This view partitions the software into design entities;
- Dependency Description – This view describes the relationships among the entities and the system resources;
- Interface Description – This view lists everything a designer, programmer, or tester needs to know to use the design entities that make up the system; and
- Detail Description – This view provides the internal design details of a design entity.

The design analysis should reveal that:

1) Each APIVS SRS requirement is fulfilled by at least one entity and design view;
2) Related design views are consistent with each other;
3) The design is "sufficient" or "insufficient" for satisfying the requirement; and
4) Additional modifications (if any) are necessary to make the SDD acceptable.

4 GENERAL NOTES AND ACTIONS FROM THE WALKTHROUGH

SDD Section	Notes and Actions
General	*Discussed that the API Standard should include a requirement for the loopback drivers. This to be put into the project team's list of enhancements.* ***[To be done as part of the API 5401 Update.]*** *Overall, the SDD needs to answer the requirements in the design details. The narrative more of a summary.* ***[Change made as suggested.]***
1	*Add VSE to glossary.* ***[Change made as suggested.]*** *Why are we using IEEE 1016-1998?* ***[Contract requirement.]***
2	*Clean-up the use of "test hardware platform" from Section 2.* ***[Change made as suggested.]*** *It was suggested that a diagram from the APIVS SRS could be included in the SDD for context.* ***[Change made as suggested.]***
3	*It was suggested that the diagrams for the APIVS decomposition be done in a manner that maintains the layered architecture of the API software.* ***[Change made as suggested.]***
4	*None*
Appendices	*None*

5 WALKTHROUGH MATRIX

Req ID	Requirement Description	Design Sufficient to Fulfill Requirement and API Functions?	Actions / Notes
3.1	**No Cost Distribution** The APIVS software shall not have any content or component that requires a fee for the distribution of its source code and documentation.	*Yes*	*Design and/or narrative adequately cover item.*
3.2	**Open Source** The APIVS software shall be available to anyone through an Open Source Software (OSS) environment consistent with the USDOT approved API Reference Implementation Project Open Source Concept Paper (see Section 1.4).	*Yes*	*Design and/or narrative adequately cover item.*
3.3	**ITE Approved Software License** The APIVS software shall have all source code distributed using a software license that is approved by ITE.	*Yes*	*Design and/or narrative adequately cover item.*
3.4	**Unrestricted Use by Users** The APIVS software shall have a license that allows for unrestricted use by the software by users.	*Yes*	*Design and/or narrative adequately cover item.*
3.5	**Redistribution of Modified Source Code** The APIVS software shall have a license that requires entities that make enhancements to the APIVS software and redistribute it, to provide the source code for those enhancements publically and to ITE.	*Investigate/ Update*	*End users should be advised in the APIVS documentation to request the source code for the APIVS they are using to test the API software. Also, users should specify a particular version or later of the APIVS. Suggest that it is stated as unmodified.* ***[To be included in User's Manual or other documentation.]*** *Remove "and ITE" from the requirement in the SRS. It is not attainable.* ***[Change made as suggested.]***
3.6	**No Cost Redistributed APIVS Source Code** The APIVS software shall have a license that requires redistributed APIVS source code to be at no cost to users.	*Correction*	*State the software will use a GPL license.* ***[Change made as suggested.]***

Req ID	Requirement Description	Design Sufficient to Fulfill Requirement and API Functions?	Actions / Notes
3.7	**Testing Environment** The APIVS software shall be operational within the test environment and testing approach described in Section 2.1.	*Correction*	*State that all of the software is designed to run on an ATC Controller. Suggest addition to Section 2 and reference here.* ***[Change made as suggested.]***
3.8	**C Programming Language** The APIVS software shall be written using the C programming language as described by ISO/IEC 9899:2011" (see APIVS SRS Section 1.4).	*Investigate*	*State that the software is being written in a manner that is compatible with uclibc. In a future version of the SRS, we may include a requirement for uclibc and/or glibc. There may be issues with the how these libraries are specified in ATC 5201. Add trace to a place where it is located in document.* ***[Change made as suggested.]***
3.9	**Source Code Quality** The APIVS software shall be written in a fashion consistent with the GNU Coding Standards (see Section 1.4).	*Yes*	*Design and/or narrative adequately cover item.*
3.10	**XML Scripting Language** The APIVS software shall use an XML (Extensible Markup Language) based scripting language to define tests.	*Yes*	*Design and/or narrative adequately cover item.*
3.11	**Interpreted Test Scripts** The APIVS software shall execute XML defined tests without recompilation of the APIVS software.	*Yes*	*Design and/or narrative adequately cover item.*
3.12	**Run All Tests** The APIVS software shall have a user option to run all of the tests available in the test suite.	*Yes*	*Design and/or narrative adequately cover item.*

Req ID	Requirement Description	Design Sufficient to Fulfill Requirement and API Functions?	Actions / Notes
3.13	**Run Selected Tests** The APIVS software shall have a user option to run a selected subset of the tests that are available in the test suite.	*Investigate*	*The APIVS software may be configured to run one or more individual tests from the full set of tests supplied in the distribution.* *How will the user select the tests? Need to describe how this will work in the SDD.* *Add explicit sentence(s) that address this requirement in Section 2.2.* ***[Change made as suggested.]*** *Provide I/O details in Section 3.3 and 3.4 for each of the entities. Need to add the mechanisms to accomplish the capability (i.e. Data structures).* ***[Change made suggested. However, I/O is only covers external APIVS aspects. Some of the entities are simply functions within the same body of code.]***
3.14	**Continuous Loop** The APIVS software shall have a user option to run selected subsets of tests in the test suite sequentially in a continuous loop.	*Investigate*	*The APIVS software may be configured to run one or more individual tests from the full set of tests supplied in the distribution.* *Expand to show the method in which this is done.* *Run a number of times?* *Make it configurable to abort on any error or to continue and log. Ability to abort the test and preserve the log file.* *Add explicit sentence(s) that address this requirement in Section 2.2.* ***[Change made as suggested.]*** *Provide I/O details in Section 3.3 and 3.4 for each of the entities. Need to add the mechanisms to accomplish the capability (i.e. Data structures).* ***[The entities are the same as for any other test.]***
3.15	**Conformance Indication** The APIVS software shall return a value of 0 when a set of tests selected from tests available in the test suite are found to conform to the ATC 5401 Standard.	*Update*	*Change narrative to use VSE instead of APIVS software.* *Also, reference section in body of document that addresses it.* ***[Change made as suggested.]***
3.16	**Nonconformance Indication** The APIVS software shall return value of -1 when a set of tests selected from tests available in the test suite are found not to conform to the ATC 5401 Standard.	*Update*	*Change narrative to use VSE instead of APIVS software.* *Also, reference section in body of document that addresses it.* ***[Change made as suggested.]***

Req ID	Requirement Description	Design Sufficient to Fulfill Requirement and API Functions?	Actions / Notes
3.17	**Detailed Log** The APIVS software shall have a user option to produce a detailed log of the tests performed including: • The library, function and arguments on an API function call and the return values; • If a function fails, guidance to the user on the cause of the failure; • The test being executed; • Line # in the APIVS source code; • Step in the test; and • Time stamps for each step in the test.	*Investigate*	*Need to show that the requirement is being satisfied. Show an example output? Look to show in the design to accomplish the requirement.* *Might also add the reference in the body of the document. Have a description of the log file.* ***[Changed made as suggested.]***
3.18	**Summary Result** The APIVS software shall have a summary result that lists each test performed and the conformance/nonconformance result.	*Investigate*	*The format of the conformance report, as described in 3.4.6, allows interactive viewing of test results at increasing levels of detail.* *List the format of each of the types of output.* *Maybe include in an appendix a screen grab of the output file. Reference it from the Section.* *Identify the details in section 3.4.6 and/or other section. Describe the format of the XML file.* ***[Changed made as suggested. Example XML output file included in appendix.]***
3.19	**Output Options** The APIVS software shall have output options as follows: a) Conformance/nonconformance Indication only; b) Conformance/nonconformance indication and summary result; and c) Conformance/nonconformance indication, summary result and all logs and traces.	*Investigate*	*The format of the conformance report, as described in 3.4.6, allows interactive viewing of test results at increasing levels of detail.* *List the format of each of the types of output.* *Maybe include in an appendix a screen grab of the output file. Reference it from the Section.* *Identify the details in section 3.4.6 and/or other section. Describe the format of the XML file. How to select the options?* ***[Changed made as suggested. Example XML output file included in appendix.]***

Req ID	Requirement Description	Design Sufficient to Fulfill Requirement and API Functions?	Actions / Notes
3.20	**XML Output Files** The APIVS software shall output any summary, log or traces into a file in an XML format.	*Investigate*	*Show how this is done. Need description of how the multiple runs will be handled. Suggest that the summary result is added to the end of the continuous report.* *SDD 01.01 narrative will be sufficient if the other items in this section are addressed.* ***[The summary report of an iteration of the test case is inline in the output file. It is not aggregated. Will look at the posiibility of such an output as the project continues.]***
3.21.1	**FPUI Library C Function Present** The APIVS software shall validate that each FPUI function defined in Section 4.1 of the ATC 5401 Standard is present in the API software.	*Investigate*	*A validation test configuration file "FPUI_every_func.xml" is provided in the APIVS software distribution which includes calls to all FPUI library functions and indicates any absent functions in the conformance report.* *Need description in the body of the document for FPUI_every_func.xml content and how it is used.* ***[Change made as suggested. The details of the XML test files will be available following the integrated testing.]***
3.21.2	**FPUI Library C Function Conforming Arguments** The APIVS software shall validate that each FPUI function has the correct arguments as defined in Section 4.1 of the ATC 5401 Standard.	*Investigate*	*Add entities involved.* *Possibly include a snippet of XML that shows how to do it.* *API WG to discuss the level of testing (parameter checking) and user definition required.* *Possibly include the XML description in an appendix.* ***[Change made as suggested. The details of the XML test files will be available following the integrated testing.]***
3.21.3	**FPUI Library C Function Error Checking** The APIVS software shall validate that each FPUI function returns the correct error codes for the error conditions defined in Section 4.1 of the ATC 5401 Standard.	*Investigate*	*Needs detail on how this is done. Each function has its own test routine.* *Possibly include a snippet of XML that shows how to do it.* *API WG to discuss the level of testing (parameter checking) and user definition required.* *Possibly include the XML description in an appendix.* ***[Change made as suggested. The details of the XML test files will be available following the integrated testing.]***

Req ID	Requirement Description	Design Sufficient to Fulfill Requirement and API Functions?	Actions / Notes
3.21.4	**FPUI Library C Function Argument Boundary Checking** The APIVS software shall validate that the boundaries of the arguments to the FPUI functions operate as defined in Section 4.1 of the ATC 5401 Standard.	*Investigate*	*Needs detail on how this is done. Each function has its own test routine.* *Possibly include a snippet of XML that shows how to do it.* *API WG to discuss the level of testing (parameter checking) and user definition required.* *Possibly include the XML description in an appendix.* ***[Examples of XML test files will be will be available following the integrated testing.]***
3.21.5	**FPUI Library Composite Testing** The APIVS software shall validate that each FPUI function operates correctly under typical operating conditions with other API functions using at least one composite test.	*Investigate*	*Reference 3.4.14. Possibly include a snippet of XML that shows how to do it.* ***[Example descriptions are included in Appendix C. Examples of XML test files will be will be available following the integrated testing.]***
3.21.6	**Front Panel Manager Window Testing** The APIVS software shall perform predefined tests that validate that the Front Panel Manager Window operates per the requirements established in Section 3.1.1.1 of the ATC 5401 Standard.	*Investigate*	*A validation test configuration file is provided in the APIVS software distribution and described in 3.4.14, which calls multiple FPUI library functions in a scenario which validates typical use of the Front Panel Manager Window system, checking each function call for correctness and validating the resultant controller behavior against expected results.* *Expect to see the requirements from Section 3.1.1.1 of the ATC 5401 Standard addressed in the design section of the SDD.* ***[These will be fulfilled in XML scenario tests.]*** *Look at some connective statements for the sections in this area.* *Put scenario stuff in appendix. Reference appendix from section that uses it. Reference section/appendix in narrative.* ***[This is addressed by the test scenarios to be developed. See Appendix C for details.]*** *Use terminology that will be used across documents (i.e. Test Cases vs. test configuration files). Correct throughout document.* ***[Change made as suggested.]***

Req ID	Requirement Description	Design Sufficient to Fulfill Requirement and API Functions?	Actions / Notes
3.21.7	**ATC Configuration Window Testing** The APIVS software shall perform predefined tests that validate that the ATC Configuration Window operates per the requirements established in Section 3.2.1 of the ATC 5401 Standard.	*Investigate*	*A validation test configuration file is provided in the APIVS software distribution and described in 3.4.15, which calls multiple FPUI library functions in a scenario which validates typical use of the ATC Configuration Window system, checking each function call for correctness and validating the resultant controller behavior against expected results.* *Expect to see the requirements from Section 3.2.1 of the ATC 5401 Standard addressed in the design section of the SDD.* ***[These will be fulfilled in XML scenario tests.]*** *Put scenario stuff in appendix. Reference appendix from section that uses it. Reference section/appendix in narrative.* ***[This is addressed by the test scenarios to be developed. See Appendix C for details.]***
3.21.8	**Configuration Utility Testing** The APIVS software shall perform predefined tests that validate that the Configuration Utilities operate per the requirements established in Sections 3.2.2 through 3.2.6 of the ATC 5401 Standard.	*Investigate*	*Expect to see the requirements from Section 3.2.2-3.2.6 of the ATC 5401 Standard addressed in the design section of the SDD.* ***[These will be fulfilled in XML scenario tests.]*** *Address concerns with the configuration utility issues.* ***[This is addressed by the test scenarios to be developed.]*** *Put scenario stuff in appendix. Reference appendix from section that uses it. Reference section/appendix in narrative.* ***[This is addressed by the test scenarios to be developed.]***
3.22.1	**FIO Library C Function Present** The APIVS software shall validate that each FIO function defined in Section 4.1 of the ATC 5401 Standard is present in the API software.	*Investigate*	*A validation test configuration file "FIO_every_func.xml" is provided in the APIVS software distribution which includes calls to all FIO library functions and indicates any absent functions in the conformance report.* *Need description in the body of the document for FIO_every_func.xml content and how it is used.* ***[Change made as suggested. The details of the XML test files will be available following the integrated testing.]***

Req ID	Requirement Description	Design Sufficient to Fulfill Requirement and API Functions?	Actions / Notes
3.22.2	**FIO Library C Function Conforming Arguments** The APIVS software shall validate that each FIO function has the correct arguments as defined in Section 4.1 of the ATC 5401 Standard. .	*Investigate*	*Add entities involved.* *Possibly include a snippet of XML that shows how to do it.* *API WG to discuss the level of testing (parameter checking) and user definition required.* *Possibly include the XML description in an appendix.* ***[Change made as suggested. The details of the XML test files will be available following the integrated testing.]***
3.22.3	**FIO Library C Function Error Checking** The APIVS software shall validate that each FIO function returns the correct error codes for the error conditions defined in Section 4.1 of the ATC 5401 Standard.	*Investigate*	*Needs detail on how this is done. Each function has its own test routine.* ***[Change made as suggested. The details of the XML test files will be available following the integrated testing.]***
3.22.4	**FIO Library C Function Argument Boundary Checking** The APIVS software shall validate that the boundaries of the arguments to the FIO functions operate as defined in Section 4.1 of the ATC 5401 Standard.	*Investigate*	*Needs detail on how this is done. Each function has its own test routine.* ***[Examples of XML test files will be will be available following the integrated testing.]***
3.22.5	**FIO Library Composite Testing** The APIVS software shall validate that each FIO function operates correctly under typical operating conditions with other API functions using at least one composite test.	*Investigate*	*A validation test configuration file "FIO_every_func.xml" is provided in the APIVS software distribution which includes calls to all FIO library functions and indicates any absent functions in the conformance report.* *Should have a list of the APIs exercised. Maybe other items.* *Reference 3.4.14. Possibly include a snippet of XML that shows how to do it.* ***[Example descriptions are included in Appendix C. Examples of XML test files will be will be available following the integrated testing.]***

Req ID	Requirement Description	Design Sufficient to Fulfill Requirement and API Functions?	Actions / Notes
3.22.6	**Field I/O Manager Testing** The APIVS software shall perform predefined tests that validate that the Field I/O Manager operates per the requirements established in Section 3.1.2 of the ATC 5401 Standard.	*Investigate*	*A validation test configuration file is provided in the APIVS software distribution and described in X.Y.Z which calls multiple FIO library functions in a scenario which validates typical use of the Field I/O checking each function call for correctness and validating the resultant controller behavior against expected results.* *Expect to see the requirements from Section W.X.Y.Z of the ATC 5401 Standard addressed in the design section of the SDD.* ***[These will be fulfilled in XML scenario tests.]***
3.23.1	**TOD Library C Function Present** The APIVS software shall validate that each TOD function defined in Section 4.1 of the ATC 5401 Standard is present in the API software.	*Investigate*	*A validation test configuration file "TOD_every_func.xml" is provided in the APIVS software distribution which includes calls to all TOD library functions and indicates any absent functions in the conformance report.* ***[Change made as suggested. The details of the XML test files will be available following the integrated testing.]***
3.23.2	**TOD Library C Function Conforming Arguments** The APIVS software shall validate that each TOD function has the correct arguments as defined in Section 4.1 of the ATC 5401 Standard.	*Investigate*	***[Change made as suggested. The details of the XML test files will be available following the integrated testing.]***
3.23.3	**TOD Library C Function Error Checking** The APIVS software shall validate that each TOD function returns the correct error codes for the error conditions defined in Section 4.1 of the ATC 5401 Standard.	*Investigate*	*Needs detail on how this is done. Each function has its own test routine.* ***[Change made as suggested. The details of the XML test files will be available following the integrated testing.]***
3.23.4	**TOD Library C Function Argument Boundary Checking** The APIVS software shall validate that the boundaries of the arguments to the TOD functions operate as defined in Section 4.1 of the ATC 5401 Standard.	*Investigate*	*Needs detail on how this is done. Each function has its own test routine.* ***[Examples of XML test files will be will be available following the integrated testing.]***

Req ID	Requirement Description	Design Sufficient to Fulfill Requirement and API Functions?	Actions / Notes
3.23.5	**TOD Library Composite Testing** The APIVS software shall validate that each TOD function operates correctly under typical operating conditions with other API functions using at least one composite test.	*Investigate*	*A validation test configuration file "TOD_every_func.xml" is provided in the APIVS software distribution which includes calls to all TOD library functions and indicates any absent functions in the conformance report. Should have a list of the APIs exercised. Maybe other items.* ***[Example descriptions are included in Appendix C. Examples of XML test files will be will be available following the integrated testing.]***
3.24	**Multiple and Concurrent Applications** The APIVS software shall perform predefined tests that validate 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. [Guidance: This requirement could be met with multiple subprocesses or threads of the same test application program.]	*Investigate*	*The APIVS software supports the ability for multiple instances of the VSE program to run concurrently in order to test multiple client support of the API libraries.* *Suggest that instead of multiple instances, that separate apps could be run.* ***[Alternatives are now included in Section 2.2.]***