## CHANGE HISTORY

<table>
<thead>
<tr>
<th>DATE</th>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/18/16</td>
<td>v01.00 Initial Draft APIRI Policies and Procedures</td>
</tr>
<tr>
<td>09/22/16</td>
<td>v01.01 Complete document.</td>
</tr>
<tr>
<td>10/04/16</td>
<td>v01.02 Updates per API Working Group.</td>
</tr>
</tbody>
</table>
## CONTENTS

1 INTRODUCTION .............................................................................................................................. 4
   1.1 Purpose ............................................................................................................................... 4
   1.2 Scope .................................................................................................................................. 4
   1.3 Document Organization ...................................................................................................... 4

2 MISSION STATEMENT ................................................................................................................... 5

3 PROJECT ROLES ........................................................................................................................... 5

4 POLICY, PROCESS AND PROCEDURE MANAGEMENT ............................................................ 6
   4.1 Policy, Process and Procedure Development .................................................................... 6
   4.2 Policy, Process and Procedure Style .................................................................................. 6

5 POLICIES, PROCESSES AND PROCEDURES DASHBOARD .................................................... 7

6 POLICIES ........................................................................................................................................ 7
   6.1 Consistency with the ATC 5401 Standard .......................................................................... 7
   6.2 Open and Free Software ..................................................................................................... 8

7 PROCESSES ................................................................................................................................... 8
   7.1 Integration Workflow ........................................................................................................... 8

8 PROCEDURES .............................................................................................................................. 10
   8.1 Submitting an Issue ........................................................................................................... 11
   8.2 Submitting a Pull Request ................................................................................................. 11

APPENDICES ............................................................................................................................................. 13

APPENDIX A: GLOSSARY .................................................................................................................. 14

APPENDIX B: REFERENCES ............................................................................................................... 17

APPENDIX C: PROPOSED FUTURE UPDATES...................................................................................... 18
1 INTRODUCTION

1.1 Purpose

This policies, processes and procedures document provides the operational guidelines for protecting and maintaining the software and documents that make up the enterprise known as the Advanced Transportation Controller (ATC) Application Programming Interface (API) Reference Implementation Project (APIRI Project). They have been developed for:

a) The USDOT Intelligent Transportation Systems (ITS) Joint Program Office (JPO) which has sponsored much of the work;

b) The Institute of Transportation Engineers (ITE) which oversees the project on behalf of the standard development organizations (SDOs) making up the ATC standards program.

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

d) The transportation industry as a whole that depends upon the software produced from this project to support operational programs on ATC controller equipment; and

e) Any party that wishes to contribute to the project.

1.2 Scope

The policies, processes and procedures in this project apply to the two major products of the APIRI Project: 1) the Application Programming Interface Reference Implementation Software (APIRI Software) which is an implementation of the ATC 5401 Standard developed in an open source software (OSS) environment and 2) the API Validation Suite Software (APIVS Software) which is used to validate the APIRI software and is also implemented in an OSS environment. The products of the APIRI Project are maintained in two GitHub repositories as follows:

- https://github.com/apiriadmin/APIRI
- https://github.com/apiriadmin/APIVS

This policies, processes and procedures document is intended to be sufficient for purpose as determined by the API WG. It is not a comprehensive organizational-type document or manual. It is used to establish policies that are deemed important to maintaining the integrity of the APIRI Project products and to define processes and procedures that insure repeated success.

1.3 Document Organization

The remainder of this document has the following sections:

- Mission Statement – The purpose of the APIRI Project;
- Project Roles – Roles within the APIRI Project to which the policies, processes and procedures apply;
- Policy, Process and Procedure Management – Describes how policies, processes and procedures are developed and maintained;
- Policy, Process and Procedure Dashboard – A table showing the associations of the policies, processes and procedures with links to the sections where they are described;
- Policies – This design view presents the high level interfaces of the software;
- Processes – This design view highlights any dependencies on the software to be developed;
- Procedures – This design view presents detailed information for each design entity; and
- Appendices – Glossary, References, Proposed Additions and Modifications.
Policies, processes and procedures serve different purposes:

- **A policy** is a guideline or statement of position with respect to a given topic;
- **A process** is the highest level description of a large task or series of related tasks; and
- **A procedure** tells how a series of sequential tasks are performed to achieve a specific outcome.

For purposes of this document, procedures may include forms, checklists, templates, or other items used in the operation of the project.

A policy may or may not have an associated process or procedure. A process or procedure may not be necessary in some circumstances for a process or procedure may not be defined at the time the policy is established. A policy may be associated to multiple processes and procedures.

## 2 MISSION STATEMENT

The mission of the Application Programming Interface Reference Implementation Project is to facilitate the development of application programs for Advanced Transportation Controller users that are compatible, portable and interchangeable. This is accomplished by providing an open source reference implementation of the software defined by the ATC 5401 Standard that is consistent, reliable, free of cost to users, and maintainable by the transportation community.

## 3 PROJECT ROLES

To achieve the mission of the APIRI Project particular roles are identified that are referenced in the policies, processes and procedures.

- **User** – An individual who downloads any information from the APIRI Project repositories.
- **Contributor** – A user that also makes contributions to the APIRI Project repositories.
- **Developer** – A contributor who is a software developer.
- **Integration Manager (IM)** – A contributor that is appointed by the ATC Standards Manager to maintain the APIRI Project repositories for the users and contributors. The IM insures that the policies and procedures for the APIRI Project are followed and determines when to merge or close Pull Requests.
- **ATC Standards Manager** – This individual is the manager of the ATC Standards Program for the Institute of Transportation Engineers. The ATC Standards Manager appoints the Integration Manager for the APIRI Project repositories.
- **API Working Group** – A technical subcommittee of the ATC Joint Committee that is responsible for the development and maintenance of the ATC 5401 Standard. It is the responsibility of the API WG to approve the policies, processes and procedures for the APIRI project. Minor corrections or additions to the policies and procedures may be approved by the Chairs of the API WG.
- **API Working Group Chairs** – Leaders of the API WG identified by the ATC Joint Committee Chair.
- **ATC Joint Committee** – A committee that oversees the development and publication ATC standards. The ATC Joint Committee (JC) is of made up of representatives from three standards development organizations (SDOs): the American Association of State Highway and...
Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE) and the National Electrical Manufacturers Association (NEMA).

- **ATC Joint Committee Chair** – The leader of the ATC Joint Committee appointed by agreement of the members of the ATC JC. The ATC JC Chair appoints the API WG Chairs.

4 **POLICY, PROCESS AND PROCEDURE MANAGEMENT**

4.1 **Policy, Process and Procedure Development**

It is the responsibility of the API WG to approve the policies, processes and procedures for the APIRI Project. The IM reviews the policies, processes, and procedures with the API WG on a yearly basis to determine their effectiveness. The steps for establishing a new policy, process or procedure are as follows:

1) Any contributor to the project may make a proposal to the API WG Chairs or the IM that the policies, processes and procedures be updated to cover a matter concerning the APIRI Project.
2) The API WG Chairs and the IM discuss the proposal and determine if it is appropriate for consideration by the API WG.
3) If the API WG Chairs and the IM determine that the proposal is appropriate for API WG consideration, then the proposal is presented to the API WG for discussion; otherwise, the proposal is denied and the contributor is notified.
4) If it is the consensus of the API WG that the update to the policies, processes and procedures is to be developed, then a) a resource (or resource team) is identified to draft the policies, processes and procedures and b) schedule is established for completion of the work; otherwise, the proposal is denied and the contributor is notified.
5) If a resource is not identified or a schedule is not established, then the effort does not proceed and the contributor is notified.
6) The modifications to the draft policies, processes and procedures are made and submitted to API WG for review.
7) If it is the consensus of the API WG that the draft policies, processes and procedures are acceptable, then the IM integrates the updated policies, processes and procedures into the APIRI Project as appropriate; otherwise,
   a) The API WG provides comments for suggested changes and the development continues from Step 5 or
   b) The work on the policies, processes and procedures concludes (does not continue) and the contributor is notified.

Minor corrections or additions to the policies, processes and procedures may be made by the IM or another contributor and approved by the Chairs of the API WG. If a proposal for an update to the policies, processes and procedures does not go forward because a resource was not available or the work could not be completed in a timely manner, the IM may add the proposal to Appendix C of this document for future consideration.

4.2 **Policy, Process and Procedure Style**

Policies, processes and procedures are to be written in the present tense. The word “shall” is to be avoided. Instead of using “shall” use: "must" for an obligation, "must not" for a prohibition, "may" for a discretionary action and "should" for a recommendation.
5 POLICIES, PROCESSES AND PROCEDURES DASHBOARD

The dashboard below is hyperlinked. Ctrl+Click the section number to go to the definition of the policy, process or procedure.

<table>
<thead>
<tr>
<th>POLICY</th>
<th>PROCESS</th>
<th>PROCEDURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Consistency with the ATC 5401 Standard</td>
<td>7.1 Integration Workflow</td>
<td>8.1 Submitting an Issue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.2 Submitting a Pull Request</td>
</tr>
<tr>
<td>6.2 Open and Free Software</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6 POLICIES

This section contains the specific policies of the APIRI Project and Repositories.

Each policy has the following outline:

- Section Number and Title – The title is not to be excessively long.
- Effective Date – Date when the policy took effect.
- Approved By – Typically the IM or the API Working Group.
- Policy Contact – The point of contact when the policy needs to change. The point of contact is to be one of the project roles identified in Section 3.
- Supersedes – Title and date of any policies superseded by this policy.
- Last Reviewed/Updated – Date when the policy was last reviewed or updated.
- Applies To – The project roles of those the policy applies to.
- History – History of major changes to the policy.
- Related Policies – Policies that have a related concern or are causally connected.
- Policy Statement – State the policy. Include: a) the situation(s) when this policy does or does not apply, b) the major conditions or restrictions and c) any exclusions or special situations.

Definitions of terms and acronyms used in the policies are included Appendix A, Acronyms, Abbreviations and Definitions.

6.1 Consistency with the ATC 5401 Standard

Effective Date:        July 28, 2016
Approved By:          (Under review by the API Working Group)
Policy Contact:        Integration Manager
Supersedes:            None
Last Reviewed/Updated: July 28, 2016
Applies To:            Contributors
History:               None
Related Policies:      None

Policy Statement:
It is the policy of the APIRI Project that there be releases, called formal releases of APIRI Software and APIVS Software that are representative of a Jointly Approved version of the ATC 5401 Standard. The
purpose is so that agencies that reference the ATC 5401 Standard in their procurement specifications have corresponding representative APIRI Software available for use by manufacturers. The formal releases of software include corresponding releases of user manuals and test documentation including anything necessary to run a successful test of the APIVS software on the APIRI software. It is not necessary that any Concept of Operations, Software Requirements Specifications, or Software Design Documents be updated unless funded to do so by the USDOT or some other party.

6.2 Open and Free Software

Effective Date: July 28, 2016
Approved By: (Under review by the API Working Group)
Policy Contact: Integration Manager
Supersedes: None
Last Reviewed/Updated: July 28, 2016
Applies To: Contributors
History: None
Related Policies: None

Policy Statement:
It is the policy of the APIRI Project that the software and documents of the associated repositories are open source, free of cost, and free for use subject to the licensing or copyright restrictions which govern the various products.

7 PROCESSES

This section contains the specific processes of the APIRI Project and Repositories.

Each process has the following outline:

- Section Number and Title – The title is not to be excessively long.
- Effective Date – Date when the process took effect.
- Approved By – Typically the IM or the API Working Group.
- Process Contact – The point of contact when the process needs to change. The point of contact is to be one of the project roles identified in Section 3.
- Supersedes – Title and date of any processes superseded by this process.
- Last Reviewed/Updated – Date when the process was last reviewed or updated.
- Applies To – The project roles of those the process applies to.
- Process Statement – State the process.
- History – History of major changes to the process.

Definitions of terms and acronyms used in the policies are included Appendix A, Acronyms, Abbreviations and Definitions.

7.1 Integration Workflow

Effective Date: July 28, 2016
Approved By: (Under review by the API Working Group)
Process Contact: Integration Manager
Supersedes: None
Process Statement:

The APIRI Project uses an Integration Workflow where an Integration Manager (IM) determines when a Pull Request from a contributor is merged into a managed branch of the project or is otherwise closed. The IM uses input from other contributors when assessing a Pull Request. A progressive-stability type of branching is used to facilitate official releases of the most stable versions of the software as shown in Figure 1. Code that is stable and that has been or will be released is in the Master Branch. A Master Branch is intended to correspond to a particular version of the ATC 5401 Standard. Code that is less stable, possibly where features are still in development, or is still being tested, is in the Develop Branch. Topic Branches are typically short-lived branches to fix a bug or develop a feature until it is ready to be merged into the Develop Branch. It is possible that the IM may choose to have more than one level of Develop Branches. Contributions from Developers are only accepted by the IM from public branches within the project.

In practice, branches are named to be more descriptive as shown in the example in Figure 2. In this illustration, a Develop Branch was started at some point called “APIRI v01 Dev.” There were 17 commits to the branch before it was deemed ready for deployment. At that point a master branch was started named “APIRI v01.” Later two more commits, C18 and C19, were made to the APIRI v01 Dev branch. At some point the IM determined that commit C18 was stable and it was merged to the APIRI v01 branch as C20. A branch named Issue #11 was started by a contributor working on a bug fix which is not yet ready for merging up to the APIRI v01 Dev branch.

![Figure 1. Progressive-Stability Branching.](image-url)
8 PROCEDURES

This section contains the specific procedures of the APIRI Project and Repositories.

Each procedure has the following outline:

- Section Number and Title – The title is not to be excessively long.
- Effective Date – Date when the procedure took effect.
- Approved By – Typically the IM or the API Working Group.
- Procedure Contact – The point of contact when the procedure needs to change. The point of contact is to be one of the project roles identified in Section 3.
- Supersedes – Title and date of any procedures superseded by this procedure.
- Last Reviewed/Updated – Date when the procedure was last reviewed or updated.
- Applies To – The project roles of those the procedure applies to.
- Procedure Statement – State the procedure. A procedure may include forms, checklists, templates, or other items used in the operation of the project.

Definitions of terms and acronyms used in the policies are included Appendix A, Acronyms, Abbreviations and Definitions.
8.1 Submitting an Issue

The following template is to be used when submitting an issue due to a bug or for suggesting a change/improvement.

```markdown
<!--- Provide a general summary of the issue in the Title above -->

### Expected Behavior
<!--- If you're describing a bug, tell us what should happen -->
<!--- If you're suggesting a change/improvement, tell us how it should work -->

### Current Behavior
<!--- If describing a bug, tell us what happens instead of the expected behavior -->
<!--- If suggesting a change/improvement, explain the difference from current behavior -->

### Possible Solution
<!--- Not obligatory, but suggest a fix/reason for the bug, -->
<!--- or ideas how to implement the addition or change -->

### Steps to Reproduce (for bugs)
<!--- Provide a link to a live example, or an unambiguous set of steps to -->
<!--- reproduce this bug. Include code to reproduce, if relevant -->
1. 
2. 
3. 
4. 

### Context
<!--- How has this issue affected you? What are you trying to accomplish? -->
<!--- Providing context helps us come up with a solution that is most useful in the real world -->

### Your Environment
<!--- Include as many relevant details about the environment you experienced the bug in -->
* Version of the Project:
* Operating System and Version:
* Hardware Specifications:

---------------------------------------------------------------
Attach files by dragging & dropping, selecting them, or pasting from the clipboard.
```

[Go Back to Dashboard]

8.2 Submitting a Pull Request

When preparing a pull request the following guidelines are to be followed:

- The pull request should address one issue. Keep extraneous or unrelated fixes out of the code submitted for the pull request.
- Keep code width and documentation to 80 characters to facilitate side-by-side comparisons.
- Avoid code reformatting unless there is a separate pull request to do so.
- Make sure the code builds.
- Verify that all tests pass.
- If there is not a test to validate the pull request, add one. Follow the testing strategy already in existence.
The following template is to be used when submitting the pull request.

```markdown
<!--- Provide a general summary of your changes in the Title above -->

## Description
<!--- Describe your changes in detail -->

## Related Issue
<!--- This project only accepts pull requests related to open issues -->
<!--- If suggesting a new feature or change, please discuss it in an issue first -->
<!--- If fixing a bug, there should be an issue describing it with steps to reproduce -->
<!--- Please link to the issue here: -->

## Motivation and Context
<!--- Why is this change required? What problem does it solve? -->

## How Has This Been Tested?
<!--- Please describe in detail how you tested your changes. -->
<!--- Include details of your testing environment, and the tests you ran to -->
<!--- see how your change affects other areas of the code, etc. -->

## Screenshots (if appropriate):

## Types of changes
<!--- What types of changes does your code introduce? Put an `x` in all the boxes that apply: -->
- [ ] Bug fix (non-breaking change which fixes an issue)
- [ ] New feature (non-breaking change which adds functionality)
- [ ] Breaking change (fix or feature that would cause existing functionality to change)

## Checklist:
<!--- Go over all the following points, and put an `x` in all the boxes that apply. -->
<!--- If you're unsure about any of these, don't hesitate to ask. We're here to help! -->
- [ ] My code follows the code style of this project.
- [ ] My change requires a change to the documentation.
- [ ] I have updated the documentation accordingly.
- [ ] I have read the **CONTRIBUTING** document.
- [ ] I have added tests to cover my changes.
- [ ] All new and existing tests passed.

---

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

[Go Back to Dashboard]
```
APPENDICES
Table 1. Glossary of Terms and Acronyms.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AASHTO</td>
<td>American Association of State Highway and Transportation Officials</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>APIRI Project</td>
<td>Entire project managed by this PMP including software, hardware and documentation.</td>
</tr>
<tr>
<td>APIRI Software</td>
<td>API Reference Implementation Software</td>
</tr>
<tr>
<td>APIVS Software</td>
<td>API Validation Suite Software</td>
</tr>
<tr>
<td>APIVSXML</td>
<td>APIVS Extensible Markup Language (XML) as defined by the <em>API Validation Suite APIVSXML Specification</em> (see Section 1.5 References). This version of XML includes elements for use with the APIVS software. APIVSXML is used to create test case specifications that are both human-readable and machine-readable. APIVSXML and XML are used synonymously within this document.</td>
</tr>
<tr>
<td>Application Program</td>
<td>Any program designed to perform a specific function directly for the user or, in some cases, for another application program. Examples of application programs include word processors, database programs, Web browsers and traffic control programs. Application programs use the services of a computer's O/S and other supporting programs such as an application programming interface.</td>
</tr>
<tr>
<td>API</td>
<td>Application Programmer Interface</td>
</tr>
<tr>
<td>ATC</td>
<td>Advanced Transportation Controller</td>
</tr>
<tr>
<td>ATC Device Drivers</td>
<td>Low-level software not included in a typical Linux distribution that is necessary for ATC-specific devices to operate in a Linux O/S environment.</td>
</tr>
<tr>
<td>ATP</td>
<td>Authorization to Proceed</td>
</tr>
<tr>
<td>Board Support Package</td>
<td>Software usually provided by processor board manufacturers which provides a consistent software interface for the unique architecture of the board. In the case of the ATC, the Board Support Package also includes the O/S</td>
</tr>
<tr>
<td>BSP</td>
<td>See Board Support Package</td>
</tr>
<tr>
<td>ConOps</td>
<td>Concept of Operations</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit. A programmable logic device that performs the instruction, logic and mathematical processing in a computer.</td>
</tr>
<tr>
<td>Device Driver</td>
<td>A software routine that links a peripheral device to the operating system. It acts like a translator between a device and the application programs that use it.</td>
</tr>
<tr>
<td>FHWA</td>
<td>Federal Highway Administration</td>
</tr>
<tr>
<td>FIO</td>
<td>Field Input and Output</td>
</tr>
<tr>
<td>FPUUI</td>
<td>Front Panel User Interface</td>
</tr>
<tr>
<td>Formal Release</td>
<td>A distribution of APIRI Software that represents a Jointly Approved version of the ATC 5401 Standard.</td>
</tr>
<tr>
<td>H/W</td>
<td>Hardware</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>ITE</td>
<td>Institute of Transportation Engineers</td>
</tr>
<tr>
<td>ITS</td>
<td>Intelligent Transportation Systems</td>
</tr>
<tr>
<td>JC</td>
<td>Joint Committee</td>
</tr>
<tr>
<td>Linux</td>
<td>Low-level software that is freely available in the Linux community for use with common hardware components operating in a standard fashion.</td>
</tr>
<tr>
<td>Linux Kernel</td>
<td>The Unix-like operating system kernel that was begun by Linus Torvalds in 1991. The Linux Kernel provides general O/S functionality. This includes functions for things typical in any computer system such as file I/O, serial I/O, interprocess communication and process scheduling. It also includes Linux utility functions necessary to run programs such as shell scripts and console commands. It is generally available as open source (free to the public). The Linux Kernel referenced in this standard is defined in the ATC Controller Standard Section 2.2.5, Annex A and Annex B.</td>
</tr>
<tr>
<td>Loopback Driver</td>
<td>A virtual device driver that loops back the output ports to a device to the input ports from a device without actually going to through the physical device.</td>
</tr>
<tr>
<td>N/A</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>O/S</td>
<td>Operating System</td>
</tr>
<tr>
<td>OSS</td>
<td>Open Source Software</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>PMP</td>
<td>Project Management Plan</td>
</tr>
<tr>
<td>POP</td>
<td>Period of Performance</td>
</tr>
<tr>
<td>PRL</td>
<td>Protocol Requirements List</td>
</tr>
<tr>
<td>RI</td>
<td>Reference Implementation</td>
</tr>
<tr>
<td>RITA</td>
<td>Research and Innovative Technology Administration</td>
</tr>
<tr>
<td>RTC</td>
<td>Real-Time Clock</td>
</tr>
<tr>
<td>RTM</td>
<td>Requirements Traceability Matrix</td>
</tr>
<tr>
<td>SDD</td>
<td>Software Design Descriptions</td>
</tr>
<tr>
<td>SDO</td>
<td>Standards Development Organization</td>
</tr>
<tr>
<td>SE</td>
<td>Systems Engineer</td>
</tr>
<tr>
<td>SEP</td>
<td>Systems Engineering Process</td>
</tr>
<tr>
<td>SEMP</td>
<td>Systems Engineering Management Plan</td>
</tr>
<tr>
<td>SOW</td>
<td>Statement of Work</td>
</tr>
<tr>
<td>SPDD</td>
<td>Serial Port Device Driver</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirements Specification</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell. An encrypted network protocol for initiating text-based shell sessions.</td>
</tr>
<tr>
<td>S/W</td>
<td>Software</td>
</tr>
<tr>
<td>TBD</td>
<td>To Be Determined</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>TCS</td>
<td>Test Case Specification</td>
</tr>
<tr>
<td>Test Case Specification</td>
<td>For the purposes of the APIVS software, a TCS is a test case file written in APIVSXML.</td>
</tr>
<tr>
<td>TOD</td>
<td>Time of Day</td>
</tr>
<tr>
<td>TOPR</td>
<td>Task Order Proposal Request</td>
</tr>
<tr>
<td>TX</td>
<td>Transmission</td>
</tr>
<tr>
<td>US</td>
<td>United States</td>
</tr>
<tr>
<td>USDOT</td>
<td>United States Department of Transportation</td>
</tr>
<tr>
<td>User Developer</td>
<td>A software developer that designs and develops programs for controllers.</td>
</tr>
<tr>
<td>VD</td>
<td>Virtual Display: the virtual front-panel display data maintained by the VSE during a test run.</td>
</tr>
<tr>
<td>VSE</td>
<td>Validation Suite Engine: the main executable program of the APIVS software.</td>
</tr>
<tr>
<td>Walkthrough</td>
<td>A step-by-step presentation by the author of a document in order to gather information and to establish a common understanding of its content.</td>
</tr>
<tr>
<td>WG</td>
<td>Working Group</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language. Used synonymously with APIVSXML within this document.</td>
</tr>
</tbody>
</table>
APPENDIX B: REFERENCES

http://standards.ieee.org/index.html

http://standards.ieee.org/index.html

http://www.ite.org/standards/index.asp

Institute of Transportation Engineers, *API Validation Suite APIVSXML Specification v02.00*. ATC Joint Committee, 31 December 2010.  
http://www.ite.org/standards/index.asp

Institute of Transportation Engineers, *Advanced Transportation Controller (ATC) Application Programming Interface (API) Validation Suite (APIVS) Concept of Operations (ConOps) v02.04*. ATC Joint Committee, 20 November 2014.  
http://www.ite.org/standards/index.asp

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, *ATC APIRI PMP v01.03 Project Management Plan (PMP) for the Advanced Transportation Controller (ATC) Application Programming Interface (API) Reference Implementation Project*. ATC Joint Committee, 5 November 2014.  
http://www.ite.org/standards/index.asp

http://www.ite.org/standards/index.asp

Institute of Transportation Engineers, *User Comment Draft ATC 5201 Advanced Transportation Controller (ATC) Standard Version 06.10*. ATC Joint Committee, 30 July 2012.  
http://www.ite.org/standards/index.asp


APPENDIX C: PROPOSED FUTURE UPDATES

1) A procedure for the IM in approving pull requests.

2) A procedure for a formal release. Include time frame, pre-release candidates. Possibly include users in this procedure.

3) A procedure for addressing issues. For bugs, include priorities. Critical bugs need resources immediately. May have a different procedure for enhancements.