

# API Standard v02.02 Comments Report

<b>Action Taken</b>	Accepted
<b>Comment ID</b>	12
<b>Commenter Link</b>	4
<b>Page</b>	9
<b>Paragraph</b>	1.2 Scope
<b>Comment</b>	Expand the description of the scope to elaborate on the use and maintenance of the API Standard. Specifically reference Appendix D.
<b>Response</b>	Will add a statement that the document is maintained under the direction of the ATC Joint Committee. The Scope adequately addresses the use of the standard. Appendix D is only there as a desire of the working group to have a reference implementation. It has not been determined if funding can be achieved to accomplish it.
<b>Status</b>	Closed
<b>Action Taken</b>	Accepted
<b>Comment ID</b>	2
<b>Commenter Link</b>	2
<b>Page</b>	
<b>Paragraph</b>	1.3 and others
<b>Comment</b>	Although the Standard references the use of a Linux Kernel within the controller, the standard does not address which version of the Kernel will be used. There are many proprietary Linux Kernels now on the market, among them are Sun Microsystems. VDOT's concern is that the controller will be locked in with a proprietary kernel and limit applications to one set of vendor drivers. VDOT would rather see a non-proprietary kernel such as the one found in the Public Linux Archive at kernel.org. VDOT would like to see the use of the Public Linux Kernel stated within the standard, also a prescribed method to upgrade the Kernel within the controller when new versions are available.
<b>Response</b>	The Linux Kernel is defined in the ATC Controller Standard (see ATC Controller Standard, Section 2.2.5, Annex A and Annex B). It is built on publicly available versions of the O/S. If specific drivers are necessary they are also called out in the ATC Controller Standard. The Section 1.3 definition of the API Standard will be modified to make this clear.
<b>Status</b>	Closed

**Action Taken** Accepted  
**Comment ID** 13  
**Commenter Link** 4  
**Page** 10  
**Paragraph** 1.3 Definitions, Acronyms and Abbreviation  
**Comment** Consider adding the following terms: Kbps, FAT, PCB, USB, BSP, AUX, SIU, CMU, BIU, MMU, SDLC, FIO, FSA, WINDOW / VIRTUAL DISPLAY SCREEN, FAULT MONITOR, WATCHDOG, HANDSHAKING, HEALTH MONITOR, DARK CHANNEL MAP, LOAD SWITCH FLASH, ELF, Cmd & Rsp.  
**Response** Will add abbreviations/terms for the following:  
Kbps, FAT, PCB, USB, BSP, SIU, CMU, BIU, MMU, SDLC, FIO, FSA, and ELF.  
  
The other terms were considered conceptual or requiring more than a simple definition.  
**Status** Closed

**Action Taken** Accepted  
**Comment ID** 14  
**Commenter Link** 4  
**Page** 10  
**Paragraph** 1.3 Definitions, Acronyms and Abbreviation  
**Comment** Consider adding "OS" to term "O/S" or change all to one style for consistency. See figure 5.  
**Response** Changes made as recommended.  
**Status** Closed

**Action Taken** Accepted  
**Comment ID** 15  
**Commenter Link** 4  
**Page** 10  
**Paragraph** 1.3 Definitions, Acronyms and Abbreviation  
**Comment** Recommend arranging terms in alphabetical order. For example, "API Managers" should be after "API".  
**Response** Changes made as recommended.  
**Status** Closed

**Action Taken** Accepted  
**Comment ID** 16  
**Commenter Link** 4  
**Page** 10  
**Paragraph** 1.3 Definitions, Acronyms and Abbreviation  
**Comment** Suggest expanding on the definitions for the following: CPU, LED.  
**Response** Changes made as recommended.  
**Status** Closed

**Action Taken** Accepted  
**Comment ID** 17  
**Commenter Link** 4  
**Page** 13  
**Paragraph** 1.5 Overview  
**Comment** Expand the description to expound on the Brackets in each section and how they are specifically indicating a user needs in Section 2, a specific requirement in Section 3 and an API function in Section 4. Discuss and reference Appendix A and how it ties the needs, requirements and functions together.  
**Response** Changes made as recommended.  
**Status** Closed

**Action Taken** Accepted  
**Comment ID** 18  
**Commenter Link** 4  
**Page** 14  
**Paragraph** 2 Overall Description  
**Comment** Expand the description to expound on the Brackets specifically indicating a user needs in Section 2. Reference Appendix A and how it ties the needs, requirements and functions together.  
**Response** Changes made as recommended. In addition, made similar changes to the beginning of Section 3 Requirements.  
**Status** Closed

**Action Taken** Accepted  
**Comment ID** 19  
**Commenter Link** 4  
**Page** 15  
**Paragraph** 2.1 Product Perspective  
**Comment** Delete "It is said that" from the start of sentence 5.  
**Response** Changes made as recommended.  
**Status** Closed

**Action Taken** Accepted  
**Comment ID** 20  
**Commenter Link** 4  
**Page** 16  
**Paragraph** 2.1 Product Perspective  
**Comment** Consistency of capitalization in list of items a) to h). E.g. universal serial bus and Portable Memory Device.  
**Response** Changes made as recommended. Capitalized Universal Serial Bus as it is used as a name of a device. Made Portable Memory Device lowercase as it is a description. An effort is being made here to capitalize terms that are typically capitalized or capitalized elsewhere in the standard.  
**Status** Closed

**Action Taken** Accepted  
**Comment ID** 24  
**Commenter Link** 4  
**Page** 27  
**Paragraph** 3.1 API Manager Requirements  
**Comment** Suggest adding / inserting information (brief synopsis) on the API Manager Requirements.  
**Response** Changes made as recommended.  
**Status** Closed

<b>Action Taken</b>	Accepted
<b>Comment ID</b>	25
<b>Commenter Link</b>	4
<b>Page</b>	27
<b>Paragraph</b>	3.1.1 Front Panel Manager Requirements
<b>Comment</b>	Suggest clarifying the term "exposed" and "focus". Maybe the use of other words such as "shown on panel display screen" could develop a faster interpretation by the reader. See also other sections in document with related terms "exposed" and "focus".
<b>Response</b>	Replaced the term "exposed" with "displayed". The term "focus" is an industry standard term and shall be remain.
<b>Status</b>	Closed
<b>Action Taken</b>	Accepted
<b>Comment ID</b>	26
<b>Commenter Link</b>	4
<b>Page</b>	28
<b>Paragraph</b>	3.1.1.1 Front Panel Manager Window
<b>Comment</b>	Figure 7 Notes Consider revising last sentence "It user settable" for clarity.
<b>Response</b>	Changes made as recommended. Removed sentence fragment.
<b>Status</b>	Closed
<b>Action Taken</b>	Accepted
<b>Comment ID</b>	29
<b>Commenter Link</b>	4
<b>Page</b>	42
<b>Paragraph</b>	3.1.2 Field I/O Manager Requirements
<b>Comment</b>	Paragraph n. Consider deleting line from underlined period "."
<b>Response</b>	Changes made as recommended.
<b>Status</b>	Closed

**Action Taken** Accepted  
**Comment ID** 30  
**Commenter Link** 4  
**Page** 49  
**Paragraph** 3.3 Performance Requirements  
**Comment** Add reference before Section 3.3.2  
**Response** Typographical mistake corrected to 3.1.2.  
**Status** Closed

**Action Taken** Accepted  
**Comment ID** 31  
**Commenter Link** 4  
**Page** 49  
**Paragraph** 3.5 Software System Attributes  
**Comment** Suggest adding / inserting information (brief synopsis) on the Software "System Attributes".  
**Response** Changes made as recommended.  
**Status** Closed

**Action Taken** Accepted  
**Comment ID** 32  
**Commenter Link** 4  
**Page**  
**Paragraph** 4.3 Utility Functions  
**Comment** replace "theneed" with "the need"  
**Response** Changes made as recommended.  
**Status** Closed

**Action Taken** Accepted

**Comment ID** 33

**Commenter Link** 4

**Page**

**Paragraph** General Comment

**Comment** Suggest Making the API Software available to the public under some for of administration / governing body. The objective is to avoid independent software development on a large scale that may cause integration conflicts, delay and high cost.

**Response** The API Software that was developed as part of the API Standard development effort is available from ITE. The purpose of this software was to help prove out the API functions. It is not complete. There has been discussion of completing the software as a reference implementation via a federally sponsored project or colaborative effort of vendors and developers but this has not been determined. Ultimately, the source of the API software will be ATC Controller vendors.

**Status** Closed

<b>Action Taken</b>	Accepted
<b>Comment ID</b>	4
<b>Commenter Link</b>	3
<b>Page</b>	67
<b>Paragraph</b>	Section 4.1, function "fpui_get_cursor_blink"
<b>Comment</b>	<p>Under the NOTES heading...</p> <p>Concerning the statement:          "This routine uses an ioctl(2) interface to gather its information.</p> <p>Is another example of the above problem. [Previous Comment]</p> <p>As with the previous proposed change. Remove this statement and similar statements from the document, since they do not convey any necessary information to the application programmer.</p> <p>The following list contains the names of the many of the functions for the front panel that contain references to system calls such as (open, close, and ioctl function call references) that are irrelevant to the actual API functions provided:</p> <p>fpui_apiver, fpui_close,          fpui_close_aux_switch, fpui_character_blink, fpui_get_cursor, fpui_get_focus,          fpui_get_led, fpui_get_reverse_video, fpui_get_underline, fpui_get_window_size, fpui_open,</p> <p>fpui_open_aux_switch, fpui_read, fpui_read_aux_switch, fpui_read_char, fpui_read_string,          fpui_refresh, fpui_set_emergency,          fpui_set_led, fpui_write_at, fpui_write_char, fpui_write_char_at, fpui_write_string,          fpui_write_string_at</p>
<b>Response</b>	Changes made as recommended.
<b>Status</b>	Closed

<b>Action Taken</b>	Accepted
<b>Comment ID</b>	5
<b>Commenter Link</b>	3
<b>Page</b>	69
<b>Paragraph</b>	Section 4.1, function "fpui_get_focus"
<b>Comment</b>	<p>Concerning the following statements: "ERRORS Refer to ioctl(2) for additional errors. NOTES This routine uses an ioctl(2) interface to gather its information."</p> <p>ioctl() can legally return nearly any error code supported by Linux. The specific errors that can be returned should be defined.</p> <p>Additionally, the reference to the ioctl system call should be removed, since it requires a specific implementation of the API, that is irrelevant to the application programmer.</p>
<b>Response</b>	Changes made as recommended.
<b>Status</b>	Closed

<b>Action Taken</b>	Accepted
<b>Comment ID</b>	3
<b>Commenter Link</b>	3
<b>Page</b>	77
<b>Paragraph</b>	Section 4.1, function "fpui_open"
<b>Comment</b>	<p>Under the NOTES heading...</p> <p>Concerning the statement:          "This routine actually wraps an open(2) call, and an ioctl(2) call."</p> <p>This statement unnecessarily causes any implementation of the API that does not call ioctl to implement the fpui_get_cursor_blink function to be non-standard compliant, even though some implementations may not find it necessary to utilize the ioctl system call to perform the required functionality of the fpui_open function.</p> <p>The proposed library functions descriptions should not specify, which system calls (ioctl, open, read, etc.) will be performed in their implementation, since such information is not needed by the user nor is it necessary to maintain binary compatibility among user applications, and additionally restricts improvements in the implementation of the shared library functions that might be made without breaking binary compatibility. As long as all the functions defined by the API are properly exported from the shared object libraries, and conform to the functionality defined in the API, the implementation of the shared objects is irrelevant to the user and not pertinent information to be included in the standard documentation.</p> <p>Completely remove the references to all system calls from the Proposed ATC API Standard, since all function calls from the application, will be made to library functions.</p>
<b>Response</b>	Changes made as recommended.
<b>Status</b>	Closed
<b>Action Taken</b>	Accepted
<b>Comment ID</b>	11
<b>Commenter Link</b>	4
<b>Page</b>	8
<b>Paragraph</b>	Table of Contents
<b>Comment</b>	Consider adding "LIST OF TABLES" and "LIST OF FIGURES"
<b>Response</b>	A lists of figures and tables are not required to "IEEE Recommended Practice for Software Requirements Specifications", IEEE Std 830-1998.
<b>Status</b>	Closed

**Action Taken** Deferred  
**Comment ID** 27  
**Commenter Link** 4  
**Page** 27  
**Paragraph** 3.1.1 Front Panel Manager  
**Comment** First Paragraph...  
General comment about capability. Is there any provision for supporting a limited display from two applications at the same time? Such that status of two different applications may be shown?  
**Response** There was not user need established to view two application windows simultaneously. This will be brought up in a future API WG Meeting.  
**Status** Closed

**Action Taken** Deferred  
**Comment ID** 1  
**Commenter Link** 1  
**Page**  
**Paragraph** 3.2.1 and others  
**Comment** I am becoming more convinced that the API should provide some utility to manage shared system settings such as date/time, IP addresses, etc. Because those settings affect all applications, they should really be managed outside of the applications. But the users still need some front panel UI to access these settings. One potential solution for this would be to create a "System Settings" utility that shows up in the front panel manager just like any other application. Or it could be more tightly integrated with the front panel manager. We should talk about this at our next meeting, whenever that might be.  
**Response** This subject was discussed extensively by the API Working Group. Although this type of utility is highly desirable, it was decided to postpone this to a subsequent version of the standard. The comment is rejected concerning this round of development but will be discussed further by the WG.  
**Status** Closed

**Action Taken** Deferred

**Comment ID** 7

**Commenter Link** 3

**Page**

**Paragraph** General Comment

**Comment** If there is any implementation using a kernel driver it would be highly useful to additionally present at least two tty character device nodes (ex. /dev/fpui1, /dev/fpui2, etc.) suitable for running a basic terminal on a virtual front panel. This addition would allow the standard input and standard output to be redirected to a virtual front panel, as is possible when communicating directly to the front panel. This also permits the use of "getty" or a similar variant thereof to launch a shell on the virtual front panel.

Add an optional additional component into the standard, which presents tty character device nodes for use by the user. Add a function in the API to see whether the implementation supports presenting tty character device nodes, and if so, provides the names of the device nodes currently present that can support virtual terminals on the front panel.

**Response** Enhancement to be discussed at a future meeting of the API WG.

**Status** Closed

**Action Taken** No Longer Applicable

**Comment ID** 6

**Commenter Link** 3

**Page**

**Paragraph** Section 3.5.1 Portability

**Comment** Concerning the section:

3.5.1 Portability

"It is not required that the implementation of this standard is portable. The API specified by this standard facilitates portability of application programs by defining a common software interface for application programs operating on ATC controller units."

It may be favorable for a manufacturer providing a board support package, to provide a user-space implementation of the API. Such implementations do not make use of specific system calls to a driver, which would allow them to maintain binary compatibility with other systems using the same processor architecture as long as they use the same C library. This is an advantage, since a kernel driver cannot be loaded into another machine (even with the same hardware) running a slightly different kernel version, or even the same kernel version that was compiled with different options (such as a different HZ value) – the entire kernel build tool-chain would be necessary to recompile the driver, just to use the API. In some cases the kernel build tool-chain may not even be provided with the board support package (though the kernel source would be), since it is not necessary for compiling applications. This would require the user to build the kernel chain from the source and that they use the same configuration options as the running kernel.

However, even with different kernel versions, it would be possible to run a user-space implementation directly by copying the binary library file to the other controller if the C library of another system's board support package is the same; and if it is not, the drivers from their BSP could be copied into a different user space file-system that uses the same C library, and the binary version of the API would still be able to work on the other hardware (given similar processor architectures).

Kernel functions for using the synchronous serial ports, necessary for a complete kernel implementation of the field i/o manager should not be a requirement of this standard, since it undermines the ATC Standard, imposing further requirements for the synchronous serial driver.

Such a user-space library can be written (using system V shared memory and semaphores) to be comparably as efficient as a kernel implementation, and often even more efficient if the kernel implementation also has a user-space server counterpart to service requests, as is the case with a current, still incomplete, implementation of the API.

In many cases having a user-space implementation that only uses drivers specified in the ATC 5.2 Standard circumvents the necessity of recompiling the API library and/or needing the kernel tool-chain to recompile a supporting kernel driver counterpart to the API (i.e. this removes the necessity to change the approved ATC standard and make Linux Kernel calls available for some BSP functions such as the SDLC drivers).

**Response** This comment speaks to the merits of the approach of the software implemented as part of the development of the standard. As the commenter has noted in previous remarks, the underlying architecture will not be dictated in the standard.

It was the intent of the Working Group for API libraries to be built in the driver fashion as indicated by the group's approval of the design approach by the WG contractor. This was



done for consistency, reliability and interoperability of implementations.

<b>Status</b>	Closed
<b>Action Taken</b>	Rejected
<b>Comment ID</b>	21
<b>Commenter Link</b>	4
<b>Page</b>	18
<b>Paragraph</b>	2.1 Product Perspective
<b>Comment</b>	Consider referencing ATC Controller Standard Version as the section number may change in future versions.
<b>Response</b>	The referenced version of the ATC Controller Standard is listed in the Section 1.4 References section of the document. An additional sentence was added to refer the reader to it.
<b>Status</b>	Closed
<b>Action Taken</b>	Rejected
<b>Comment ID</b>	22
<b>Commenter Link</b>	4
<b>Page</b>	20
<b>Paragraph</b>	2.1.5 Communications Interfaces
<b>Comment</b>	Consider referencing ATC Controller Standard Version as the section number may change in future versions. This would affect other places in the document where a particular section of the ATC Standard is referenced.
<b>Response</b>	The referenced version of the ATC Controller Standard is listed in the Section 1.4 References section of the document. An additional sentence was added to Section 2.1 to refer the reader to it.
<b>Status</b>	Closed

**Action Taken** Rejected  
**Comment ID** 23  
**Commenter Link** 4  
**Page** 22  
**Paragraph** 2.2 Product Features  
**Comment** Consider referencing ATC Controller Standard Version as the section number may change in future versions. This would affect other places in the document where a particular section of the ATC Standard is referenced.  
**Response** The referenced version of the ATC Controller Standard is listed in the Section 1.4 References section of the document. An additional sentence was added to Section 2.1 to refer the reader to it.  
**Status** Closed

**Action Taken** Rejected  
**Comment ID** 28  
**Commenter Link** 4  
**Page** 40  
**Paragraph** 3.1.2 Field I/O Manager Requirements  
**Comment** "The API shall use this mapping to set the contents of FIOMMU Frame 0 and FIOCMU Frames 61 and 67." Does this need a reference to another document?  
**Response** The message frames and their definitions are defined in different ATC Standards as described in the third paragraph of Section 3.1.2.  
**Status** Closed

**Action Taken** Rejected

**Comment ID** 8

**Commenter Link** 4

**Page**

**Paragraph** General Comment

**Comment** All of the other ATC documents provide a glossary in the back of the document and refer to Annex as opposed to Appendix. Is there a particular reason for structuring this document with the definitions and abbreviations in the front and referring to the Added items at the end as Appendix instead of Annex?

**Response** The API Standard is modeled after the following: "IEEE Recommended Practice for Software Requirements Specifications", IEEE Std 830-1998.

**Status** Closed

  

**Action Taken** Rejected

**Comment ID** 9

**Commenter Link** 4

**Page**

**Paragraph** General Comment

**Comment** Where possible, consider adding a brief synopsis on referenced subject / term instead of having the reader totally relying on other documents for clarification/explanation.

**Response** Without specific references it is difficult to understand what the concern is. Section 2.1 gives background information on the ATC and API standards. Where specific information is needed, references must be given to assure accuracy.

**Status** Closed

  

**Action Taken** Rejected

**Comment ID** 10

**Commenter Link** 4

**Page** 8

**Paragraph** Table of Contents

**Comment** Consider changing "APPENDIXES" to "APPENDICES"

**Response** "Appendixes" is proper according to "IEEE Recommended Practice for Software Requirements Specifications", IEEE Std 830-1998.

**Status** Closed