LynxOS-178 FACE Conformance Document

LynxOS-178

DOC-2207-00

Printed in the United States of America.

Lynx Software Technologies, Inc. makes no representations, express or implied, with respect to this documentation or the software it describes, including (with no limitation) any implied warranties of utility or fitness for any particular purpose; all such warranties are expressly disclaimed. Neither Lynx Software Technologies, Inc., nor its distributors, nor its dealers shall be liable for any indirect, incidental, or consequential damages under any circumstances.

(The exclusion of implied warranties may not apply in all cases under some statutes, and thus the above exclusion may not apply. This warranty provides the purchaser with specific legal rights. There may be other purchaser rights which vary from state to state within the United States of America.)

Preface

Typographical Conventions

The typefaces used in this manual, summarized below, emphasize important concepts. All references to filenames and commands are case-sensitive and should be typed accurately.

Kind of Text	**Examples**
Body text; *italicized* for emphasis, new terms, and book titles	Refer to the *LynxOS-178 FACE Conformance Document*
Environment variables, filenames, functions, methods, options, parameter names, path names, commands, and computer data	`ls -l myprog.c /dev/null`

Commands that need to be highlighted within body text, or commands that must be typed as is by the user are **`bolded`**.	`login:` **`myname`**
	`#` **`cd /usr/home`**

Text that represents a variable, such as a filename or a value that must be entered by the user, is *italicized*.	`cat` *`<filename>`*
	`mv` *`<file1>`* *`<file2>`*

Blocks of text that appear on the display screen after entering instructions or commands	`Loading file /tftpboot/shell.kdi`
	`into 0x4000`
	`....................................`
	`.`
	`File loaded. Size is 1314816`
	`© 2015 Lynx Software Technologies,`
	`Inc. All rights reserved.`

Keyboard options, button names, and menu sequences	**`Enter, Ctrl-C`**

Technical Support

Lynx Software Technologies handles support requests from current support subscribers. For questions regarding Lynx Software Technologies products, evaluation CDs, or to become a support subscriber; our knowledgeable sales staff will be pleased to help you. Please visit us at:
`http://www.lynx.com/training-support/contact-support/`

How to Submit a Support Request

When you are ready to submit a support request, please include *all* of the following information:

- First name, last name, your job title
- Phone number, e-mail address
- Company name, address
- Product version number
- Target platform (for example, PowerPC)
- Board Support Package (BSP), Current Service Pack Revision, Development Host OS version
- Detailed description of the problem that you are experiencing:
- Is there a requirement for a US Citizen or Green Card holder to work on this issue?
- Priority of the problem - Critical, High, Medium, or Low?

Where to Submit a Support Request

Support, Europe	tech_europe@lynx.com +33 1 30 85 93 96
Support, worldwide except Europe	support@lynx.com +1 800-327-5969 or +1 408-979-3940 +81 33 449 3131 [for Japan]
Training and Courses	USA: training-usa@lynx.com Europe: training-europe@lynx.com USA: +1 408-979-4353 Europe: +33 1 30 85 06 00

CHAPTER 1

Introduction

Overview

The Future Airborne Capability Environment (FACE) Technical Standard is defined by The Open Group FACE Consortium to aid software robustness, interoperability, portability, safety and security. FACE specifies an open reference architecture encompassing functionally diverse software components potentially written in different programming languages interconnected with standardized interfaces. FACE also specifies a conformance verification process to test candidate software products as Units of Conformance (UoC) constituting a particular component of the FACE reference architecture. The FACE Consortium develops and provides a Conformance Test Suite and a Conformance Verification Matrix to help facilitate this task.

This document records the conformance of the LynxOS-178 operating system to the FACE Technical Standard, Edition 2.1, published by The Open Group in May 2014.

LynxOS-178 bundles multiple components such as the operating system kernel, C language run-time and standard library, ARINC 653 support driver and library, etc. As a candidate UoC, all of LynxOS-178's components reside and execute within the Operating System Segment (OSS) of the FACE reference architecture.

LynxOS-178 targets the API feature set of the Safety Extended profile of the FACE OSS for user applications developed in the C language.

Unless stated otherwise below, this document describes API feature set inclusion and behavior as it applies to LynxOS-178 production mode only.

FACE Conformance Test Suite

The FACE Conformance Test Suite (CTS) is developed by the FACE Consortium to help facilitate the task of testing an operating system for FACE conformance. The CTS tests the presence of the API feature set (types, constants and functions) required by the corresponding FACE segment, profile and components targeted by a candidate UoC.

Conformance Test Environment

The conformance test environment consists of the following:

- A development host running CentOS Linux 7.2 or Windows 7 with Cygwin
- LynxOS-178 build environment with cross-developmenttoolchain
- Python 2.6+ (but not 3.x)
- Java 1.7+
- JRE
- TkInter
- The FACE Conformance Test Suite 2.1.0r5

A LynxOS-178 production mode build environment is installed with the cross-development toolchain prior to running the CTS. Then, the test suite is configured and executed as per the procedure described in the official FACE CTS documentation.

For LynxOS-178, the CTS is configured as follows:

• Language:	C
• Segment:	OSS
• Profile:	Safety Extended
• Partitions:	POSIX + ARINC 653
• Components:	C Standard Library, POSIX, ARINC 653, HMFM

Additionally, the CTS is modified as follows:

- All test cases for the ARINC 653 component of the test suite are updated to rename the `main()` function as `arinc653_main()`. This is needed because the LynxOS-178 ARINC 653 support library implementation provides the `main()` function for C programs by itself and requires the user application to provide the `arinc653_main()` function instead. This is a design element of the LynxOS-178 ARINC 653implementation.
- A common header for all test cases for the HMFM component of the test suite is updated to define macros prefixing FACE_ or FACE_HMFM_ to the type, constant and function names used or provided by the FACE HMFM API. This is needed because the LynxOS-178 FACE HMFM implementation does not define namespace-prefixed names as specified

in Sec. 3.6.4.1.2 of the FACE 2.1 Technical Standard. This is a FACE non-conformance issue in the LynxOS-178 FACE HMFM implementation.

Conformance Test Results

The conformance test results summary for all test cases across API components is summarized below:

Component	Test Cases	PASS	FAIL
C Standard Library	103	103	0
POSIX	335	326	9
ARINC 653	81	81	0
HMFM	21	10	11
Total	**540**	**520**	**20**

These test results are analyzed in detail in Chapter 2.

FACE Conformance Verification Matrix

The FACE Conformance Verification Matrix (CVM) is provided by the FACE Consortium to help facilitate the task of verifying an operating system for FACE conformance. The CVM is a spreadsheet containing the text of the FACE Technical Standard as line items corresponding to requirements in separate rows, along with columns specifying how conformance to that requirement is intended to be verified in the candidate UoC. The results obtained by running the FACE Conformance Test Suite are also part of the verification process laid out in the CVM. The *LynxOS-178 FACE Conformance Verification Matrix* extends the spreadsheet with three additional columns elaborating (a) the status of support of the corresponding line item requirement, (b) the evidence in the form of product artifacts proving the said conformance, and (c) any extra comments as appropriate.

CVM Requirements

The CVM spreadsheet provided by the FACE Consortium is filtered to include only those rows that specify OSS (either by itself or in conjunction with other segments) in the “FACE Segment” column, and C Language Safety Extended Profile, Configuration, Direct Network Usage, HMFM, Language Run-time, Network, Partitioning, Provides HMFM, Safety Profile, Safety Profile Security Profile Partitioning, Safety Profile Partitioning, Safety Profile Security Profile, Uses Centralized Configuration or nothing in the “Conditional Requirements” column.

This exercise leaves 65 rows (line items corresponding to FACE Technical Standard requirements) in the CVM to be verified to determine the conformance of LynxOS-178 to the FACE OSS Safety Extended profile. Of these, 48 items are intended to be verified by manual inspection of the product artifacts, 14 items are intended to be verified by executing the CTS and 3 items are intended to be verified by manual inspection of the product artifacts as well as by executing the CTS.

CVM Summary

The verification status of matrix requirements for LynxOS-178 is summarized below:

Verification Status	Number of CVM Items
Fully Supported Requirements	44
Partially Supported Requirements	16
Unsupported Requirements	1
Inapplicable Requirements	4
Total	**65**

These requirements are analyzed in detail in Chapter 3.

CHAPTER 2 *Analysis of Test Results*

Passed Test Cases

As described in Chapter 1, the FACE Conformance Test Suite configured for LynxOS-178 runs a total of 540 test cases across API components. Other than those described in the next section, all test cases across all API components are successfully passed. The total number of PASS test cases is 520.

Failed Test Cases

This section provides analyses for all failed test cases. The total number of FAIL test cases is 20. Of these, 9 test cases belong to the POSIX API component and 11 test cases belong to the HMFM API component.

Component: POSIX Conformance Tests

The 9 failed test cases below test the existence of the functions `posix_spawn()`, `posix_spawnattr_destroy()`, `posix_spawnattr_getflags()`, `posix_spawnattr_getsigdefault()`, `posix_spawnattr_getsigmask()`, `posix_spawnattr_init()`, `posix_spawnattr_setflags()`, `posix_spawnattr_setsigdefault()` and `posix_spawnattr_setsigmask()` exported by the `spawn.h` header of the POSIX API.

Test for `posix_spawn()` from `spawn.h`

Testing definition of `posix_spawn` → FAILED

Test Code:

```
/*************************************************************************************
 * DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. *
 *                                                                                   *
 * Product produced under DoD SENSIAC contract HC104705D4000 under the sponsorship   *
 * of the Defense Technical Information Center,                                      *
 * ATTN: DTIC-AI, 8723 John J. Kingman Rd., Ste 0944, Fort Belvoir, VA 22060-6218.   *
 * SENSIAC is a DoD Information Analysis Center Sponsored by the Defense Technical   *
 * Information Center.                                                               *
 *                                                                                   *
 * HANDLING AND DESTRUCTION NOTICE - Comply with distribution statement and destroy *
 * by any method that will prevent disclosure of the contents or reconstruction of    *
 * the document.                                                                      *
 *                                                                                    *
 * Copyright (c) Vanderbilt University, 2016                                          *
 *                                                                                    *
 * ALL RIGHTS RESERVED, UNLESS OTHERWISE STATED                                       *
 *                                                                                    *
 * Developed under contract for the U.S. Department of Defense                        *
 * by the Institute for Software Integrated Systems (ISIS), Vanderbilt University     *
 *                                                                                    *
 * The FACE Consortium has approved results from this release for                     *
 * certifying software components as FACE Conformant                                 *
 *                                                                                    *
 * Vanderbilt University disclaims all warranties with regard to this software,       *
 * including all implied warranties of merchantability and fitness. In no event      *
 * shall Vanderbilt University be liable for any special, indirect or                 *
 * consequential damages or any damages whatsoever resulting from loss of use,        *
 * data or profits, whether in an action of contract, negligence or other tortious    *
 * action, arising out of or in connection with the use or performance of this        *
 * software.                                                                          *
 *                                                                                    *
 * Conformance Test Suite Version 2.1.0r5 supporting FACE Standard Version 2.1        *
 *                                                                                    *
 * This file is generated and may be overwritten.                                     *
 *                                                                                    *
 * Release Date: 18-May-2017                                                          *
 ************************************************************************************/

#include <spawn.h>

/* restrict is not a keyword in C++ so define it away. */
#ifdef___cplusplus
    #ifndef restrict
        #define restrict
    #endif
#endif

int main()
{
    pid_t *restrict* var0 = {0};
    const char *restrict* var1 = {0};
    const posix_spawn_file_actions_t ** var2 = {0};
    const posix_spawnattr_t *restrict* var3 = {0};
    char *const* var4[] = {0};
    char *const* var5[] = {0};

    posix_spawn(*var0, *var1, *var2, *var3, *var4, *var5);

    return 0;
}
```

Test Log File:

```
command:
make safetyExt/spawn.h/Pass/Test000001
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o safetyExt/spawn.h/Pass/
Test000001.o -c safetyExt/spawn.h/Pass/Test000001.c
gcc -o safetyExt/spawn.h/Pass/Test000001 safetyExt/spawn.h/Pass/Test000001.o
/usr/src/lynx/los178/lib/libc.a /usr/src/lynx/los178/lib/libm.a
safetyExt/spawn.h/Pass/Test000001.o: In function `main':
Test000001.c:(.text+0x94): undefined reference to `posix_spawn'
collect2: ld returned 1 exit status
make: *** [safetyExt/spawn.h/Pass/Test000001] Error 1
```

Test for posix_spawnattr_destroy() from spawn.h

Testing definition of posix_spawnattr_destroy → FAILED

Test Code:

```
/***********************************************************************************
 * DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. *
 *                                                                                 *
 * Product produced under DoD SENSIAC contract HC104705D4000 under the sponsorship    *
 * of the Defense Technical Information Center,                                    *
 * ATTN: DTIC-AI, 8723 John J. Kingman Rd., Ste 0944, Fort Belvoir, VA 22060-6218.    *
 * SENSIAC is a DoD Information Analysis Center Sponsored by the Defense Technical    *
 * Information Center.                                                             *
 *                                                                                 *
 * HANDLING AND DESTRUCTION NOTICE - Comply with distribution statement and destroy *
 * by any method that will prevent disclosure of the contents or reconstruction of    *
 * the document.                                                                   *
 *                                                                                 *
 * Copyright (c) Vanderbilt University, 2016                                       *
 *                                                                                 *
 * ALL RIGHTS RESERVED, UNLESS OTHERWISE STATED                                    *
 *                                                                                 *
 * Developed under contract for the U.S. Department of Defense                     *
 * by the Institute for Software Integrated Systems (ISIS), Vanderbilt University     *
 *                                                                                 *
 * The FACE Consortium has approved results from this release for                  *
 * certifying software components as FACE Conformant                               *
 *                                                                                 *
 * Vanderbilt University disclaims all warranties with regard to this software,       *
 * including all implied warranties of merchantability and fitness. In no event      *
 * shall Vanderbilt University be liable for any special, indirect or              *
 * consequential damages or any damages whatsoever resulting from loss of use,       *
 * data or profits, whether in an action of contract, negligence or other tortious    *
 * action, arising out of or in connection with the use or performance of this       *
 * software.                                                                       *
 *                                                                                 *
 * Conformance Test Suite Version 2.1.0r5 supporting FACE Standard Version 2.1       *
 *                                                                                 *
 * This file is generated and may be overwritten.                                  *
 *                                                                                 *
 * Release Date: 18-May-2017                                                       *
 ***********************************************************************************/

#include <spawn.h>

int main()
{
    posix_spawnattr_t ** var18 = {0};

    posix_spawnattr_destroy(*var18);

    return 0;
}
```

Test Log File:

```
command:
make safetyExt/spawn.h/Pass/Test000002
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o safetyExt/spawn.h/Pass/
```

```
Test000002.o -c safetyExt/spawn.h/Pass/Test000002.c
gcc -o safetyExt/spawn.h/Pass/Test000002 safetyExt/spawn.h/Pass/Test000002.o
/usr/src/lynx/los178/lib/libc.a /usr/src/lynx/los178/lib/libm.a
safetyExt/spawn.h/Pass/Test000002.o: In function `main':
Test000002.c:(.text+0x2c): undefined reference to `posix_spawnattr_destroy'
collect2: ld returned 1 exit status
make: *** [safetyExt/spawn.h/Pass/Test000002] Error 1
```

Test for posix_spawnattr_getflags() from spawn.h

Testing definition of posix_spawnattr_getflags → FAILED

Test Code:

```
/**************************************************************************************
 * DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. *
 *                                                                                    *
 * Product produced under DoD SENSIAC contract HC104705D4000 under the sponsorship    *
 * of the Defense Technical Information Center,                                       *
 * ATTN: DTIC-AI, 8723 John J. Kingman Rd., Ste 0944, Fort Belvoir, VA 22060-6218.    *
 * SENSIAC is a DoD Information Analysis Center Sponsored by the Defense Technical    *
 * Information Center.                                                                *
 *                                                                                    *
 * HANDLING AND DESTRUCTION NOTICE - Comply with distribution statement and destroy *
 * by any method that will prevent disclosure of the contents or reconstruction of    *
 * the document.                                                                      *
 *                                                                                    *
 * Copyright (c) Vanderbilt University, 2016                                          *
 *                                                                                    *
 * ALL RIGHTS RESERVED, UNLESS OTHERWISE STATED                                       *
 *                                                                                    *
 * Developed under contract for the U.S. Department of Defense                        *
 * by the Institute for Software Integrated Systems (ISIS), Vanderbilt University     *
 *                                                                                    *
 * The FACE Consortium has approved results from this release for                     *
 * certifying software components as FACE Conformant                                  *
 *                                                                                    *
 * Vanderbilt University disclaims all warranties with regard to this software,       *
 * including all implied warranties of merchantability and fitness. In no event       *
 * shall Vanderbilt University be liable for any special, indirect or                 *
 * consequential damages or any damages whatsoever resulting from loss of use,        *
 * data or profits, whether in an action of contract, negligence or other tortious    *
 * action, arising out of or in connection with the use or performance of this        *
 * software.                                                                          *
 *                                                                                    *
 * Conformance Test Suite Version 2.1.0r5 supporting FACE Standard Version 2.1        *
 *                                                                                    *
 * This file is generated and may be overwritten.                                     *
 *                                                                                    *
 * Release Date: 18-May-2017                                                          *
 *************************************************************************************/

#include <spawn.h>

/* restrict is not a keyword in C++ so define it away. */
#ifdef___cplusplus
    #ifndef restrict
        #define restrict
    #endif
#endif

int main()
{
    const posix_spawnattr_t *restrict* var19 = {0};
    short *restrict* var20 = {0};
    posix_spawnattr_getflags(*var19, *var20);

    return 0;
}
```

Test Log File:

```
command:
make safetyExt/spawn.h/Pass/Test000003
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o safetyExt/spawn.h/Pass/
Test000003.o -c safetyExt/spawn.h/Pass/Test000003.c
gcc -o safetyExt/spawn.h/Pass/Test000003 safetyExt/spawn.h/Pass/Test000003.o
/usr/src/lynx/los178/lib/libc.a /usr/src/lynx/los178/lib/libm.a
safetyExt/spawn.h/Pass/Test000003.o: In function `main':
Test000003.c:(.text+0x44): undefined reference to `posix_spawnattr_getflags'
collect2: ld returned 1 exit status
make: *** [safetyExt/spawn.h/Pass/Test000003] Error 1
```

Test for `posix_spawnattr_getsigdefault()` from `spawn.h`

Testing definition of `posix_spawnattr_getsigdefault` → FAILED

Test Code:

```
/**************************************************************************************
 * DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. *
 *                                                                                   *
 * Product produced under DoD SENSIAC contract HC104705D4000 under the sponsorship    *
 * of the Defense Technical Information Center,                                       *
 * ATTN: DTIC-AI, 8723 John J. Kingman Rd., Ste 0944, Fort Belvoir, VA 22060-6218.    *
 * SENSIAC is a DoD Information Analysis Center Sponsored by the Defense Technical    *
 * Information Center.                                                                *
 *                                                                                   *
 * HANDLING AND DESTRUCTION NOTICE - Comply with distribution statement and destroy *
 * by any method that will prevent disclosure of the contents or reconstruction of    *
 * the document.                                                                      *
 *                                                                                   *
 * Copyright (c) Vanderbilt University, 2016                                          *
 *                                                                                   *
 * ALL RIGHTS RESERVED, UNLESS OTHERWISE STATED                                       *
 *                                                                                   *
 * Developed under contract for the U.S. Department of Defense                        *
 * by the Institute for Software Integrated Systems (ISIS), Vanderbilt University     *
 *                                                                                   *
 * The FACE Consortium has approved results from this release for                     *
 * certifying software components as FACE Conformant                                  *
 *                                                                                   *
 * Vanderbilt University disclaims all warranties with regard to this software,       *
 * including all implied warranties of merchantability and fitness. In no event       *
 * shall Vanderbilt University be liable for any special, indirect or                 *
 * consequential damages or any damages whatsoever resulting from loss of use,        *
 * data or profits, whether in an action of contract, negligence or other tortious    *
 * action, arising out of or in connection with the use or performance of this        *
 * software.                                                                          *
 *                                                                                   *
 * Conformance Test Suite Version 2.1.0r5 supporting FACE Standard Version 2.1        *
 *                                                                                   *
 * This file is generated and may be overwritten.                                    *
 *                                                                                   *
 * Release Date: 18-May-2017                                                         *
 ************************************************************************************/

#include <spawn.h>

/* restrict is not a keyword in C++ so define it away. */
#ifdef___cplusplus
    #ifndef restrict
        #define restrict
    #endif
#endif

int main()
{
    const posix_spawnattr_t *restrict* var27 = {0};
    sigset_t *restrict* var28 = {0};

    posix_spawnattr_getsigdefault(*var27, *var28);

    return 0;
}
```

Test Log File:

```
command:
make safetyExt/spawn.h/Pass/Test000004
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o safetyExt/spawn.h/Pass/
Test000004.o -c safetyExt/spawn.h/Pass/Test000004.c
gcc -o safetyExt/spawn.h/Pass/Test000004 safetyExt/spawn.h/Pass/Test000004.o
/usr/src/lynx/los178/lib/libc.a /usr/src/lynx/los178/lib/libm.a
safetyExt/spawn.h/Pass/Test000004.o: In function `main':
Test000004.c:(.text+0x44): undefined reference to `posix_spawnattr_getsigdefault'
collect2: ld returned 1 exit status
make: *** [safetyExt/spawn.h/Pass/Test000004] Error 1
```

Test for `posix_spawnattr_getsigmask()` from `spawn.h`

Testing definition of `posix_spawnattr_getsigmask` → FAILED

Test Code:

```
/************************************************************************************
 * DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. *
 *                                                                                   *
 * Product produced under DoD SENSIAC contract HC104705D4000 under the sponsorship   *
 * of the Defense Technical Information Center,                                      *
 * ATTN: DTIC-AI, 8723 John J. Kingman Rd., Ste 0944, Fort Belvoir, VA 22060-6218.   *
 * SENSIAC is a DoD Information Analysis Center Sponsored by the Defense Technical   *
 * Information Center.                                                               *
 *                                                                                   *
 * HANDLING AND DESTRUCTION NOTICE - Comply with distribution statement and destroy *
 * by any method that will prevent disclosure of the contents or reconstruction of   *
 * the document.                                                                     *
 *                                                                                   *
 * Copyright (c) Vanderbilt University, 2016                                         *
 *                                                                                 *
 * ALL RIGHTS RESERVED, UNLESS OTHERWISE STATED                                    *
 *                                                                                 *
 * Developed under contract for the U.S. Department of Defense                     *
 * by the Institute for Software Integrated Systems (ISIS), Vanderbilt University  *
 *                                                                                 *
 * The FACE Consortium has approved results from this release for                  *
 * certifying software components as FACE Conformant                               *
 *                                                                                 *
 * Vanderbilt University disclaims all warranties with regard to this software,    *
 * including all implied warranties of merchantability and fitness. In no event    *
 * shall Vanderbilt University be liable for any special, indirect or              *
 * consequential damages or any damages whatsoever resulting from loss of use,     *
 * data or profits, whether in an action of contract, negligence or other tortious *
 * action, arising out of or in connection with the use or performance of this     *
 * software.                                                                       *
 *                                                                                 *
 * Conformance Test Suite Version 2.1.0r5 supporting FACE Standard Version 2.1     *
 *                                                                                 *
 * This file is generated and may be overwritten.                                  *
 *                                                                                 *
 * Release Date: 18-May-2017                                                       *
 ***********************************************************************************/

#include <spawn.h>

/* restrict is not a keyword in C++ so define it away. */
#ifdef___cplusplus
    #ifndef restrict
        #define restrict
    #endif
#endif

int main()
{
    const posix_spawnattr_t *restrict* var29 = {0};
    sigset_t *restrict* var30 = {0};

    posix_spawnattr_getsigmask(*var29, *var30);

    return 0;
}
```

Test Log File:

```
command:
make safetyExt/spawn.h/Pass/Test000005
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o safetyExt/spawn.h/Pass/
Test000005.o -c safetyExt/spawn.h/Pass/Test000005.c
gcc -o safetyExt/spawn.h/Pass/Test000005 safetyExt/spawn.h/Pass/Test000005.o
/usr/src/lynx/los178/lib/libc.a /usr/src/lynx/los178/lib/libm.a
safetyExt/spawn.h/Pass/Test000005.o: In function `main':
Test000005.c:(.text+0x44): undefined reference to `posix_spawnattr_getsigmask'
collect2: ld returned 1 exit status
make: *** [safetyExt/spawn.h/Pass/Test000005] Error 1
```

Test for `posix_spawnattr_init()` from `spawn.h`

Testing definition of `posix_spawnattr_init` → FAILED

Test Code:

```
/**************************************************************************************
 * DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. *
 *                                                                                    *
 * Product produced under DoD SENSIAC contract HC104705D4000 under the sponsorship    *
 * of the Defense Technical Information Center,                                       *
 * ATTN: DTIC-AI, 8723 John J. Kingman Rd., Ste 0944, Fort Belvoir, VA 22060-6218.    *
 * SENSIAC is a DoD Information Analysis Center Sponsored by the Defense Technical    *
 * Information Center.                                                                *
 *                                                                                    *
 * HANDLING AND DESTRUCTION NOTICE - Comply with distribution statement and destroy *
 * by any method that will prevent disclosure of the contents or reconstruction of    *
 * the document.                                                                      *
 *                                                                                    *
 * Copyright (c) Vanderbilt University, 2016                                          *
 *                                                                                    *
 * ALL RIGHTS RESERVED, UNLESS OTHERWISE STATED                                       *
 *                                                                                    *
 * Developed under contract for the U.S. Department of Defense                        *
 * by the Institute for Software Integrated Systems (ISIS), Vanderbilt University     *
 *                                                                                    *
 * The FACE Consortium has approved results from this release for                     *
 * certifying software components as FACE Conformant                                  *
 *                                                                                    *
 * Vanderbilt University disclaims all warranties with regard to this software,       *
 * including all implied warranties of merchantability and fitness. In  no event      *
 * shall Vanderbilt University be liable for any special, indirect or                 *
 * consequential damages or any damages whatsoever resulting from loss of use,        *
 * data or profits, whether in an action of contract, negligence or other tortious    *
 * action, arising out of or in connection with the use or performance of this        *
 * software.                                                                          *
 *                                                                                    *
 * Conformance Test Suite Version 2.1.0r5 supporting FACE Standard Version 2.1        *
 *                                                                                    *
 * This file is generated and may be overwritten.                                     *
 *                                                                                    *
 * Release Date: 18-May-2017                                                          *
 **************************************************************************************/

#include <spawn.h>

int main()
{
    posix_spawnattr_t ** var31 = {0};

    posix_spawnattr_init(*var31);

    return 0;
}
```

Test Log File:

```
command:
make safetyExt/spawn.h/Pass/Test000006
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/home/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o safetyExt/spawn.h/Pass/
```

```
Test000006.o -c safetyExt/spawn.h/Pass/Test000006.c
gcc -o safetyExt/spawn.h/Pass/Test000006 safetyExt/spawn.h/Pass/Test000006.o     /
/usr/src/lynx/los178/lib/libc.a /usr/src/lynx/los178/lib/libm.a
safetyExt/spawn.h/Pass/Test000006.o: In function `main':
Test000006.c:(.text+0x2c): undefined reference to `posix_spawnattr_init'
collect2: ld returned 1 exit status
make: *** [safetyExt/spawn.h/Pass/Test000006] Error 1
```

Test for posix_spawnattr_setflags() from spawn.h

Testing definition of posix_spawnattr_setflags → FAILED

Test Code:

```
/*******************************************************************************
 * DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. *
 *                                                                             *
 * Product produced under DoD SENSIAC contract HC104705D4000 under the sponsorship    *
 * of the Defense Technical Information Center,                                *
 * ATTN: DTIC-AI, 8723 John J. Kingman Rd., Ste 0944, Fort Belvoir, VA 22060-6218.    *
 * SENSIAC is a DoD Information Analysis Center Sponsored by the Defense Technical    *
 * Information Center.                                                         *
 *                                                                             *
 * HANDLING AND DESTRUCTION NOTICE - Comply with distribution statement and destroy *
 * by any method that will prevent disclosure of the contents or reconstruction of    *
 * the document.                                                               *
 *                                                                             *
 * Copyright (c) Vanderbilt University, 2016                                   *
 *                                                                             *
 * ALL RIGHTS RESERVED, UNLESS OTHERWISE STATED                                *
 *                                                                             *
 * Developed under contract for the U.S. Department of Defense                 *
 * by the Institute for Software Integrated Systems (ISIS), Vanderbilt University     *
 *                                                                             *
 * The FACE Consortium has approved results from this release for              *
 * certifying software components as FACE Conformant                           *
 *                                                                             *
 * Vanderbilt University disclaims all warranties with regard to this software,       *
 * including all implied warranties of merchantability and fitness. In no event       *
 * shall Vanderbilt University be liable for any special, indirect or          *
 * consequential damages or any damages whatsoever resulting from loss of use,        *
 * data or profits, whether in an action of contract, negligence or other tortious    *
 * action, arising out of or in connection with the use or performance of this        *
 * software.                                                                   *
 *                                                                             *
 * Conformance Test Suite Version 2.1.0r5 supporting FACE Standard Version 2.1        *
 *                                                                             *
 * This file is generated and may be overwritten.                              *
 *                                                                             *
 * Release Date: 18-May-2017                                                   *
 *******************************************************************************/

#include <spawn.h>

int main()
{
    posix_spawnattr_t ** var32 = {0};
    short* var33 = {0};

    posix_spawnattr_setflags(*var32, *var33);

    return 0;
}
```

Test Log File:

```
command:
make safetyExt/spawn.h/Pass/Test000007
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o safetyExt/spawn.h/Pass/
Test000007.o -c safetyExt/spawn.h/Pass/Test000007.c
gcc -o safetyExt/spawn.h/Pass/Test000007 safetyExt/spawn.h/Pass/Test000007.o
/usr/src/lynx/los178/lib/libc.a /usr/src/lynx/los178/lib/libm.a
safetyExt/spawn.h/Pass/Test000007.o: In function `main':
Test000007.c:(.text+0x48): undefined reference to `posix_spawnattr_setflags'
collect2: ld returned 1 exit status
make: *** [safetyExt/spawn.h/Pass/Test000007] Error 1
```

Test for `posix_spawnattr_setsigdefault()` from `spawn.h`

Testing definition of `posix_spawnattr_setsigdefault` → FAILED

Test Code:

```
/*************************************************************************************
 * DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. *
 *                                                                                   *
 * Product produced under DoD SENSIAC contract HC104705D4000 under the sponsorship   *
 * of the Defense Technical Information Center,                                      *
 * ATTN: DTIC-AI, 8723 John J. Kingman Rd., Ste 0944, Fort Belvoir, VA 22060-6218.   *
 * SENSIAC is a DoD Information Analysis Center Sponsored by the Defense Technical   *
 * Information Center.                                                               *
 *                                                                                   *
 * HANDLING AND DESTRUCTION NOTICE - Comply with distribution statement and destroy *
 * by any method that will prevent disclosure of the contents or reconstruction of   *
 * the document.                                                                     *
 *                                                                                   *
 * Copyright (c) Vanderbilt University, 2016                                         *
 *                                                                                   *
 * ALL RIGHTS RESERVED, UNLESS OTHERWISE STATED                                      *
 *                                                                                   *
 * Developed under contract for the U.S. Department of Defense                       *
 * by the Institute for Software Integrated Systems (ISIS), Vanderbilt University    *
 *                                                                                   *
 * The FACE Consortium has approved results from this release for                    *
 * certifying software components as FACE Conformant                                 *
 *                                                                                   *
 * Vanderbilt University disclaims all warranties with regard to this software,      *
 * including all implied warranties of merchantability and fitness. In no event      *
 * shall Vanderbilt University be liable for any special, indirect or                *
 * consequential damages or any damages whatsoever resulting from loss of use,       *
 * data or profits, whether in an action of contract, negligence or other tortious   *
 * action, arising out of or in connection with the use or performance of this       *
 * software.                                                                         *
 *                                                                                   *
 * Conformance Test Suite Version 2.1.0r5 supporting FACE Standard Version 2.1       *
 *                                                                                   *
 * This file is generated and may be overwritten.                                    *
 *                                                                                   *
 * Release Date: 18-May-2017                                                         *
 ************************************************************************************/
```

```
#include <spawn.h>

/* restrict is not a keyword in C++ so define it away. */
#ifdef __cplusplus
    #ifndef restrict
        #define restrict
    #endif
#endif

int main()
{
    posix_spawnattr_t *restrict* var40 = {0};
    const sigset_t *restrict* var41 = {0};

    posix_spawnattr_setsigdefault(*var40, *var41);

    return 0;
}
```

Test Log File:

```
command:
make safetyExt/spawn.h/Pass/Test000008
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions   -o safetyExt/spawn.h/Pass/
Test000008.o -c safetyExt/spawn.h/Pass/Test000008.c
gcc -o safetyExt/spawn.h/Pass/Test000008 safetyExt/spawn.h/Pass/Test000008.o
/usr/src/lynx/los178/lib/libc.a /usr/src/lynx/los178/lib/libm.a
safetyExt/spawn.h/Pass/Test000008.o: In function `main':
Test000008.c:(.text+0x44): undefined reference to `posix_spawnattr_setsigdefault'
collect2: ld returned 1 exit status
make: *** [safetyExt/spawn.h/Pass/Test000008] Error 1
```

Test for `posix_spawnattr_setsigmask()` from `spawn.h`

Testing definition of `posix_spawnattr_setsigmask` → FAILED

Test Code:

```
/***************************************************************************************
 * DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. *
 *                                                                                    *
 * Product produced under DoD SENSIAC contract HC104705D4000 under the sponsorship    *
 * of the Defense Technical Information Center,                                       *
 * ATTN: DTIC-AI, 8723 John J. Kingman Rd., Ste 0944, Fort Belvoir, VA 22060-6218.    *
 * SENSIAC is a DoD Information Analysis Center Sponsored by the Defense Technical    *
 * Information Center.                                                                *
 *                                                                                    *
 * HANDLING AND DESTRUCTION NOTICE - Comply with distribution statement and destroy *
 * by any method that will prevent disclosure of the contents or reconstruction of    *
 * the document.                                                                      *
 *                                                                                    *
 * Copyright (c) Vanderbilt University, 2016                                          *
 *                                                                                    *
 * ALL RIGHTS RESERVED, UNLESS OTHERWISE STATED                                       *
 *                                                                                    *
 * Developed under contract for the U.S. Department of Defense                        *
 * by the Institute for Software Integrated Systems (ISIS), Vanderbilt University     *
 *                                                                                    *
 * The FACE Consortium has approved results from this release for                    *
 * certifying software components as FACE Conformant                                  *
 *                                                                                    *
 * Vanderbilt University disclaims all warranties with regard to this software,       *
 * including all implied warranties of merchantability and fitness. In no event       *
 * shall Vanderbilt University be liable for any special, indirect or                 *
 * consequential damages or any damages whatsoever resulting from loss of use,        *
 * data or profits, whether in an action of contract, negligence or other tortious    *
 * action, arising out of or in connection with the use or performance of this        *
 * software.                                                                          *
 *                                                                                    *
 * Conformance Test Suite Version 2.1.0r5 supporting FACE Standard Version 2.1        *
 *                                                                                    *
 * This file is generated and may be overwritten.                                     *
 *                                                                                    *
 * Release Date: 18-May-2017                                                          *
 *************************************************************************************/

#include <spawn.h>

/* restrict is not a keyword in C++ so define it away. */
#ifdef___cplusplus
    #ifndef restrict
        #define restrict
    #endif
#endif

int main()
{
    posix_spawnattr_t *restrict* var42 = {0};
    const sigset_t *restrict* var43 = {0};

    posix_spawnattr_setsigmask(*var42, *var43);

    return 0;
}
```

Test Log File:

```
command:
make safetyExt/spawn.h/Pass/Test000009
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o safetyExt/spawn.h/Pass/
Test000009.o -c safetyExt/spawn.h/Pass/Test000009.c
gcc -o safetyExt/spawn.h/Pass/Test000009 safetyExt/spawn.h/Pass/Test000009.o
/usr/src/lynx/los178/lib/libc.a /usr/src/lynx/los178/lib/libm.a
safetyExt/spawn.h/Pass/Test000009.o: In function `main':
Test000009.c:(.text+0x44): undefined reference to `posix_spawnattr_setsigmask'
collect2: ld returned 1 exit status
make: *** [safetyExt/spawn.h/Pass/Test000009] Error 1
```

Analysis

LynxOS-178 production mode does not provide the following POSIX API functions for the FACE OSS Safety Extended profile as required by Appendix A of the FACE 2.1 Technical Standard:

posix_spawn(),posix_spawnattr_destroy(),
posix_spawnattr_getflags(),posix_spawnattr_getsigdefault(),
posix_spawnattr_getsigmask(),posix_spawnattr_init(),
posix_spawnattr_setflags(),posix_spawnattr_setsigdefault(),
posix_spawnattr_setsigmask()

These functions are available in development mode only. User applications cannot use these functions in production mode.

Component: HMFM Conformance Tests

The 11 failed test cases below test the existence of various types and constants provided by the FACE HMFM API.

Test for basic FACE types

Testing FACE types are defined correctly → FAILED

Test Code:

```
// DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.
//
// Product produced under DoD SENSIAC contract HC104705D4000 under
// the sponsorship of the Defense Technical Information Center,
// ATTN: DTIC-AI, 8723 John J. Kingman Rd., Ste 0944, Fort Belvoir, VA
// 22060-6218. SENSIAC is a DoD Information Analysis Center Sponsored
// by the Defense Technical Information Center.
//
// HANDLING AND DESTRUCTION NOTICE - Comply with distribution statement
// and destroy by any method that will prevent disclosure of
// contents or reconstruction of the document.

/* CONFORMANCE_TEST_FACE_EXACT_TYPES.h contain the correct mapping (user specified) */
/* for your build environment. See manual for more information */
#include "CONFORMANCE_TEST_FACE_EXACT_TYPES.h"

/* user_includes.h contain all header files specified in configuration file for segment
*/
#include "user_includes.h"

/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

#include <stddef.h> /* To get the wchar_t value */

int main(int argc, char *argv[])
{

    /* Checking sizes of all FACE defined types using the values in
CONFORMANCE_TEST_FACE_EXACT_TYPES.h */
    /* FACE defined types are on the left, conformance test values are on the right */
    {
        FACE_STATIC_ASSERT(sizeof(FACE_short) == sizeof(FACE_int16_t));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    {
        FACE_STATIC_ASSERT(sizeof(FACE_long) == sizeof(FACE_int32_t));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    {
        FACE_STATIC_ASSERT(sizeof(FACE_long_long) == sizeof(FACE_int64_t));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    {
        FACE_STATIC_ASSERT(sizeof(FACE_unsigned_short) == sizeof(FACE_uint16_t));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    {
        FACE_STATIC_ASSERT(sizeof(FACE_unsigned_long) == sizeof(FACE_uint32_t));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    {
        FACE_STATIC_ASSERT(sizeof(FACE_unsigned_long_long) ==sizeof(FACE_uint64_t));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    {
        FACE_STATIC_ASSERT(sizeof(FACE_char) == sizeof(FACE_uint8_t));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    {
        FACE_STATIC_ASSERT(sizeof(FACE_octet) == sizeof(FACE_uint8_t));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    /* These are defined exactly by the C standard */
    {
        FACE_STATIC_ASSERT(sizeof(FACE_float) == sizeof(float));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    {
        FACE_STATIC_ASSERT(sizeof(FACE_double) == sizeof(double));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }


    {
        FACE_STATIC_ASSERT(sizeof(FACE_long_double) == sizeof(long double));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }


    {
        FACE_STATIC_ASSERT(sizeof(FACE_wchar) == sizeof(wchar_t));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    {
        FACE_STATIC_ASSERT(sizeof(FACE_boolean) == sizeof(_Bool));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    return 0;
}
```

Test Log File:

```
command:
make Test1
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test1.o -c Test1.c
Test1.c: In function 'main':
Test1.c:31:3: error: 'FACE_short' undeclared (first use in this function)
Test1.c:31:3: note: each undeclared identifier is reported only once for each
function it appears in
Test1.c:36:3: error: 'FACE_long' undeclared (first use in this function)
Test1.c:41:3: error: 'FACE_long_long' undeclared (first use in this function)
Test1.c:46:3: error: 'FACE_unsigned_short' undeclared (first use in this function)
Test1.c:51:3: error: 'FACE_unsigned_long' undeclared (first use in this function)
Test1.c:56:3: error: 'FACE_unsigned_long_long' undeclared (first use in this function)
Test1.c:61:3: error: 'FACE_char' undeclared (first use in this function)
Test1.c:66:3: error: 'FACE_octet' undeclared (first use in this function)
Test1.c:72:3: error: 'FACE_float' undeclared (first use in this function)
Test1.c:77:3: error: 'FACE_double' undeclared (first use in this function)
Test1.c:83:3: error: 'FACE_long_double' undeclared (first use in this function)
Test1.c:89:3: error: 'FACE_wchar' undeclared (first use in this function)
Test1.c:94:3: error: 'FACE_boolean' undeclared (first use in this function)
make: *** [Test1.o] Error 1
```

Test for FACE_SYSTEM_TIME_TYPE

Testing FACE_SYSTEM_TIME_TYPE conformance → FAILED

Test Code:

```
/* Test2 */

/* user_includes.h is generated every test run and contains all header files */
/* specified in the configuration file for the segment under test            */
#include "user_includes.h"

/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

int main(int argc, char *argv[])
{
    /* typedef conformance test */
    {
        /* FACE_SYSTEM_TIME_TYPE size test (FACE_long_long) */
        FACE_STATIC_ASSERT(sizeof(FACE_SYSTEM_TIME_TYPE) == sizeof(FACE_long_long));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    /* FACE_SYSTEM_TIME_TYPE type test (FACE_long_long) */
    FACE_SYSTEM_TIME_TYPE var = 0;
    FACE_long_long cvar;
    cvar = var;
    var = cvar;  /* code used to avoid unused variable warning */
}
```

Test Log File:

```
command:
make Test2
return code:
2
```

```
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test2.o -c Test2.c
Test2.c: In function 'main':
Test2.c:18:9: error: 'FACE_long_long' undeclared (first use in this function)
Test2.c:18:9: note: each undeclared identifier is reported only once for each
function it appears in
Test2.c:24:20: error: expected ';' before 'cvar'
Test2.c:25:5: error: 'cvar' undeclared (first use in this function)
make: *** [Test2.o] Error 1
```

Test for FACE_INF_TIME_VALUE

Testing FACE_INF_TIME_VALUE conformance → FAILED

Test Code:

```
/* Test3 */

/* user_includes.h is generated every test run and contains all header files */
/* specified in the configuration file for the segment under test             */
#include "user_includes.h"

/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

int main(int argc, char *argv[])
{
    /* constant conformance test */
    {
        /* FACE_INF_TIME_VALUE size test (FACE_SYSTEM_TIME_TYPE) */
        FACE_STATIC_ASSERT(sizeof(FACE_INF_TIME_VALUE) ==
sizeof(FACE_SYSTEM_TIME_TYPE));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    {
        /* Testing constant FACE_INF_TIME_VALUE value (-1) is conformant */
        FACE_STATIC_ASSERT(FACE_INF_TIME_VALUE ==-1);
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }
}
```

Test Log File:

```
command:
make Test3
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test3.o -c Test3.c
Test3.c: In function 'main':
Test3.c:18:9: error: 'FACE_INF_TIME_VALUE' undeclared (first use in this function)
Test3.c:18:9: note: each undeclared identifier is reported only once for each
function it appears in
make: *** [Test3.o] Error 1
```

Test for FACE_CONFIGURATION_RESOURCE

Testing FACE_CONFIGURATION_RESOURCE conformance → FAILED

Test Code:

```
/* Test4 */

/* user_includes.h is generated every test run and contains all header files */
/* specified in the configuration file for the segment under test             */
#include "user_includes.h"

/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

int main(int argc, char *argv[])
{
    /* typedef sequence test */
    {
        /* FACE_CONFIGURATION_RESOURCE size test */
        FACE_STATIC_ASSERT(sizeof(FACE_CONFIGURATION_RESOURCE) ==
sizeof(FACE_char[256]));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    // Testing FACE_CONFIGURATION_RESOURCE type and size is conformant
    FACE_CONFIGURATION_RESOURCE var;
    FACE_char cvar;
    var[0] = var[256-1] = 0;
    cvar = var[0];
}
```

Test Log File:

```
command:
make Test4
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test4.o -c Test4.c
Test4.c: In function 'main':
Test4.c:18:13: error: 'FACE_CONFIGURATION_RESOURCE' undeclared (first use in
this function)
Test4.c:18:13: note: each undeclared identifier is reported only once for each
function it appears in
Test4.c:18:13: error: 'FACE_char' undeclared (first use in this function)
Test4.c:22:33: error: expected ';' before 'var'
Test4.c:23:15: error: expected ';' before 'cvar'
Test4.c:24:5: error: 'var' undeclared (first use in this function)
Test4.c:25:5: error: 'cvar' undeclared (first use in this function)
make: *** [Test4.o] Error 1
```

Test for FACE_RETURN_CODE_TYPE

Testing FACE_RETURN_CODE_TYPE conformance → FAILED

Test Code:

```
/* Test5 */
/* user_includes.h is generated every test run and contains all header files */
/* specified in the configuration file for the segment under test             */
#include "user_includes.h"

/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

int main(int argc, char *argv[])
{
    /* Enumeration Conformance Test */
    {
        /* Enumeration the same size as the enumeration under test */
        typedef enum {
            FACE_NO_ERROR_CONFORMANCE_TEST,
            FACE_NO_ACTION_CONFORMANCE_TEST,
            FACE_NOT_AVAILABLE_CONFORMANCE_TEST,
            FACE_ADDR_IN_USE_CONFORMANCE_TEST,
            FACE_INVALID_PARAM_CONFORMANCE_TEST,
            FACE_INVALID_CONFIG_CONFORMANCE_TEST,
            FACE_PERMISSION_DENIED_CONFORMANCE_TEST,
            FACE_INVALID_MODE_CONFORMANCE_TEST,
            FACE_TIMED_OUT_CONFORMANCE_TEST,
            FACE_MESSAGE_STALE_CONFORMANCE_TEST,
            FACE_CONNECTION_IN_PROGRESS_CONFORMANCE_TEST,
            FACE_CONNECTION_CLOSED_CONFORMANCE_TEST,
            FACE_DATA_BUFFER_TOO_SMALL_CONFORMANCE_TEST
        } FACE_RETURN_CODE_TYPE_CONFORMANCE_TEST;

        /* FACE_RETURN_CODE_TYPE size test (Enumeration)*/
        FACE_STATIC_ASSERT(sizeof(FACE_RETURN_CODE_TYPE) ==
sizeof(FACE_RETURN_CODE_TYPE_CONFORMANCE_TEST));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    /* FACE_RETURN_CODE_TYPE test that all enumeration items exists */
    FACE_RETURN_CODE_TYPE var;

    var = FACE_NO_ERROR;
    var = FACE_NO_ACTION;
    var = FACE_NOT_AVAILABLE;
    var = FACE_ADDR_IN_USE;
    var = FACE_INVALID_PARAM;
    var = FACE_INVALID_CONFIG;
    var = FACE_PERMISSION_DENIED;
    var = FACE_INVALID_MODE;
    var  =  FACE_TIMED_OUT;
    var = FACE_MESSAGE_STALE;
    var = FACE_CONNECTION_IN_PROGRESS;
    var = FACE_CONNECTION_CLOSED;
    var = FACE_DATA_BUFFER_TOO_SMALL;
}
```

Test Log File:

```
command:
make Test5
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test5.o -c Test5.c
Test5.c: In function 'main':
Test5.c:47:11: error: 'FACE_ADDR_IN_USE' undeclared (first use in this function)
Test5.c:47:11: note: each undeclared identifier is reported only once for each
```

```
function it appears in
Test5.c:50:11: error: 'FACE_PERMISSION_DENIED' undeclared (first use in this function)
Test5.c:53:11: error: 'FACE_MESSAGE_STALE' undeclared (first use in this function)
Test5.c:54:11: error: 'FACE_CONNECTION_IN_PROGRESS' undeclared (first use in
this function)
Test5.c:55:11: error: 'FACE_CONNECTION_CLOSED' undeclared (first use in this function)
Test5.c:56:11: error: 'FACE_DATA_BUFFER_TOO_SMALL' undeclared (first use in this
function)
make: *** [Test5.o] Error 1
```

Test for `FACE_TIMEOUT_TYPE`

Testing `FACE_TIMEOUT_TYPE` conformance → FAILED

Test Code:

```
/* Test7 */

/* user_includes.h is generated every test run and contains all header files */
/* specified in the configuration file for the segment under test             */
#include "user_includes.h"

/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

int main(int argc, char *argv[])
{
    /* typedef conformance test */
    {
        /* FACE_TIMEOUT_TYPE size test (FACE_SYSTEM_TIME_TYPE) */
        FACE_STATIC_ASSERT(sizeof(FACE_TIMEOUT_TYPE) == sizeof(FACE_SYSTEM_TIME_TYPE));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    /* FACE_TIMEOUT_TYPE type test (FACE_SYSTEM_TIME_TYPE) */
    FACE_TIMEOUT_TYPE var = 0;
    FACE_SYSTEM_TIME_TYPE cvar;
    cvar = var;
    var = cvar;  /* code used to avoid unused variable warning */
}
```

Test Log File:

```
command:
make Test7
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test7.o -c Test7.c
Test7.c: In function 'main':
Test7.c:18:9: error: 'FACE_TIMEOUT_TYPE' undeclared (first use in this function)
Test7.c:18:9: note: each undeclared identifier is reported only once for each
function it appears in
Test7.c:23:23: error: expected ';' before 'var'
Test7.c:25:12: error: 'var' undeclared (first use in this function)
make: *** [Test7.o] Error 1
```

Test for FACE_MESSAGE_RANGE_TYPE

Testing FACE_MESSAGE_RANGE_TYPE conformance → FAILED

Test Code:

```
/* Test8 */

/* user_includes.h is generated every test run and contains all header files */
/* specified in the configuration file for the segment under test            */
#include "user_includes.h"

/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

int main(int argc, char *argv[])
{
    /* typedef conformance test */
    {
        /* FACE_MESSAGE_RANGE_TYPE size test (FACE_long) */
        FACE_STATIC_ASSERT(sizeof(FACE_MESSAGE_RANGE_TYPE) == sizeof(FACE_long));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    /* FACE_MESSAGE_RANGE_TYPE type test (FACE_long) */
    FACE_MESSAGE_RANGE_TYPE var = 0;
    FACE_long cvar;
    cvar = var;
    var = cvar;  /* code used to avoid unused variable warning*/
}
```

Test Log File:

```
command:
make Test8
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test8.o -c Test8.c
Test8.c: In function 'main':
Test8.c:18:9: error: 'FACE_long' undeclared (first use in this function)
Test8.c:18:9: note: each undeclared identifier is reported only once for each
function it appears in
Test8.c:24:15: error: expected ';' before 'cvar'
Test8.c:25:5: error: 'cvar' undeclared (first use in this function)
make: *** [Test8.o] Error 1
```

Test for FACE_HMFM_FAULT_MESSAGE_SIZE_TYPE

Testing FACE_HMFM_FAULT_MESSAGE_SIZE_TYPE conformance → FAILED

Test Code:

```
/* Test9 */

/* user_includes.h is generated every test run and contains all header files */
/* specified in the configuration file for the segment under test            */
#include "user_includes.h"
```

```
/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

int main(int argc, char *argv[])
{
    /* typedef conformance test */
    {
        /* FACE_HMFM_FAULT_MESSAGE_SIZE_TYPE size test (FACE_long) */
        FACE_STATIC_ASSERT(sizeof(FACE_HMFM_FAULT_MESSAGE_SIZE_TYPE) ==
sizeof(FACE_long));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    /* FACE_HMFM_FAULT_MESSAGE_SIZE_TYPE type test (FACE_long) */
    FACE_HMFM_FAULT_MESSAGE_SIZE_TYPE var = 0;
    FACE_long cvar;
    cvar = var;
    var = cvar;  /* code used to avoid unused variable warning */
}
```

Test Log File:

```
command:
make Test9
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test9.o -c Test9.c
Test9.c: In function 'main':
Test9.c:18:9: error: 'FACE_long' undeclared (first use in this function)
Test9.c:18:9: note: each undeclared identifier is reported only once for each
function it appears in
Test9.c:24:15: error: expected ';' before 'cvar'
Test9.c:25:5: error: 'cvar' undeclared (first use in this function)
make: *** [Test9.o] Error 1
```

Test for `FACE_HMFM_THREAD_ID_TYPE`

Testing `FACE_HMFM_THREAD_ID_TYPE` conformance → FAILED

Test Code:

```
/* Test11 */

/* user_includes.h is generated every test run and contains all header files */
/* specified in the configuration file for the segment under test            */
#include "user_includes.h"

/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

int main(int argc, char *argv[])
{
    /* typedef conformance test */
    {
        /* FACE_HMFM_THREAD_ID_TYPE size test (FACE_long) */
        FACE_STATIC_ASSERT(sizeof(FACE_HMFM_THREAD_ID_TYPE) == sizeof(FACE_long));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    /* FACE_HMFM_THREAD_ID_TYPE type test (FACE_long) */
    FACE_HMFM_THREAD_ID_TYPE var = 0;
    FACE_long cvar;
    cvar = var;
    var = cvar;  /* code used to avoid unused variable warning*/
}
```

Test Log File:

```
command:
make Test11
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter //usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test11.o -c Test11.c
Test11.c: In function 'main':
Test11.c:18:9: error: 'FACE_long' undeclared (first use in this function)
Test11.c:18:9: note: each undeclared identifier is reported only once for each
function it appears in
Test11.c:24:15: error: expected ';' before 'cvar'
Test11.c:25:5: error: 'cvar' undeclared (first use in this function)
make: *** [Test11.o] Error 1
```

Test for FACE_HMFM_STACK_SIZE_TYPE

Testing FACE_HMFM_STACK_SIZE_TYPE conformance → FAILED

Test Code:

```
/* Test13 */

/* user_includes.h is generated every test run and contains all header files */
/* specified in the configuration file for the segment under test             */
#include "user_includes.h"

/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

int main(int argc, char *argv[])
{
    /* typedef conformance test */
    {
        /* FACE_HMFM_STACK_SIZE_TYPE size test (FACE_unsigned_long) */
        FACE_STATIC_ASSERT(sizeof(FACE_HMFM_STACK_SIZE_TYPE) ==
sizeof(FACE_unsigned_long));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    /* FACE_HMFM_STACK_SIZE_TYPE type test (FACE_unsigned_long) */
    FACE_HMFM_STACK_SIZE_TYPE var = 0;
    FACE_unsigned_long cvar;
    cvar = var;
    var = cvar;  /* code used to avoid unused variable warning*/
}
```

Test Log File:

```
command:
make Test13
return code:
```

```
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test13.o -c Test13.c
Test13.c: In function 'main':
Test13.c:18:9: error: 'FACE_unsigned_long' undeclared (first use in this function)
Test13.c:18:9: note: each undeclared identifier is reported only once for each
function it appears in
Test13.c:24:24: error: expected ';' before 'cvar'
Test13.c:25:5: error: 'cvar' undeclared (first use in this function)
make: *** [Test13.o] Error 1
```

Test for FACE_HMFM_FAULT_MESSAGE_TYPE

Testing FACE_HMFM_FAULT_MESSAGE_TYPE conformance → FAILED

Test Code:

```
/* Test14 */

/* user_includes.h is generated every test run and contains all header files */
/* specified in the configuration file for the segment under test            */
#include "user_includes.h"

/* Used to catch assertions at compile time */
#define FACE_STATIC_ASSERT(e) char face_static_assert[ e ? 1 : -1 ]

int main(int argc, char *argv[])
{
    /* typedef sequence test */
    {
        /* FACE_HMFM_FAULT_MESSAGE_TYPE size test */
        FACE_STATIC_ASSERT(sizeof(FACE_HMFM_FAULT_MESSAGE_TYPE) ==
sizeof(FACE_char[FACE_HMFM_FAULT_MESSAGE_MAXIMUM_SIZE]));
        face_static_assert[0] += 1; /* code used to avoid unused variable warning */
    }

    // Testing FACE_HMFM_FAULT_MESSAGE_TYPE type and size is conformant
    FACE_HMFM_FAULT_MESSAGE_TYPE var;
    FACE_char cvar;
    var[0] = var[FACE_HMFM_FAULT_MESSAGE_MAXIMUM_SIZE-1] = 0;
    cvar = var[0];
}
```

Test Log File:

```
command:
make Test14
return code:
2
Output:
gcc -std=c99 -nostdinc -I. -idirafter /usr/src/lynx/los178/usr/include
-DLynx -D__Lynx_ -Dpowerpc -D__powerpc_ -fsigned-char -fcommon  -fno-builtin
-I/usr/src/lynx/los178/tst/FACE/FACEConformanceTestSuite-2.1.0r5/
compilerSpecific/C/systemLibraryDefinitions    -o Test14.o -c Test14.c
Test14.c: In function 'main':
Test14.c:18:13: error: 'FACE_char' undeclared (first use in this function)
Test14.c:18:13: note: each undeclared identifier is reported only once for each
function it appears in
Test14.c:23:15: error: expected ';' before 'cvar'
Test14.c:25:5: error: 'cvar' undeclared (first use in this function)
make: *** [Test14.o] Error 1
```

Analysis

LynxOS-178 does not define the following FACE HMFM API types as required by Sec. 3.6.4.1.2.1 of the FACE 2.1 Technical Standard:

> `FACE_short`, `FACE_long`, `FACE_long_long`, `FACE_unsigned_short`, `FACE_unsigned_long`, `FACE_unsigned_long_long`, `FACE_float`, `FACE_double`, `FACE_long_double`, `FACE_char`, `FACE_wchar`, `FACE_octet`, `FACE_fixed_d_s`, `FACE_boolean`

LynxOS-178 does not define the following FACE HMFM API types and constants as required by Appendix B and Appendix C Sec. C.2 of the FACE 2.1 Technical Standard:

> `FACE_TIMEOUT_TYPE`, `FACE_INF_TIME_VALUE`, `FACE_CONFIGURATION_RESOURCE`, `FACE_ADDR_IN_USE`, `FACE_MESSAGE_STALE`, `FACE_CONNECTION_IN_PROGRESS`, `FACE_CONNECTION_CLOSED`, `FACE_DATA_BUFFER_TOO_SMALL`

User applications that need to use these missing types and constants would need to define them appropriately by themselves.

CHAPTER 3 Analysis of Matrix Requirements

Fully Supported Requirements

As described in Chapter 1, the FACE Conformance Verification Matrix filtered for LynxOS-178 produces a total of 65 requirements to be verified. Other than those described in the subsequent sections, all requirements are fully supported. This section provides additional commentary for some requirements that are fully supported. The total number of fully supported requirements is 44.

Requirement: Sec. 3.1.1 #2

"The operating system shall execute within the OSS."

LynxOS-178 bundles multiple components such as the operating system kernel, C language run-time and standard library, ARINC 653 support driver and library, etc. As a candidate UoC, all of LynxOS-178's components are defined to reside and execute within the OSS of the FACE reference architecture.

Requirement: Sec. 3.1.1 #7

"If the FACE programming language run-time or FACE framework as defined in Section 3.12 and Section 3.13 are bundled with the Operating System kernel and use a proprietary interface, then they shall be a component of the OSS delivered as a singular bundle."

LynxOS-178 does not bundle any language run-time utilizing any non-standard proprietary interface other than the C language run-time and standard library. As a candidate UoC, all of LynxOS-178's components are defined to reside and execute within the OSS of the FACE reference architecture.

Requirement: Sec. 3.1.1 #21 (a)

"For Safety-profiled partitions, only one partition shall be allowed write access permissions to a volume, per ARINC 653 Part 2."

The satisfaction of this requirement needs appropriate manual configuration.

The LynxOS-178 partitioning implementation establishes a one-to-one correspondence between the concepts of UIDs and ARINC 653 partitions. File I/O functions defined by ARINC 653 (such as `OPEN_FILE()`, `WRITE_FILE()`, `MAKE_DIRECTORY()`, etc.) support existing requirements ensuring that a particular user (and therefore, a partition) is allowed write access to a given file on a given volume only if the corresponding permission is granted. In conjunction with standard file permissions, the Virtual Machine Configuration Table used to configure partitions and file systems can be specified in a way such that only one user (and therefore, only one partition) has write access to a particular volume.

Requirement: Sec. 3.1.1 #25

"Time and space partitioning shall be provided for the Safety Profile when required by the system.

Note: The OSS permits the system integrator to choose the type of partitioning (e.g., time, space, time and space, none) for a computing platform running in the FACE Safety Profile. The minimum suggested type is space partitioning."

LynxOS-178 provides time and space partitioning unconditionally.

Requirement: Sec. 3.1.1 #27

"Whenever partitions are allocated identical time windows, all threads/processes associated with the applications assigned to the partitions shall utilize a priority preemptive scheduling algorithm.

Note: It is intended that Operating Systems supporting partitions with identical time windows also provide priority preemptive scheduling.

Note: The threads in these partitions may not be considered sufficiently segregated to allow qualification at different levels of criticality."

LynxOS-178 implements a priority preemptive algorithm for intrapartition scheduling and a fixed cyclic algorithm for interpartition scheduling. If partitions are allocated identical time windows, the processes and threads in them will effectively be scheduled as per the priority preemptive intrapartition scheduling algorithm, when the corresponding partition is running.

Requirement: Sec. 3.11.4

"Partial support shall be included for POSIX `CLOCK_REALTIME` that permits a calendar-based clock time to be set so it becomes available to all partitions. Due to safety and security-related considerations in the POSIX definitions of this clock's behavior, the configuration data specifies which partitions (preferably one partition) have permission to set (using the `clock_settime()` API) this clock. All partitions may have access to read this clock, but an error is returned for any partition that attempts to set an alarm based on this clock that is based on absolute time (setting of an alarm based on a relative time is supported). Partitions authorized to set the calendar time are also authorized to set the time zone value that is visible to all other partitions."

The LynxOS-178 implementation of the `clock_settime()` POSIX API allows only a process running as effective UID 0 (i.e., the VM0 partition) to set the calendar-based system clock `CLOCK_REALTIME`. This behavior is unconditional and not user-configurable.

Requirement: Sec. 3.15 #2

"FACE components shall use this configuration information, when required, to initialize parameters of the software component, or to enable or disable certain software component capabilities."

LynxOS-178 uses the Virtual Machine Configuration Table (VCT) to specify configuration data for the operating system, individual partitions, device drivers, or file systems. VCT-configurable items include initialization parameters, partitioning parameters, limits on resources, permissions, etc.

Requirement: Sec. 3.16.2 #4

"The HMFM shall be configured at system start-up."

Requirement: Sec. 3.16.2 #5

"The HMFM shall provide monitoring to detect faults and failures."

Requirement: Sec. 3.16.2 #6

"The HMFM shall execute response or recovery actions for detected faults and failures as configured."

Requirement: Sec. 3.16.3 #1

"The OSS shall include a Health Monitor (HM) functionality at the module level (i.e., computing platform level), partition-level, and process-level (i.e., thread-level).

Note: Health Monitor functionality includes start and terminating the module, restart, and idle partitions within the module, and halting and restarting threads."

LynxOS-178 supports HMFM capabilities at the system level via a dedicated kernel thread and device driver. In a typical production system, the HM driver may be loaded at start-up with a health monitor application running in the VM0 partition. This driver provides an API that allows other LynxOS-178 components to log or raise system errors for fault management. Configurable parameters of the HM driver component (such as the desired action for exceptions, etc.) are specified in the VCT file that is parsed at system start-up. LynxOS-178 components may automatically invoke this system-level functionality on conditions such as resource exhaustion or exceptions (via `khm_post()`, etc.).

The ARINC 653 component of LynxOS-178 provides the HM API to create and use user-implemented threads for partition-level fault management. The C Standard Library component of LynxOS-178 provides the FACE HMFM API to create and use user-implemented threads for process-level fault management. In a typical production system, the user may create the HMFM threads at start-up. Thereafter, user applications may programmatically invoke partition-level or process-level functionality using standard ARINC 653 HM or FACE HMFM APIs (such as `RAISE_APPLICATION_ERROR()`, `Raise_Application_Fault()`, etc.).

Requirement: Sec. 3.16.3 #4

"The ARINC 653 Part 1 Health Monitoring Types and Health Monitoring Services shall be available to a POSIX-based partition."

For user applications linked with the ARINC 653 support library, LynxOS-178 uses an internal implementation based on the ARINC 653 partition-level HM API to provide support for HMFM capabilities. Otherwise, LynxOS-178 uses an internal implementation in the C Standard Library based on the POSIX process-level FACE HMFM API to provide support for HMFM capabilities.

Requirement: Appendix A #3

"For the Security and Safety Profiles, memory mapping support (e.g., `mmap()` API) shall be `_POSIX_SHARED_MEMORY_OBJECTS` only."

LynxOS-178 defines the `_POSIX_SHARED_MEMORY_OBJECTS` POSIX.1 option but not the `_POSIX_MAPPED_FILES` POSIX.1 option. Memory mapping support in the `mmap()` POSIX API is restricted to shared memory objects only.

Requirement: Appendix C Sec. C.3

"Upon return, the `return_code` output parameter contains a value indicating that the method executed successfully or failed for a specific reason. The return code value returned from `Initialize(HMFM)` shall be one of the following:

- *`NO_ERROR` to indicate successful completion of the operation*
- *`INVALID_CONFIG` to indicate that an underlying operating system API call failed"*

The LynxOS-178 implementation of the `Initialize()` FACE HMFM API never fails and always returns `NO_ERROR`.

Requirement: Appendix C Sec. C.5

"The `Report_Application_Message(HMFM)` method shall allow for a component to send a message to the HM fault handler which invokes the registered fault handler to process the message."

The LynxOS-178 implementation of the `Report_Application_Message()` FACE HMFM API writes the given message to the system-level HM driver that queues it into a system-global pool. It is up to the user application to ensure that the user-implemented process-level fault handler thread created and registered via `Create_Fault_Handler()` is implemented such that it periodically reads the system-global pool via the system-level HM driver to dequeue any messages for itself.

Requirement: Appendix C Sec. C.6

"The `Get_Fault_Status(HMFM)` method shall allow for the fault handler registered by a component to obtain information regarding the current fault."

The LynxOS-178 implementation of the `Get_Fault_Status()` FACE HMFM API is intended to be used by the user-implemented process-level fault handler thread created and registered via `Create_Fault_Handler()`.

Partially Supported Requirements

This section provides analyses for requirements that are partially supported. The total number of partially supported requirements is 16.

Requirement: Sec. 3.1.1 #10

"The Operating System shall support ARINC 653 defined configuration data types (XML-based schema) for allocating computing platform resources."

Requirement: Sec. 3.15 #3

"The parameters or capabilities that are configurable shall be defined in the specification for each FACE component in XML."

Requirement: Sec. 3.15 #4

"The XML file format shall be defined by an XSD including but not limited to the following elements:

a. *Data elements and structure are defined by:*

i. Descriptive information

b. Data elements are constrained by:

i. Data types

ii. Limits

iii. Precision

iv. Valid values

v. XML Schema Version"

Requirement: Sec. 3.15.2 #4

"Each FACE component shall provide schema information for required configuration data."

LynxOS-178 uses the Virtual Machine Configuration Table (VCT) to specify configuration data for the operating system, individual partitions, device drivers, or file systems, including resource allocation information. The VCT is not XML; it is a text-based file employing a formal syntactical grammar essentially equivalent in functionality to XML. The VCT format, including data elements, descriptions, types, limits, precisions, valid values and the VCT format version, is defined as a set of production rules in Backus Naur Form notation that is essentially equivalent in functionality to XSD.

Requirement: Sec. 3.11 #8

"FACE operating systems shall specify which partitions are specifically authorized to bind/connect to an endpoint, receive on an endpoint, or transmit on an endpoint."

The satisfaction of this requirement needs appropriate manual configuration.

The LynxOS-178 partitioning implementation establishes a one-to-one correspondence between the concepts of UIDs and ARINC 653 partitions. UNIX domain stream sockets functionality is bundled as part of LynxOS-178. Such sockets use a special file with a known path as the address for client-server communication. The corresponding `connect()` and `bind()` functions (the prerequisites to transmit or receive data over them) support existing requirements ensuring that a particular user (and therefore, a partition) can successfully invoke them only if the appropriate permission to open or create the special file at the given path on the corresponding file system is granted. In conjunction with standard file permissions, the Virtual Machine Configuration Table used to configure partitions and file systems can be specified in a way such that only one user (and therefore, only one partition) has the necessary permissions to connect() or `bind()` (and therefore, transmit or receive) to some specific UNIX domain socket with a given path address. Non-UNIX domain sockets are supported by LynxOS-178 if and when provided as part of a separate network stack; whether they support the required functionality depends on the particular network stack implementation.

Requirement: Sec. 3.11 #9

"FACE operating systems shall configure sockets:

a. Specify the size of a socket's receive buffer (space reserved for the socket) and a socket's transmit buffer

b. Specify and enforce a socket's maximum receive and transmit message size"

LynxOS-178 provides the `setsockopt()` POSIX API that may be used to configure the mentioned socket options. UNIX domain stream sockets functionality is bundled as part of LynxOS-178, but does not support the `setsockopt()` function. Non-UNIX domain sockets are supported by LynxOS-178 if and when provided as part of a separate network stack; whether they support the mentioned socket options depends on the particular network stack implementation.

Requirement: Sec. 3.11 #10

"FACE operating systems shall get the status of a socket:

a. The size of its receive buffer (space reserved for the socket) and transmit buffer

b. Maximum receive and transmit message size

c. Retrieve configuration status that indicates success or an error status"

LynxOS-178 provides the `getsockopt()` POSIX API that may be used to obtain the mentioned socket options. UNIX domain stream sockets functionality is bundled as part of LynxOS-178, and supports the `getsockopt()` function only to retrieve peer credentials or success/error status (i.e., sub item *c*). Non-UNIX domain sockets are supported by LynxOS-178 if and when provided as part of a separate network stack; whether they support the mentioned socket options depends on the particular network stack implementation.

Requirement: Sec. 3.11.2

"FACE Safety Operating System implementations shall provide the POSIX APIs defined in Appendix A for the Safety (Base and/or Extended) Profiles."

Requirement: Sec. 3.11.2

"FACE Safety Profile implementations for POSIX shall consist of:

1. *Subsets of APIs (supported subset adds to the FACE Security Operating System API) from the following POSIX 1003.13 Profile 53 service categories as defined in Appendix A:*

   a. C Language Math

   b. C Language Support

   c. Threads

   d. Thread Attributes and Priority Scheduling

   e. Mutexes

   f. Shared Memory Objects

   g. Message Queues

   h. Signals and Real-time Signals

   i. Semaphores

   j. Timers

   k. Single-Process

   l. Multi-Process

   m. Spawn

   n. File System

   o. Device I/O

2. *Subset of APIs associated with:*

   a. Sockets

   b. ARINC 653 Sampling and Queuing Port APIs (when communicating with an ARINC 653-based component)"

Requirement: Appendix A

"Table of POSIX APIs"

LynxOS-178 provides all POSIX 1003.13 Profile 53 APIs defined in Appendix A of the FACE 2.1 Technical Standard for the Safety Extended profile, except for the following related to `_POSIX_SPAWN`:

```
posix_spawn(),posix_spawnattr_destroy(),
posix_spawnattr_getflags(),
```

`posix_spawnattr_getsigdefault()`, `posix_spawnattr_getsigmask()`,`posix_spawnattr_init()`, `posix_spawnattr_setflags()`, `posix_spawnattr_setsigdefault()`, `posix_spawnattr_setsigmask()`

Requirement: Sec. 3.16.2 #1

“The HMFM capabilities shall use ARINC 653 Part 1 Health Monitoring Types and Health Monitoring Services or the FACE HMFM APIs as defined in Appendix C.”

Requirement: Sec. 3.16.2 #2

“The HMFM capabilities shall be provided for the Safety and Security Profiles.”

Requirement: Appendix C Sec. C.2

“HMFM Services API and Message Definitions”

For user applications linked with the ARINC 653 support library, LynxOS-178 uses an internal implementation based on the ARINC 653 partition-level HM API to provide support for HMFM capabilities. Otherwise, LynxOS-178 uses an internal implementation in the C Standard Library based on the POSIX process-level FACE HMFM API to provide support for HMFM capabilities. The FACE HMFM API implementation conforms to the specification in Appendix C of the FACE 2.1 Technical Standard, except for the following:

1. The types `FACE_short`, `FACE_long`, `FACE_long_long`, `FACE_unsigned_short`, `FACE_unsigned_long`, `FACE_unsigned_long_long`, `FACE_float`, `FACE_double`, `FACE_long_double`, `FACE_char`, `FACE_wchar`, `FACE_octet`, `FACE_fixed_d_s` and `FACE_boolean` required by Sec. 3.6.4.1.2.1 of the FACE 2.1 Technical Standard are not defined.
2. The types or constants `FACE_TIMEOUT_TYPE`, `FACE_CONFIGURATION_RESOURCE`, `FACE_INF_TIME_VALUE`, `FACE_ADDR_IN_USE`, `FACE_MESSAGE_STALE`, `FACE_CONNECTION_IN_PROGRESS`, `FACE_CONNECTION_CLOSED`and `FACE_DATA_BUFFER_TOO_SMALL` required by Appendix B and Appendix C Sec. C.2 of the FACE 2.1 Technical Standard are not defined.

3. The names of FACE HMFM API types, constants and functions are not namespace-prefixed with FACE_ or FACE_HMFM_ to map the IDL to a C language implementation.

Requirement: Appendix C Sec. C.4

"Upon return, the return_code output parameter contains a value indicating that the method executed successfully or failed for a specific reason. The return code value returned from Create_Fault_Handler(HMFM) shall be one of the following:

- *NO_ERROR to indicate successful completion of the operation*
- *NO_ACTION to indicate that an fault handler has already been created*
- *INVALID_CONFIG to indicate that the thread could not be created*
- *INVALID_PARAM to indicate that the stack_size parameter is invalid*
- *INVALID_MODE to indicate that the system is in the incorrect mode to perform this operation"*

The LynxOS-178 implementation of the Create_Fault_Handler() FACE HMFM API never returns INVALID_PARAM; instead, INVALID_CONFIG is returned when the stack_size parameter is invalid. It also never returns INVALID_MODE; the corresponding condition is not supported.

Requirement: Appendix C Sec. C.5

"Upon return, the return_code output parameter contains a value indicating that the method executed successfully or failed for a specific reason. The return code value returned from Report_Application_Message(HMFM) shall be one of the following:

- *NO_ERROR to indicate successful completion of the operation*
- *INVALID_PARAM to indicate the length parameter is invalid"*

The LynxOS-178 implementation of the Report_Application_Message() FACE HMFM API returns NOT_AVAILABLE when the given message could not be written to the system-level HM driver.

Requirement: Appendix C Sec. C.6

"When this operation is successful, the fault status referenced by the fault parameter shall contain information regarding the fault. Upon return, the `return_code` output parameter contains a value indicating that the method executed successfully or failed for a specific reason. The return code value returned from `Get_Fault_Status(HMFM)` shall be one of the following:

- *`NO_ERROR` to indicate successful completion of the operation*
- *`INVALID_PARAM` to indicate the `length` parameter is invalid*
- *`INVALID_CONFIG` to indicate that the current thread is not the fault handler*
- *NO_ACTION to indicate that there are no currentfaults"*

The LynxOS-178 implementation of the `Get_Fault_Status()` FACE HMFM API never returns `INVALID_PARAM`or `NO_ACTION`; the corresponding conditions are not supported.

Unsupported Requirements

This section provides analyses for requirements that are not supported. There is 1 unsupported requirement.

Requirement: Appendix A #2

"For the Security and Safety Profiles, mutex support (e.g., `pthread_mutexattr_setprotocol()` API) shall be `_POSIX_THREAD_PRIO_PROTECT` only."

LynxOS-178 defines both the `_POSIX_THREAD_PRIO_PROTECT` and `_POSIX_THREAD_PRIO_INHERIT` POSIX.1 options. Mutex protocol support in the `pthread_mutexattr_setprotocol()` POSIX API is not restricted to the priority protection protocol only, but also provides the priority inheritance protocol as well as the priority agnostic protocol.

Inapplicable Requirements

This section provides analyses for requirements that are not applicable to LynxOS-178. The total number of inapplicable requirements is 4.

Requirement: Sec. 3.1.1 #6

"The Operating System shall support the execution of a framework through the OS interface (e.g., OSGi) when required by the system.

Note: The acceptable frameworks that can be executed by the OSS are defined in Section 3.13."

LynxOS-178 does not bundle any component framework (such as OSGi, etc.) with itself.

Requirement: Sec. 3.17.1

"If an IPv4 and/or IPv6 network is required, then the network shall be implemented in accordance with Section F.1."

Requirement: Sec. 3.17.1

"If any network standard options as defined in Section F.1 are used, then all of the standards applicable to that network standard option shall be used."

Requirement: Appendix F Sec. F.1

"Table 153: Internetwork Standards Summary"

LynxOS-178 does not bundle an IP network stack with itself. IP networking is supported by LynxOS-178 only through a separate network stack; whether an attached IP network stack conforms to the referenced standards depends on its implementation.