National Institute of Standards and Technology
Computer Systems Laboratory

NIST POSIX Testing Policy— Certificate of Validation Requirements for FIPS 151-2

October 1, 1995

NIST POSIX Certification Authority
National Institute of Standards and Technology
Computer Systems Laboratory
Gaithersburg, MD 20899

1. Introduction

General information on the NIST POSIX testing policy is provided in a separate document, *NIST POSIX Testing Policy General Information*. Information on Accredited POSIX Testing Laboratories is provided in the *NIST POSIX Testing Policy General Information* and in the *National Voluntary Laboratory Accreditation Program— Program Handbook— Computer Applications Testing— POSIX Conformance Testing*.

This document presents information specific to conformance testing and *Certificate of Validation* requirements for the Federal Information Processing Standards Publication 151-2. All functions of the Certification Authority, as defined in the *NIST POSIX Testing Policy General Information*, are conducted by the Computer Systems Laboratory of the National Institute of Standards and Technology.

2. Terminology

2.1 Terms

Certification Authority

The Director of NIST/CSL provides the overall direction for organizing, managing, directing, and administering the NIST POSIX testing program.

client

Organization or person who employs an APTL for NIST POSIX testing.

conformance to FIPS 151-2

The state of an implementation satisfying the requirements and specifications of FIPS 151-2 as tested by the Official NIST-PCTS:151-2.

cooperating-hosted system

A single computer system that provides the functionality of both the *development system* and the *host implementation* with a single operating system, and provides the FIPS 151-2 conforming implementation with another operating system.

cooperating system

The combination of the *development system* and the *target system*.

development system

The computer system used to compile and configure a PCTS. [IEEE Std 1003.3-1991, 2.2.2.6]

host implementation

Provides the operating system kernel services needed by a *hosted implementation*.

hosted implementation

Implementation of POSIX.1 that is accomplished through interfaces from the POSIX.1 services to some alternate form of operating system kernel services. [ISO/IEC 9945-1:1990, B.2.2.2]

implementation under test

The software that implements FIPS 151-2.

native implementation

Implementation of POSIX.1 that interfaces directly to an operating system kernel. [ISO/IEC 9945-1:1990, B.2.2.2]

NIST/CSL FIPS 151 Certification Body

The group designated to perform the tasks designated by the Certification Authority.

NIST-PCTS:151-2

Conformance Test Suite for FIPS 151-2

Official NIST-PCTS:151-2

NIST-PCTS:151-2 distributed by NTIS in conjunction with the latest FIXES Patch distributed by NIST/CSL.

system under test

The computer system hardware and software on which the *implementation under test* operates. [IEEE P2003/D3.0, March 1, 1994] target system

The combination of the computer system on which the PCTS is executed and the parts of the development system that are used to generate the executable code of a PCTS. [IEEE Std 1003.3-1991, 2.2.2.13]

2.2 Abbreviations

APTL	Accredited POSIX Testing Laboratory
CSL	Computer System Laboratory (within NIST)
FIPS 151-2	Federal Information Processing Standards Publication 151-2, *Portable Operating System Interface (POSIX)— System Application Interface [C Language]*
FIPS 160	Federal Information Processing Standards Publication 160, ‘C’, March 13, 1991 with change #1, August 24, 1992.
GTI	General Terminal Interface
IUT	implementation under test
NIST	National Institute of Standards and Technology
PCD.1	POSIX Conforming Document for POSIX.1
PCTP	POSIX Conforming Test Procedure
PCTS	POSIX Conformance Test Suite
POSIX	The colloquial name for the collection of IEEE 1003 Standards, the first of which was IEEE Standard Portable Operating System Interface for Computer Environments, IEEE Std 1003.1-1988.
POSIX.1	ISO/IEC 9945-1:1990 (IEEE Std 1003.1-1990)
POSIX.3.1	IEEE Std 2003.1-1992
SUT	system under test

3. FIPS 151-2

3.1 FIPS 151-2 and POSIX.1

FIPS 151-2 adopts ISO/IEC 9945-1:1990 and in addition requires a number of the optional features defined in POSIX.1 to promote application portability among conforming implementations. An implementation that conforms to FIPS 151-2 also conforms to POSIX.1.

3.2 Chronicle

FIPS 151

- FIPS 151 approved, September 12, 1988.

FIPS 151-1

- FIPS 151-1 approved, March 28, 1992.
- NIST POSIX Testing Program for FIPS 151-1 established, May 1, 1991.

FIPS 151-2

- FIPS 151-2 approved, April 15, 1993.
- FIPS 151-2 issued, May 12, 1993.
- NIST POSIX Testing Program for FIPS 151-2 established, May 12, 1993.
- FIPS 151-2 effective, October 15, 1993.

4. NIST-PCTS:151-2

4.1 Availability

The NIST-PCTS:151-2 is designed to test conformance to FIPS 151-2 based on the test method specifications of POSIX.3.1-1992 and the additional specific requirements ‘‘a - p’’ of FIPS 151-2. The NIST-PCTS:151-2 only tests the POSIX.3.1 assertions that are applicable to FIPS 151-2.

The Official NIST-PCTS:151-2 is distributed by NIST/CSL. NIST/CSL also provides to NIST-PCTS:151-2 owners, which have e-mail accessibility, any additional updates which occur.

4.2 Testing Resources

The following resources are required to perform FIPS 151-2 conformance testing:

- IUT
- Implementation’s PCD.1 (allows the tester to accurately configure the implementation).
- POSIX.3.1 to provide a reference for the PCD.1.
- Method to load NIST-PCTS:151-2.
- Disk space to load, compile, and run the PCTS. Amount of space needed will vary from one system to another (120MB was required to install and execute on an IUT). Required disk space can be considerably reduced by compiling and runing the PCTS a section at a time.
- FIPS 160 conforming compiler.
- Time to install and execute the PCTS. For estimating time required, installation required less than three hours and execution less than two hours on a 25 SPECmark system.

The following resources are optional, but when provided by the IUT must be tested:

- Mountable file system for performing [EROFS] and [ENOSPC] testing.
- Two general interface devices interfaced to terminal ports connected closed-loop for testing assertions related to the GTI.

4.3 Reports Provided

Four reports are provided by the NIST-PCTS:151-2.

- Installation Report:
  Captures all the commands that are used to generate the utilities and test modules and all output results reported by the implementations utilities used.
- Raw Journal Report:
  Provides a step-by-step log of what test is to be performed, the preparations taken, and the results of each test.
- Output Report:
  The Output Report consists of a test result code for every assertion of each element (see *NIST-PCTS:151-2 — Installation and Testing Guide*, Appendix F).

- Validation Report:
  Compares the result codes in the Output Report with the expected test result codes and lists all discrepancies.

5. Conformance Testing

5.1 Requirements

The current Official NIST-PCTS:151-2 must be used whenever conformance testing for FIPS 151-2 is to be performed (§6.4.2). The Official NIST-PCTS:151-2 is the latest NIST-PCTS:151-2 distributed in conjunction with its latest distributed FIXES Patch. The purchaser may utilize the NIST-PCTS:151-2 as they wish, modifying or enhancing it to provide added value. However, when a test for conformance is to be performed the UNEDITED version of the Official NIST-PCTS:151-2 must be used.

5.2 NIST-PCTS:151-2

The *NIST-PCTS:151-2 — Installation and Testing Guide* provides the documentation required to install, configure and execute the NIST-PCTS:151-2. The tester must use the implementation's PCD.1 to configure this PCTS. PCTS configuration files will reference specific sections of the PCD.1 when information on the behavior of the SUT is needed.

The *NIST-PCTS:151-2 — Installation and Testing Guide* provides the FIPS 151-2 acceptable test result code for each POSIX.3.1 assertion. The Output Journal test result codes must match the FIPS 151-2 required test result codes. If report analysis verifies the implementation is non-conforming, these errors must be corrected in the SUT and the NIST-PCTS:151-2 rerun in its entirety. If report analysis shows that an installation file was not configured correctly, the configuration must be corrected and the NIST-PCTS:151-2 must be rerun in its entirety.

5.2.1 Reports

The Installation Report is generated by the NIST-PCTS:151-2 in the user's test directory with a filename of *journal*.

The Raw Journal Report provides sufficient detail such that most failures can be identified with a specific line or block of the actual source code where the test failed. This report is produced for each element tested and is placed in *./STD/POSIX.1_Section/adm/.element* (where: *element* is the name of the *element* tested).

The Output Report provides the PCTS generated test result codes. The complete list of these codes are listed below as acceptable or unacceptable test result codes.

The acceptable test result codes are:

PASS — Successful test of a base assertion.

PASS_EXTENDED — Successful test of an extended assertion.

UNTESTED — Extended assertion for which a test has not been written.

UNSUPPORTED — Conditional feature not implemented.

NOT_TESTABLE — POSIX.3.1 assertions containing a testing constraint which is not supported by the IUT. Examples illustrating when this test result code is reported are:

- GTI devices are not provided.
- {PCTS_NAME_MAX} ≤ {NAME_MAX}.

NOT_APPLICABLE — POSIX.3.1 assertion contains a conditional feature which is contradictory to the requirements of FIPS 151-2. Eamples illustrating when this test result code is reported are:

- {_POSIX_JOB_CONTROL} is not supported.
- {_POSIX_SAVED_IDS} is not supported.

UNUSED — This assertion test was changed by a POSIX.3.1 interpretation and the assertion is no longer valid.

Unacceptable test result codes must be resolved to an acceptable test code or an APTL resolved test code before a successful conformance statement can be issued. The unacceptable test result codes are:

FAIL — Unsuccessful test of a base assertion.

FAIL_EXTENDED — Unsuccessful test of an extended assertion.

UNRESOLVED — Assertion test could not be completed. (Refer to Raw Journal Report for details on why this test result code was generated.)

NOT_INITIATED — No test was initiated for this assertion number. This code results when an earlier failure causes this assertion test to be skipped.

The Validation Report results are directed to *stdout*.

5.3 FIPS 151-2 **Conformance Document Audit**

The requirement for an audit of the PCD.1 for FIPS 151-2 validation was dropped (see 7.6).

The PCD.1, for the SUT, must provide the documentation requirements of POSIX.3.1 and the additional requirement ‘‘p’’ of FIPS 151-2.

5.3.1 **Additional** FIPS 151-2 **Documentation Requirements**

The APTL must ensure that the additional documentation requirement as specified in FIPS 151-2 Item ‘‘p’’ is met.

> Implementations claiming conformance to FIPS 151-2 shall document, in the POSIX Conformance Document, the FIPS 151-2 conditional features implemented. (The term conditional features are the features or behaviors referred to in FIPS 151-2 that need not be present on all conforming implementations. IEEE Std 2003.1-1992 lists the documentation assertions for POSIX.1)

This requirement provides two necessary features:

- Ensures conditional features supported by the SUT are documented.
- Ensures that the information needed to set up the NIST-PCTS:151-2 is present in the SUT’s PCD.1.

5.3.2 Example of a PCD.1 Audit Procedure

When the implementation Certified by the APTL provides additional text based on an updated ISO/IEC 9945-1:1990 standard or to an approved IEEE standard that updates ISO/IEC 9945-1:1990, Section 7.5 provides the PCD.1 requirements for FIPS 151-2.

To provide some guidance for PCD.1 audits based on POSIX.3.1, a checklist is provided (see Appendix A). Each entry of the checklist corresponds to a POSIX.3.1 (sub)clause and documentation assertion number. This information must be presented in the PCD.1 in the (sub)clause as stated by the checklist.

The *Allowable Resolution*, of the checklist for each entry, is determined by the category of the assertion, the text of the assertion, the text of POSIX.1, and the additional requirements of FIPS 151-2.[1] The *Audit* column of Appendix A is completed by the APTL and must contain, for each entry, a ’d’, ’D’, or ’N’. Which identifier is used for an Audit entry is determined by the following rules.

d Specifies documentation is provided that states the conditional feature is NOT supported. The use of ’d’ is restricted to those conditional documentation assertion entries where the NIST-PCTS:151-2 needs to know when support for the feature documented is provided.

D Documentation is provided.

N Documentation is NOT provided.

The following rules provide the rationale on the *Allowable Resolution* column for each entry.

- A documentation assertion with an ’A’ classification requires the PCD.1 provide documentation. Therefore, the *Allowable Resolution* entry and the *Audit* entry is ’D’.

- A documentation assertion with a ’C’ classification that specifies whether the feature is supported and this feature is required by FIPS 151-2 need not be documented. The *Allowable Resolution* entry for this type of assertion is depicted as RD,N. This syntax specifies the feature is required to be supported by FIPS 151-2 and documentation need not be provided. Therefore, the *Audit* entry is either ’D’ or ’N’.

- A documentation assertion with a ’C’ classification that is implied by a FIPS 151-2 additional requirement must be documented. The *Allowable Resolution* entry for this type of assertion is depicted as RD. This syntax specifies the feature is required to be supported by FIPS 151-2 and documentation is required, since the documentation is an integral part of the support for the FIPS 151-2 additional requirement. The *Audit* entry is ’D’.

- A documentation assertion with a ’C’ classification that specifies that a feature is supported, which FIPS 151-2 does not mandate, and which the implementation supports, must be documented[2]. The *Allowable Resolution* entry for this type of assertion is depicted as either S→D,N or S→D,N,d. The initial syntax specifies that when the feature is supported it must be documented and when unsupported it is not documented. The later syntax is used when documentation for nonsupport is also acceptable to the documentation assertion. The *Audit* entry is either a ’D’ or an ’N’ and when applicable can be a ’d’.

- A documentation assertion with a ’C’ classification that is based on additional specifications, behavior, or documentation being provided need not be documented. The *Allowable Resolution* entry and the *Audit* entry for this type of assertion is depicted as D,N. The *Audit* entry is either ’D’ or ’N’.

- A documentation assertion which is not applicable to FIPS 151-2 requirements must not be documented. The *Allowable Resolution* entry for this type of assertion is depicted as NA→N. The *Audit* entry is ’N’.

The parameters specified by the APTL for the configuration file <config> in the NIST-PCTS:151-2 must be consistent with the specifications of the PCD.1.

[1] The assumption used in the creation of this checklist is that the IEEE Std 2003.1-1992 has properly specified all the documentation assertions. Errors in documentation assertions must be rectified via the IEEE Interpretation process.

[2] FIPS 151-2 specifications ’Item p’ requires that conditional features implemented be documented in the POSIX Conformance Document.

5.4 FIPS 151-2 **Conformance**

A claim of conformance to FIPS 151-2 asserts that the SUT test result codes match the required FIPS 151-2 test result codes (or possible in some cases the APTL resolved test codes) and complies fully with the FIPS 151-2 documentation requirements. Statements claiming conformance to FIPS 151-2 should state the version of the Official NIST-PCTS:151-2 that was used.

A *Certificate of Validation* for FIPS 151-2 is issued only by NIST/CSL through APTLs.

6. NIST/CSL *Certificate of Validation*

This section specifies the procedures to be followed, when a *Certificate of Validation* from NIST/CSL is desired.

6.1 APTL for FIPS 151-2

An Accredited POSIX Testing Laboratory (APTL) is a testing laboratory that has been accredited for POSIX testing by NVLAP. APTLs for FIPS 151-2 have demonstrated the proficiency required to use the NIST-PCTS:151-2 and may submit test results for a FIPS 151-2 *Certificate of Validation*.

Arrangements for conformance testing are between clients and APTLs. Arrangements for evaluation of the test results, for the purpose of issuing a *Certificate of Validation*, are between the APTLs and NIST/CSL.

The Official NIST-PCTS:151-2 is provided by the APTL.

The Official NIST-PCTS:151-2 must be used whenever the goal of the testing is to obtain a *Certificate of Validation* with the test results. The APTL will cause the generation of all required output reports for the IUT. NIST/CSL will accept FIPS 151-2 Certification Reports only from APTLS. APTLs shall not use results from BETA Patches in certification reports submitted to NIST/CSL for validation. An APTL may reference the BETA Patch as the required documentation to support the argument that a bug exists in the NIST-PCTS:151-2 for a specific assertion.

Test results submitted to NIST/CSL for validation must be the results of testing conducted by the APTL after it formally received notification of accreditation from NVLAP (see *NVLAP Handbook*, Appendix E). NIST/CSL schedule reviews of certification reports in the order in which they are received from APTLs. NIST/CSL issues a *Certificate of Validation* upon successful completion of the validation procedure.

6.2 APTL Resolved Test Codes

APTL resolved test codes allow a test result code of an unacceptable type to be resolved to one that is acceptable for obtaining a *Certificate of Validation* from NIST/CSL. To obtain a *Certificate of Validation* when the test result codes reported do not match the acceptable FIPS 151-2 test result codes and the APTL deems that the reason for the discrepancy is not the fault of the SUT, the APTL must resolve each discrepancy to an APTL resolved test result code. Supporting documentation must be provided to justify the selected APTL resolved test code. This policy is not designed to *waive* or ignore actual errors, but rather to permit acceptance of implementations which meet the intent of the standard, but through some set of errors or discrepancies fails to meet a specific encoding of a test (i.e., resolve false negatives).

When test failures occur due to hardware malfunctioning, the APTL is expected to rerun the NIST-PCTS:151-2 in its entirety on properly functioning hardware. No APTL resolved test code exists to cover the failure of a test due to hardware malfunction. The inherent assumption of a *Certificate of Validation* is that the hardware functioned properly.

APTL provided Certification Reports must consist of unmodified test results.

Each submission of an APTL resolved test code will be examined by NIST/CSL on its individual merits. When NIST/CSL accepts the APTL resolved test code, the actual test result code will not be an impediment to the issuance of a *Certificate of Validation*. If NIST/CSL does not accept the APTL resolved test code, the actual test result code applies and a *Certificate of Validation* will not be issued. An APTL shall

NEVER alter the reports generated by an Official NIST-PCTS:151-2 run that is to be used for requesting a *Certificate of Validation* from NIST/CSL.

The complete list of APTL resolved test codes are defined below:

POSIX.1_EXTENSION:

An APTL resolved test code of POSIX.1_EXTENSION specifies that the SUT provided a POSIX.1 allowable extension that was not handled properly by the NIST-PCTS:151-2 and this extension has been properly documented in the SUT's PCD.1. The APTL; shall provide supportive documentation that clearly and accurately describes the problem with the Official NIST-PCTS:151-2, should suggest corrective code that allows the test to perform successfully on the SUT, and shall quote from the PCD.1 the statements that apply.

PCD_DOCUMENTED:

An APTL resolved test code of PCD_DOCUMENTED states that: the unsuccessful result was caused by the PCTS improperly handling an implementation defined condition, the APTL verified that the required documentation is provided in the appropriate place in the PCD.1, the unsuccessful result reported by the PCTS is consistent with the documentation provided, and supporting documentation for these requirements is provided by the APTL in the Certification Report.

PCTS_FAILURE:

An APTL resolved test code of PCTS_FAILURE specifies the Official NIST-PCTS:151-2 has a "bug" that impeded the SUT from properly performing the required test. For acceptance of an APTL resolved test code of PCTS_FAILURE, the APTL generated documentation required must contain:

- The complete output results from the raw journal for the assertion test or test case that failed.
- A clear and accurate description of the problem encountered.
- Code changes or PCTP required to obtain a successful test result code. For an *extended assertion*, this requirement is optional.
- Affirmation from the APTL that successful results were obtained when the documented changes were performed. For an *extended assertion*, this requirement is optional.

POSIX.1_FAILURE:

An APTL resolved test code of POSIX.1_FAILURE specifies that the ISO/IEC 9945-1:1990 has an error or inconsistency that was resolved by an IEEE interpretation or an official update to ISO/IEC 9945-1:1990. The documentation required to use this code is a complete reference to the interpretation.

POSIX.3_FAILURE:

An APTL resolved test code of POSIX.3_FAILURE specifies that the IEEE Std 2003.1-1992 has an error or inconsistency that was resolved by an IEEE interpretation or an official update to the POSIX.3.1 standard. The documentation required to use this code is a complete reference to the interpretation.

POSIX_VERSION_UPDATE

The value of {_POSIX_VERSION} provided by the implementation corresponds either to an updated version of ISO/IEC 9945-1:1990 or to an approved IEEE standard that updates ISO/IEC 9945-1:1990. POSIX_VERSION_UPDATE is an APTL resolved test code that can be used ONLY for assertion *sym_const*[05].

The value of {_POSIX_VERSION} used by the IUT must be provided with the supporting documentation.

6.3 APTL Report Checklist

When the NIST-PCTS:151-2 reports comply with the requirements below, the APTL is ready to request NIST/CSL validation.

- Installation has been completed and no errors or warnings implying errors are reported in the Installation Report.
- The NIST-PCTS:151-2 procedures have been adhered to as documented in the NIST-PCTS:151-2 Installation Guide.
- All Output Report results match those specified in *NIST-PCTS:151-2 — Installation and Testing Guide* (§12.2) except for any test result codes of FAIL, FAIL_EXTENDED, UNRESOLVED, and NOT_INITIATED each of which has been resolved to an APTL resolved test code of POSIX.1_EXTENSION, PCD_DOCUMENTED, PCTS_FAILURE, POSIX.1_FAILURE, or POSIX.3_FAILURE.
- No APTL resolved test codes are dependent on a yet to be determined IEEE interpretation.

6.4 APTL Certification Report

6.4.1 Certificate of Validation Application Form

The application form provides information that will appear on the *Certificate of Validation* (Appendix C). A sample *Certificate of Validation Application Form* is provided as Appendix B and a template for this form is provided by the NIST-PCTS:151-2 as file *./CERT_data/NPTP_appendixB*. This template must be fully and accurately completed.

6.4.1.1 Validated FIPS 151-2 Product

The APTL must provide the name of the supplier of the software product tested and must accurately identify the product. When a trademark or a registered trademark symbol is required, the APTL will specify the requirement in ASCII text on the *Certificate of Validation Application Form*. If a trademark symbol is expected on the *Certificate of Validation*, the APTL must state this. The *Certificate of Validation* will have the option of the use of a symbol for trademarks.

The identification must include the name of the product and any other product designations needed to uniquely define the tested product. The product identified represents the IUT. This product must contain, if needed, automated procedures for configuration control during installation. Implementations requiring that the IUT be manually *patched*, cannot be identified as required and will not be validated. (i.e., An operating system that requires a type definition of *typedef int ssize t;* be added to */usr/include/sys/types.h* to PASS all NIST-PCTS:151-2 testing is not a conforming implementation. This code must be provided by the operating system during installation.)

Products need not automate configuration details required for providing FIPS 151-2 optional features nor for performing FIPS 151-2 validation testing.

6.4.1.2 Configuration of SUT for FIPS 151-2 Conformance

The APTL must specify the installation procedures required to configure the software product identified in 6.4.1.1 for FIPS 151-2 conformance. This specification is to be retained in file *./CERT_data/config_SUT* and is limited only to those procedures needed to obtain FIPS 151-2 conformance. References to specific sections in the SUT's associated system documentation is acceptable. When the SUT is such that FIPS 151-2 conformance is not dependent on installation procedures, then the APTL shall state:

> Operating system is not dependent on installation procedures for FIPS 151-2 conformance.

as the data for *./CERT_data/config_SUT*.

6.4.1.3 Test Environment

Four types of POSIX.1 implementations have been identified: *native*, *hosted*, *cooperating*, and *cooperating-hosted*. The type of the implementation tested must be stated.

The hardware implementation employed to test the product must be identified. The identification must include the name of the implementation and any other designations needed to uniquely define it (i.e., Name, Model, Release). If the product tested was on a *cooperating* implementation, then both the target and the development implementations must be identified. Other required details are:

- model number (if applicable),
- random access memory available, and
- number of central processing units available.

The complete hardware configuration is required. This includes memory provided, cpu's available, controllers installed, and the number and types of disk drives.

The APTL must also provide the name and supplier of the FIPS 160 conforming compiler and the identification scheme to uniquely identify it (Name, Release, etc).

These specifications should follow the format as shown by the following example.

Implementation Tested
Supplier: POSIX Products, Inc.
Product: PROD_456, Version 1.3, August 1993
PCD: POSIX Conformance Document, August 1993

System Tested
Type: Native
Computer Hardware Supplier: Hardware Inc.
Computer Hardware Product: TRUE_blue, Model Red_White_Blue, 48MB, 1cpu
1 - Graphics box, Model See_It_Quick
Disk Controller: Integrated SCSI controller
1 - 665MB disk drive, Model RWB_1
2 - 1.2GB disk drive, Model RWB_3
Terminal Controller: Never_a_Miss, Model RWB,

C Compiler Supplier: POSIX Products, Inc.
C Compiler Product: PROD C Compiler, Version 1.0, August 1993

NIST/CSL will attempt to automate the generation of the *Certificate of Validation* based on the input provided by the APTL and the Output Report from the NIST-PCTS:151-2. APTLs should make every effort to retain the designated format when preparing the application form file *./CERT_data/NPTP_appendixB*.

6.4.2 Timely Dispatch of Certification Reports

APTLs may submit certification reports from the prior Official NIST-PCTS:151-2 for up to thirty days from the date of the latest release of the Official NIST-PCTS:151-2. Certification reports from a prior version received more than thirty days after the official date of the updated NIST-PCTS:151-2 will not be validated.

For example: An APTL that initiates testing on 1/1/95 with the Official NIST-PCTS:151-2 and continues testing with this PCTS even though an updated Official NIST-PCTS:151-2 is released on 3/1/95 must submit the Certification Report to NIST/CSL on or before 3/31/95. This Certification Report received beyond 3/31/95 will not be accepted. The APTL will have to rerun the testing with the Official NIST-PCTS:151-2.

6.4.3 Validation Review

NIST/CSL will strive to process a problem-free Certification Report and provide to the APTL a *Certificate of Validation* within three weeks after the report and its associated *Certificate of Validation Application Form* are received.

A Certification Report which requires an update to the supporting material provided with the PCTS must be resubmitted in its entirety. If the *Certificate of Validation Application Form* is found to be in error, it must be resubmitted along with the updated Certification Report. NIST/CSL will not retain Certification Reports that require additional updates. These reports will be destroyed. The APTL must adhere to §6.4.2 in order to avoid having to rerun and repay for a *Certificate of Validation*.

When a resolution of a test result code is determined to be dependent on a not-yet-completed IEEE interpretation and/or a follow up *NIST/CSL FIPS 151 Certification Body* decision (see FIPS PUB 29-2), the APTL will be informed of this matter and the report will be destroyed. Once the required procedures are completed and the outcome is now in compliance with the results obtained from the SUT, the APTL can update the report to include the decisions rendered. The APTL must adhere to §6.4.2 in order to avoid having to rerun and repay for a *Certificate of Validation*.

An APTL should if possible avoid testing an implementation, for obtaining a *Certificate of Validation*, when interpretations are known to be required. When this condition arises, the APTL should attempt to get the interpretation problem favorably resolved prior to submitting the Certification Report to NIST/CSL.

6.4.4 Collection and Submission of Certification Report

To request a *Certificate of Validation* for FIPS 151-2 an APTL must provide to NIST/CSL the following items:

1. Certification Fee
2. File *CV_report* which is generated by *./bin/gen_cert_rept*.

The file *CV_report* contains the NIST-PCTS:151-2 configuration files and the required APTL generated reports. The APTL reports are:

1. Completed *Certificate of Validation Application Form* (§Appendix B)
2. A statement on altered **/*/src/*/svcomp* files. When no *svcomp* files have been altered, the statement shall state this fact.
3. Documentation for any APTL resolved test codes used. When no APTL resolved test codes are needed because all tests were successful, the document shall state this fact.
4. Documentation specifying additional installation procedures performed or installation options selected to obtain the environment for testing the implementation for FIPS 151-2 conformance. When system documentation is referenced (e.g., Technical Notes, manual pages, etc.), either a copy of this documentation must be provided or a summary of the referenced document must be contained within this report (file *./CERT_data/config_SUT*). When copies of system documents are provided, these will be returned when validation procedures are completed.

Section 13 in the *NIST-PCTS:151-2 — Installation and Testing Guide* documents the file names to use for the required APTL reports and the procedures available for generating *CV_report*. The file *./NIST-*

PCTS/CERT_data/LIST_FILES lists all the files to be included in the Certification Report. The APTL must as a minimum collect, from the IUT, all the required files via the extended *cpio* format creating utility once testing is completed. Once this data is collected, only the files:

./CERT_data/NPTP_appendixA
./CERT_data/NPTP_appendixB
./CERT_data/alt_svcomp
./CERT_data/resolved_trc
./CERT_data/config_SUT

may be altered by the APTL. The APTL must assure that the status times obtained for the other files collected at the IUT site are not altered in the Certification Report submitted to NIST/CSL.

The Certification Report as obtained from the execution of *./bin/gen_cert_rept* must be submitted to the NIST POSIX Cerification Authority in machine-readable form. The medium must be acceptable to NIST/CSL. E-mail sent to *151-2@swe.ncsl.nist.gov* is the prefered machine-readable medium.

The certification fee and a printed copy of the *Certificate of Validation Application Form* containing the NVLAP approved signatory shall be submitted to:

NIST POSIX Certification Authority
National Institute of Standards and Technology
Bldg 225 Room B266
Gaithersburg, MD 20899

The NIST-PCTS:151-2 provides templates of Appendices A, B, and C in directory *./CERT_data*. Templates B and C must be used when submitting a Certification Report. Certification Reports which improvise on the format of these templates will not be accepted.

The APTL must be able to reproduce paper copies of the Certification Report if requested by the NIST POSIX Certification Authority.

6.5 Certificate of Validation

A *Certificate of Validation* is issued by NIST/CSL based on the Official NIST-PCTS:151-2. Samples of a *Certificate of Validation* are provided in Appendix C. Many of the items on the *Certificate of Validation* are those specified by the APTL in the *Certificate of Validation Application Form*.

Additional *Certificate of Validation* information is available from NIST/CSL on:

- Hardware configuration including the types and models of the controllers used for the devices tested.
- List of assertions (element, assertion number, corresponding test result code, and APTL resolved test result code) for those assertions where the Output Report test result code was FAIL, UNRESOLVED, NOT_INITIATED, or FAIL_EXTENDED.

NIST/CSL maintains a register called the *NIST POSIX Testing Laboratories and Validated Products* which lists current APTLs and POSIX products. As additional products are validated, they are added to the register.

NIST/CSL also provides an e-mail file service. You can receive the most recent information on the NIST POSIX Testing Program by sending e-mail messages to *posix@nist.gov*. An e-mail message with a body consisting of ‘‘send 151-2reg’’ will return the register of validated prodocts for FIPS 151-2. The automatic mail file server will read the message and return the requested document as the body of one or more e-mail messages.

6.5.1 Reference File Number

Each *Certificate of Validation* is assigned a Reference File Number. The Reference File Number is determined by the concatenation of; the text ‘‘151-2’’, a unique three letter mnemonic for each operating system supplier, and a three digit number which starts at 001 for each operating system supplier and incremented for each *Certificate of Validation* issued. A Reference File Number of 151-2DCC001 would be assigned to the initial *Certificate of Validation* for the fictitious ‘‘Data Computer Company’’.

7. Testing Policy Decisions

This section specifies the policy decisions that have been made pertaining to the testing for conformance to FIPS 151-2. These existing policy decisions are subject to change and if additional policy decisions are needed they will be placed in this section.

7.1 FIPS 151-2 Testing

This PCTS is testing FIPS 151-2 not an updated POSIX.1-1990 standard. No testing requirement, as determined by FIPS 151-2 (and its associated IEEE Std 2003.1-1992 testing standard), will be altered because an update of the POSIX.1-1990 standard, deleted or made optional a requirement of POSIX.1-1990.

7.2 POSIX.1 Extensions

The issue of conformance testing for implementations providing POSIX.1 extensions, using the NIST-PCTS:151-2 is a problem area. As products are being prepared, vendors, users, and testers are concerned about how NIST/CSL will resolve conformance testing problems involved by POSIX.1 extensions.

Until we determine and document what action to take for an extensively documented and acceptable POSIX.1 extension, we will not validate systems that require resolved test codes of PCTS_FAILURE based on inadequately specified issues as related to POSIX.1.

7.3 POSIX.1 Unspecified Features

ISO/IEC 9945-1:1990 allows conforming implementations to contain unspecified features. Unspecified features provided in implementations under POSIX.1 umbrellas such as those introduced in Sections 2.2.2.4, 2.3.1, and 2.3.2 - which allow implementation-defined, alternate, or extended behavior while completely omitting the details for this behavior - may cause unacceptable test results when tested.

The NIST-PCTS:151-2 tests conformance to FIPS 151-2 based on the assertions (modified as needed to adhere to the additional requirements of FIPS 151-2) in the latest POSIX.3.1 document available when FIPS 151-2 was approved. Implementations with unacceptable test results, because the implementation incorporated unspecified features when tested, will not be issued a *Certificate of Validation*.

Vendors requiring a *Certificate of Validation* for implementations that provide unspecified features must ensure these features do not cause the PCTS to become errant. APTLs must ensure that the implementation tested is the unaltered implementation identified on the *Certificate of Validation Application Form*. The APTL should capture any installation setup details outside the scope of the PCTS that are needed to ensure the successful run of the PCTS. These details will be needed if a rerun of the conformance testing is required in the future.

7.4 FIPS 151-1 and FIPS 151-2 Certificates of Validation

An implementation that attains a FIPS 151-2 *Certificate of Validation* is conforming to both FIPS 151-2 and FIPS 151-1. The ISO/IEC 9945-1:1990 was defined by P1003.1 as the update to IEEE Std 1003.1-1988 which contained only corrections and additional text to explain the intended meaning of the 1988 standard. The tests of the NIST-PCTS:151-2 therefore, can be used as the tests for FIPS 151-1.

FIPS 151-2 became effective on October 15, 1993. NIST/CSL no longer processes FIPS 151-1 Certification Reports.

The Official NIST-PCTS:151-2 (based on the test methods of IEEE Std 2003.1-1992) is available for purchase from NIST/CSL (see Appendix D). NIST/CSL will validate FIPS 151-2 Certification Reports from FIPS 151-2 NVLAP Accredited POSIX Testing Laboratories.

7.5 PCD.1

To harmonize the PCD.1 requirements of FIPS 151-2 with that of ISO/IEC 9945-1:1990 and its official updates, the FIPS 151-2 PCD.1 requirements are:

- The PCD.1 must meet all the requirements as specified by the actual documentation assertions of IEEE Std 2003.1-1992 and additional requirement ‘‘p’’ of FIPS 151-2.
- The PCD.1 must not delete or change any requirements of FIPS 151-2 and its associated test method standard, IEEE Std 2003.1-1992.
- The PCD.1 must contain all the documentation requirements as specified by both FIPS 151-2 and IEEE Std 2003.1-1992.
- The PCD.1 may contain additional text related to vendor determined requirements for an official update of ISO/IEC 9945-1:1990 or for an approved IEEE standard that updates ISO/IEC 9945-1:1990 when a test method standard is not available to specify the actual requirements of the updated standard. The APTL is not required to certify that the text provided is consistent with the perceived documentation requirements of the updated standard.

7.6 PCD.1 Requirement Dropped

IEEE Interpretations has stated that a conflict exists between the requirements of IEEE Std 1003.1-1990 and the documentation assertions as stated in IEEE Std 2003.1-1992. On August 30, 1994, the audit of the PCD.1 was no longer required to obtain a *Certificate of Validation* for FIPS 151-2 from NIST/CSL.

The two unchanged requirements associated with the PCD.1 for FIPS 151-2 certification will be the documentation of its identification in the *Certificate of Validation Application Form*, and the retention of the PCD.1 as part of the APTL’s official test results file. Specification of the PCD.1 identification is confirmation by the APTL that the PCD.1 was utilized when testing was performed.

APPENDIX A

FIPS 151-2 Documentation Audit

Documentation Audit Tables Symbols:

,	— Separation notation used for listing of acceptable symbols.
S	— POSIX.1 conditional feature is supported.
d	— Documentation provided specifies conditional feature NOT supported.
D	— Documentation provided.
N	— Documentation NOT provided.
NA	— Documentation assertion is not applicable to FIPS 151-2 requirements.
S→D	— If conditional feature is supported, then documentation is required.
RD	— Implied required FIPS 151-2 conditional feature, documentation required.
RD,N	— Required FIPS 151-2 conditional feature, documentation need not be provided.
NA→N	— Documentation shall not be provided.

POSIX.3.1 (Sub)clause	#	Allowable Resolution	PCTS Conditional Feature	Audit D,N,d
1.3.1.1	1	D	POSIX.1 environment	.
1.3.1.2	2	D	name, number, and date of applicable POSIX.1 standard	.
1.3.1.2	3	D,N	international software standards	.
1.3.3	4	NA→N	C-language differences from C Standard	.
1.3.3.2	5	D	version of the C Standard supported	.
1.3.3.3	GA1	NA→N	interface differences from C Standard	.
1.3.3.3	6	NA→N	language binding differences from C Standard	.
2.2.2.4	1	D,d	associating appropriate privileges with a process	.
2.2.2.9	2	RD,N	character special files other than terminal device files	.
2.2.2.9	3	D,N	structures of character special files	.
2.2.2.27	4	RD,N	other file types	.
2.2.2.30	5	D,N	additional criteria to assign process to file group class	.
2.2.2.55	6	D	parent process ID assigned after parent ended	.
2.2.2.57	7	D	interpretation of pathname that begins with two slashes	.
2.2.2.68	8	D,N	resources returned when process terminates	.
2.2.2.69	9	D	restrictions of interfaces on read-only file systems	.
2.2.2.83	10	RD,N	effective GID returned by *getgroups*( )	.
2.3.1	1	RD,N	extended security controls	.
2.3.2	2	RD,N	additional and alternate file access controls	.
2.3.2	3	D,N	disabling alternate mechanism and *chmod*( )	.
2.3.5	4	D,N	additional time-related update specifications	.
2.3.5	5	D,N	additional time-related update specifications	.
2.3.5	6	D,N	updating time-related fields	.
2.4	1	RD,N	additional errors	.
2.4	2	D	reliable detection of [EFAULT]	.
2.4	3	D	maximum file size allowed	.
2.5	1	RD,N	additional type symbols ending in ‘‘_t’’	.
2.6	1	RD,N	other characters used for environment variable names	.
2.7.2	1	RD,N	additional feature test macros	.
2.8.3	1	D	limit for {NGROUPS_MAX} in <limits.h>	.
2.8.4	2	D	run-time invariant values in <limits.h>	.
2.8.5	3	D	changeable values in <limits.h>	.
2.9	1	D	values {_POSIX_JOB_CONTROL}, {_POSIX_SAVED_IDS}	.
2.9	2	RD,N	value of {_POSIX_CHOWN_RESTRICTED} in <unistd.h>	.
2.9	3	RD,N	value of {_POSIX_NO_TRUNC} in <unistd.h>	.
2.9	4	RD,N	value of {_POSIX_VDISABLE} in <unistd.h>	.

POSIX.3.1 (Sub)clause	#	Allowable Resolution	PCTS Conditional Feature	Audit D,N,d
2.9.4	5	RD	{_POSIX_CHOWN_RESTRICTED} value in <unistd.h>	.
2.9.4	6	RD	{_POSIX_NO_TRUNC} value in <unistd.h>	.
2.9.4	7	RD	{_POSIX_VDISABLE} value in <unistd.h>	.
3.1.1.2	1	RD,N	sharing open directory streams between parent and child	.
3.1.1.2	2	D,N	inheritance of process characteristics for *fork*()	.
3.1.1.4	3	RD,N	[ENOMEM] detection for *fork*()	.
3.1.2.2	1	D	*execlp*() or *execvp*() results on search without PATH	.
3.1.2.2	2	D,N	inheritance of process characteristics for *exec* elements	.
3.1.2.2	3	D,N	*st_atime* marked for update when *exec* fails	.
3.1.2.4	4	RD,N	execution of irregular files	.
3.1.2.4	5	RD,N	[ENOMEM] detection *exec*	.
3.2.1.1.2	1	D,N	order reporting status for two or more child processes	.
3.2.1.1.3	2	D,N	additional circumstances for reporting status	.
3.2.1.1.4	3	D,N	status value provided by *wait*() or *waitpid*() is [EINTR]	.
3.2.2.1	1	NA→N	result type for function *_exit*()	.
3.2.2.2	2	D	parent process ID assigned to child of terminated process	.
3.3.1.1	1	NA→N	support of {_POSIX_JOB_CONTROL} signals	.
3.3.1.1	2	RD,N	additional signals supported	.
3.3.1.2	3	D,N	action taken on blocked signal when action is SIG_IGN	.
3.3.1.2	4	D	signal delivered more than once if another signal is pending	.
3.3.1.2	5	D,N	delivery of multiple pending signals to process	.
3.3.1.2	6	D	details outside of POSIX.1 on generating signals	.
3.3.1.3	7	D,N	SIGFPE, SIGILL, or SIGSEGV signal ignored	.
3.3.1.3	8	D,N	action taken for SIGCHLD to SIG_IGN	.
3.3.1.3	9	D,N	action SIGFPE/SIGILL/SIGSEGV not from *kill*()/*raise*()	.
3.3.1.3	10	D,N	caught SIGCHLD signal and unwaited terminated child process	.
3.3.1.3	11	D,N	interrupted unsafe function calls an unsafe function	.
3.3.2.2	1	RD,N	send signals to processes of another user ID	.
3.3.2.2	2	D,N	excluded set of system processes when *pid* is 0	.
3.3.2.2	3	D,N	behavior of *kill*() when *pid* is -1	.
3.3.2.2	4	D,N	excluded set of system processes when *pid* is negative but not -1	.
3.3.2.2	5	D,N	restrictions on sending of non-POSIX.1 signals	.
3.3.3.1.2	1	D,N	object of type *sigset_t* not initialized but pointer to object supplied	.
3.3.3.3.4	1	RD,N	[EINVAL] detection for *sigaddset*()	.
3.3.3.4.4	1	RD,N	[EINVAL] detection for *sigdelset*()	.
3.3.3.5.4	1	RD,N	[EINVAL] detection for *sigismember*()	.
3.3.4.2	1	D,N	action taken by *sigaction*() when established by *signal*()	.
3.3.4.2	2	D,N	setting action to SIG_DFL for signal that cannot be ignored	.
3.3.5.2	1	D,N	SIGFPE, SIGILL, or SIGSEGV generated while blocked	.
3.3.6.4	1	RD,N	error conditions detected for *sigpending*()	.
3.4.3.2	1	D,N	SIGALRM generated during *sleep*() is ignored or blocked	.
3.4.3.2	2	D,N	SIGALRM is blocked from delivery	.
3.4.3.2	3	D,N	SIGALRM is not blocked or ignored	.
3.4.3.2	4	D,N	details (time, action, signal mask) SIGALARM interrupting *sleep*()	.
3.4.3.2	5	D,N	*sleep*() interrupted and *siglongjmp*() or *longjmp*() called	.

POSIX.3.1	#	Allowable	PCTS	Audit
(Sub)clause		Resolution	Conditional Feature	D,N,d
4.2.2.1.2	1	RD	appropriate privileges to change real, effective, saved setuids	.
4.2.2.2.2	1	RD	appropriate privileges to change real and effective group IDs	.
4.2.3.2	1	D,N	array entries and returned value from *getgroups*( )	.
4.2.4.3	1	D,N	return value from *getlogin*( ) possibly overwritten	.
4.2.4.4	2	S→D,N,d	error conditions detected for *getlogin*( )	.
4.3.3.2	1	NA→N	support for *setpgid*( )	.
4.4.1.2	1	D	format of *struct utsname* members	.
4.4.1.4	2	RD,N	error conditions detected for *uname*( )	.
4.5.1.4	1	RD,N,d	error conditions detected for *time*( )	.
4.5.2.3	1	D,N	return value from *times*( ) overflow range of type *clock_t*	.
4.5.2.4	2	RD,N,d	error conditions detected for *times*( )	.
4.6.1.3	1	D,N	return value from *getenv*( ) possibly overwritten	.
4.6.1.4	2	RD,N,d	error conditions detected for *getenv*( )	.
4.7.1.3	1	D,N	return value from *ctermid*( ) possibly overwritten	.
4.7.1.4	2	RD,N,d	error conditions detected for *ctermid*( )	.
4.7.2.1.2	1	D,N	return value from *ttyname*( ) possibly overwritten	.
4.7.2.1.4	2	RD,N,d	error conditions detected for *ttyname*( )	.
4.7.2.2.4	1	RD,N,d	error conditions detected for *isatty*( )	.
4.8.1.2	1	D,N	additional configuration system variables	.
5.1.1	1	D,N	internal format of directories	.
5.1.1	2	D,N	size of array *d_name*	.
5.1.2.1.2	1	D,N	file removed or added after *opendir*( )	.
5.1.2.1.4	2	RD,N	[EMFILE] detection for *opendir*( )	.
5.1.2.1.4	3	RD,N	[ENFILE] detection for *opendir*( )	.
5.1.2.2.2	1	D,N	return value from *readdir*( ) possibly overwritten	.
5.1.2.2.2	2	D,N	*readdir*( ) buffers directory entries per read operation	.
5.1.2.2.2	3	D,N	using directory stream after *exec* type function call	.
5.1.2.2.2	4	D,N	parent and child call *readdir*( )	.
5.1.2.2.4	5	D,N	passing a closed *dirp* argument to *readdir*( )	.
5.1.2.2.4	6	RD,N	[EBADF] detection for *readdir*( )	.
5.1.2.3.1	1	NA→N	result type for *readdir*( )	.
5.1.2.3.2	2	D,N	removing and adding entries to a directory	.
5.1.2.3.2	3	D,N	passing a closed *dirp* argument to *rewinddir*( )	.
5.1.2.4.2	4	D,N	child and parent processes both using *readdir*( ) and *rewinddir*( )	.
5.1.2.4.2	1	D,N	accessibility of object type DIR after call to *closedir*( )	.
5.1.2.4.2	2	D,N	passing a closed *dirp* argument to *closedir*( )	.
5.1.2.4.4	3	RD,N	[EBADF] detection for *closedir*( )	.

POSIX.3.1 (Sub)clause	#	Allowable Resolution	PCTS Conditional Feature	Audit D,N,d
5.2.2.2	1	D,N	behavior of *getcwd*() when *buf* is a NULL pointer	.
5.2.2.3	2	D,N	contents of buffer passed to *getcwd*() after an error	.
5.2.2.4	3	RD,N	[EACCES] detection for *getcwd*()	.
5.3.1.2	1	D,N	call to *open*() on a FIFO with O_RDWR set	.
5.3.1.2	2	D	group ID of new file set to group ID of its directory	.
5.3.1.2	3	D,N	*open*() with O_CREAT and bits other than mode bits are set	.
5.3.1.2	4	D,N	*open*() with O_EXCL and O_CREAT not set	.
5.3.1.2	5	D,N	block or character supporting nonblocking and O_NONBLOCK set	.
5.3.1.2	6	D,N	*path* not a block, character, or FIFO file and supports nonblocking	.
5.3.1.2	7	D	O_TRUNC on file types other than regular, FIFO, or terminal files	.
5.3.1.2	8	D,N	*open*() with O_TRUNC and O_RDONLY set	.
5.3.3.2	1	D	bits other than file permission bits in *umask*() argument	.
5.3.4.2	1	RD,N,d	*link*() across file systems	.
5.3.4.2	2	RD,N	appropriate privileges to link to a directory	.
5.3.4.2	3	RD,N,d	*link*() to directory	.
5.3.4.2	4	RD,N	access permission to access existing file for *link*() to succeed	.
5.4.1.2	1	D	effect of *mkdir*() of bits other than permission bits set in *mode*	.
5.4.1.2	2	D	group ID of new file set to group ID of its directory	.
5.4.2.2	1	D	effect of *mkfifo*() of bits other than permission bits set in *mode*	.
5.4.2.2	2	D	group ID of new file set to group ID of its directory	.
5.5.1.2	1	D	appropriate privileges for unlinking directories	.
5.5.1.2	2	RD,N	support for *unlink*() on directories	.
5.5.1.4	3	RD,N	[EBUSY] detection for *unlink*()	.
5.5.2.2	1	RD,N	*rmdir*(path) succeeds and *path* is root or current directory	.
5.5.2.2	2	RD,N	*rmdir*(path) fails and *path* is root or current directory	.
5.5.2.4	3	RD,N	[EBUSY] detection for *rmdir*()	.
5.5.3.2	1	RD,N,d	requires write permission to rename directory	.
5.5.3.4	2	RD,N	[EBUSY] detection for *rename*()	.
5.6.1	1	D,N	usage of field *st_size* in structure returned by *stat*() and *fstat*()	.
5.6.1.2	2	D,N	ORs bits other than those specified into the *st_mode* field ...	.
5.6.2.1.2	1	D,N	additional/alternate file access control mechanisms cause *stat*() fail	.
5.6.2.2.2	1	D,N	additional/alternate file access control mechanisms cause *fstat*() fail	.
5.6.3.2	1	RD,N	appropriate privileges to override file access control mechanism	.
5.6.3.2	2	D,N	appropriate privileges ... none of the execute bits for *path* are set	.
5.6.3.4	3	RD,N	[EINVAL] detection for *access*()	.
5.6.4.2	1	RD,N	appropriate privileges to change file mode	.
5.6.4.2	2	RD,N	ignore S_ISUID or S_ISGID bits	.
5.6.4.2	3	D	effect of file descriptors for files open at *chmod*() time	.
5.6.5.2	1	RD,N	appropriate privileges to change file owner	.
5.6.5.2	2	RD,N	effect of S_ISUID and S_ISGID bits on the file	.
5.6.5.4	3	RD,N	[EINVAL] detection for *chown*()	.
5.6.6.2	1	D,N	additional members of *utimbuf* structure	.

POSIX.3.1 (Sub)clause	#	Allowable Resolution	PCTS Conditional Feature	Audit D,N,d
5.7.1.1.2	1	D,N	additional configurable pathname variables	.
5.7.1.1.2	2	D,N	association of ... with other than a terminal file	.
5.7.1.1.2	3	D,N	association of ... with file types other than a directory	.
5.7.1.1.2	4	D,N	association of ... with file types other than pipe, FIFO, directory	.
5.7.1.1.4	5	RD,N	[EINVAL] detection for *pathconf*()	.
5.7.1.1.4	6	RD,N	[EACCESS] detection for *pathconf*()	.
5.7.1.1.4	7	RD,N	[ENAMETOOLONG] detection for *pathconf*()	.
5.7.1.1.4	8	RD,N	[ENOENT] detection for *pathconf*()	.
5.7.1.1.4	9	RD,N	[ENOTDIR] detection for *pathconf*()	.
5.7.1.2.2	1	D,N	association of ... with other than a terminal file	.
5.7.1.2.2	2	D,N	association of ... with file types other than a directory	.
5.7.1.2.2	3	D,N	association of ... with file types other than pipe, FIFO, directory	.
5.7.1.2.4	4	RD,N	[EBADF] detection for *fpathconf*()	.
5.7.1.2.4	5	RD,N	[EINVAL] detection for *fpathconf*()	.
6.3.1.2	1	D,N	when *close*() is interrupted by a signal	.
6.4.1.2	1	D,N	value of file offset after *read*() on file not capable of seeking	.
6.4.1.2	2	D	*read*() interrupted after reading some data	.
6.4.1.2	3	D	result of subsequent reads after *read*() returns end-of-file	.
6.4.1.2	4	D	*read*() with value greater than {SSIZE_MAX}	.
6.4.1.4	5	D,N	additional conditions for detecting [EIO] for *read*()	.
6.4.2.2	1	D,N	*write*() when *nbyte* is zero and not a regular file	.
6.4.2.2	2	D,N	value of file offset after *write*() on file not capable of seeking	.
6.4.2.2	3	D	*write*() interrupted after writing some data	.
6.4.2.2	4	D	*write*() with value greater than {SSIZE_MAX}	.
6.4.2.4	5	D,N	additional conditions for detecting [EIO] for *write*()	.
6.5.2.2	1	D,N	additional file status flags for F_SETFL for *fcntl*()	.
6.5.2.2	2	D,N,d	advisory record locking for files other than regular files	.
6.5.2.2	3	D,N	*fcntl*() for file locking when *l_len* is negative	.
6.5.2.4	4	RD,N	[EDEADLK] detected for *fcntl*()	.
6.5.3.2	1	D	behavior of *lseek*() on devices incapable of seeking	.

POSIX.3.1 (Sub)clause	#	Allowable Resolution	PCTS Conditional Feature	Audit D,N,d
7.1	1	D	device types supported by the general terminal interface ...	.
7.1.1.2	2	D,N	terminal has foreground process group when ...	.
7.1.1.3	3	D	session leader without a controlling terminal ...	.
7.1.1.3	4	D,N	close of last file descriptor associated with controlling terminal ...	.
7.1.1.3	5	D,N	session leader can reacquire a controlling terminal after ...	.
7.1.1.3	6	D,N,d	after controlling process terminates access ...	.
7.1.1.3	7	D	how controlling terminal for a session is allocated by session leader	.
7.1.1.5	8	D,N	{MAX_INPUT} limit	.
7.1.1.6	9	D,N	{MAX_CANON} limit	.
7.1.1.7	10	D,N	*read*() response when MIN is greater than {MAX_INPUT}	.
7.1.1.8	11	RD,N,d	buffers write output to terminal device	.
7.1.1.9	12	RD,N,d	START and STOP special characters can be changed	.
7.1.1.9	13	D,N	two or more special characters received which have same value	.
7.1.1.9	14	D,N	single bytes other than ... or multibytes have special meaning	.
7.1.1.10	15	D,N	EOF is returned or [EIO] is detected when modem disconnects	.
7.1.2.1	1	D,N	additional members of *termios* structure	.
7.1.2.2	2	D	break condition for non-asynchronous data transmissions	.
7.1.2.2	3	D	conditions under which START and STOP are transmitted	.
7.1.2.2	4	D	initial input control value after *open*() is specified	.
7.1.2.3	5	D	processing of output data by OPOST	.
7.1.2.3	6	D	initial output control value after *open*() is specified	.
7.1.2.4	7	D,N	non-asynchronous serial connection ignoring hardware modes	.
7.1.2.4	8	D	initial hardware control value after *open*() is specified	.
7.1.2.5	9	D,N	echoing details when no character exists to erase	.
7.1.2.5	10	D,N	details on erasing a line when ECHOK and ICANON are set	.
7.1.2.5	11	D,N,d	details on operation when IEXTEN is set	.
7.1.2.5	12	D	interaction of IEXTEN set with ICANON, ISIG, IXON, or IXOFF	.
7.1.2.5	13	D	initial local control value after *open*() is specified	.
7.1.2.6	14	NA→N	implementation ignores ...	.
7.1.2.6	15	D,N	value of NCCS	.
7.1.2.6	16	D,N,d	character values in *c_cc* indexed by START and STOP are ignored	.
7.1.2.6	17	D	initial values of all control characters	.
7.1.3.1.4	1	RD,N	error detection *cfgetospeed*()	.
7.1.3.2.2	1	RD,N	*cfsetospeed*() details on attempting to set unsupported baud rate	.
7.1.3.2.4	2	RD,N	error conditions detected for *cfsetospeed*()	.
7.1.3.2.4	1	RD,N	error conditions detected for *cfgetispeed*()	.
7.1.3.4.2	1	RD,N	*cfsetispeed*() returns error for unsupported baud rate	.
7.1.3.4.4	2	RD,N	error conditions detected for *cfsetispeed*()	.
7.2.1.1.2	1	RD,N,d	input and output baud rates that differ	.
7.2.2.1.2	1	D	period of time break signal is sent when *tcsendbreak*() ...	.
7.2.2.1.2	2	D,N	details on non-asynchronous data transmission and *tcsendbreak*()	.
7.2.3.2	1	NA→N	support for *tcgetpgrp*()	.
7.2.4.2	1	NA→N	support for *tcsetpgrp*()	.

POSIX.3.1 (Sub)clause	#	Allowable Resolution	PCTS Conditional Feature	Audit D,N,d
8.1	1	NA→N	differences from C Standard	.
8.1.3.2	1	RD,N	other locales other than ‘‘C’’	.
8.1.3.2	2	D,N	string returned from *setlocale*( ) when *locale* is a NULL pointer	.
8.1.3.2	3	D,N	additional categories supported for *setlocale*( )	.
8.1.3.2	4	D	LC_ALL not set or set to the empty string ...	.
8.1.6.1	1	NA→N	result type for *longjmp*( )	.
8.1.7.1	1	NA→N	result type for *clearerr*( )	.
8.1.36.1	1	NA→N	result type for *rewind*( )	.
8.1.40.1	1	NA→N	result type for *setbuf*( )	.
8.1.47.1	1	NA→N	result type for *srand*( )	.
8.1.48.1	1	NA→N	result type for *calloc*( ), *free*( ), *malloc*( ), and *realloc*( )	.
8.1.49.1	1	NA→N	result type for *abort*( )	.
8.1.50.1	1	NA→N	result type for *exit*( )	.
8.1.52.1	1	NA→N	result type for *bsearch*( ) and *qsort*( )	.
8.1.56.1	1	D	details of TZ environment variable beginning with a colon	.
8.1.56.1	2	D,N	meanings of any characters except ...	.
8.1.56.1	3	D,N	behavior of *offset* field of TZ environment variable ...	.
8.2.1.4	1	RD,N	error conditions detected for *fileno*( )	.
8.2.2.2	1	D,N	additional values for the *fdopen*( ) *type* argument	.
8.2.2.4	2	RD,N	error conditions detected for *fdopen*( )	.
8.2.3	1	D,N	details involving two or more handles	.
8.2.3	2	D,N	state of open file description when active handle not accessible ...	.
8.2.3	3	D,N	details on rules when file handles are not followed	.
8.2.3	4	D	conditions when applications will see all input exactly once	.
8.2.3	5	D,N	*ftell*( ) result when stream oppened in append mode ...	.
8.3.1.2.1	1	NA→N	result type for *siglongjmp*( )	.
8.3.2.1	1	NA→N	result type for *tzset*( )	.
8.3.2.2	2	D	TZ absent, default time-zone information used by *tzset*( )	.

POSIX.3.1 (Sub)clause	#	Allowable Resolution	PCTS Conditional Feature	Audit D,N,d
9.1	1	D	interpretation of null initial working directory field in user database	.
9.1	2	D	system default value for null initial user program field	.
9.2.1.1.3	1	D,N	return pointer from *getgrgid*( ) possibly overwritten	.
9.2.1.1.4	2	RD,N	error conditions detected for *getgrgid*( )	.
9.2.1.2.3	1	D,N	return pointer from *getgrnam*( ) possibly overwritten	.
9.2.1.2.4	2	RD,N	error conditions detected for *getgrnam*( )	.
9.2.2.1.3	1	D,N	return pointer from *getpwuid*( ) possibly overwritten	.
9.2.2.1.4	2	RD,N	error conditions detected for *getpwuid*( )	.
9.2.2.2.3	1	D,N	return pointer from *getpwnam*( ) possibly overwritten	.
9.2.2.2.4	2	RD,N	error conditions detected for *getpwnam*( )	.
10.1	1	D	interface to format-creating and format-reading utilities	.
10.1	2	D,N	media format and frames on the media in which the data appears	.
10.1.1	1	D,N	tape contents after two zero-filled end-of-archive indicator blocks	.
10.1.1	2	D,N	encoding used for names outside the portable filename character set	.
10.1.1	3	D	details on procedures for handling input of invalid file names	.
10.1.1	4	D,N	format-reading utility details on mode bits not in POSIX.1	.
10.1.1	5	D,N	*typeflag* field settings of CHARTYPE, BLKTYPE, or FIFOTYPE	.
10.1.1	6	D,N	*devmajor* and *devminor* fields	.
10.1.2.1	1	D,N	values for the *c_dev* and *c_ino* fields	.
10.1.2.1	2	D	character or block special files contained in *c_rdev*	.
10.1.2.2	3	D	details on procedures for handling input of invalid file names	.
10.1.2.4	4	D,N	value of *c_filesize* for special files other than FIFO, directory, trailer	.
10.1.2.4	5	D,N	contents of bytes in last block following “TRAILER!!!”	.
10.1.2.5	6	D,N	store/extract file types other than C_ISDIR, C_ISFIFO, C_ISREG	.
10.1.3	7	D	file to read as next file after end-of-file or end-of-media condition	.

APPENDIX B

Certificate of Validation Application Form[3]

Client

Name: ____________________________________
Address: ____________________________________

Implementation Tested

Supplier: Vendor’s name who supplied the tested operating system
Product: Identification of operating system tested
PCD: POSIX.1 Conformance Document identification

System Tested

Type: Native, Hosted, Cooperating-Hosted, or Cooperating

Native, Host, or Target Computer System:
Supplier: Vendor’s name
Product: Identification RAM CPUs
Disk Controller: Identification
Identification of devices on disk controller
Terminal Controller: Identification
Identification of devices on terminal controller

Host and/or Development Operating System
Supplier: Vendor’s name } *When applicable*
Product: Identification

Development Computer System
Supplier: Vendor’s name } *When applicable*
Product: Identification

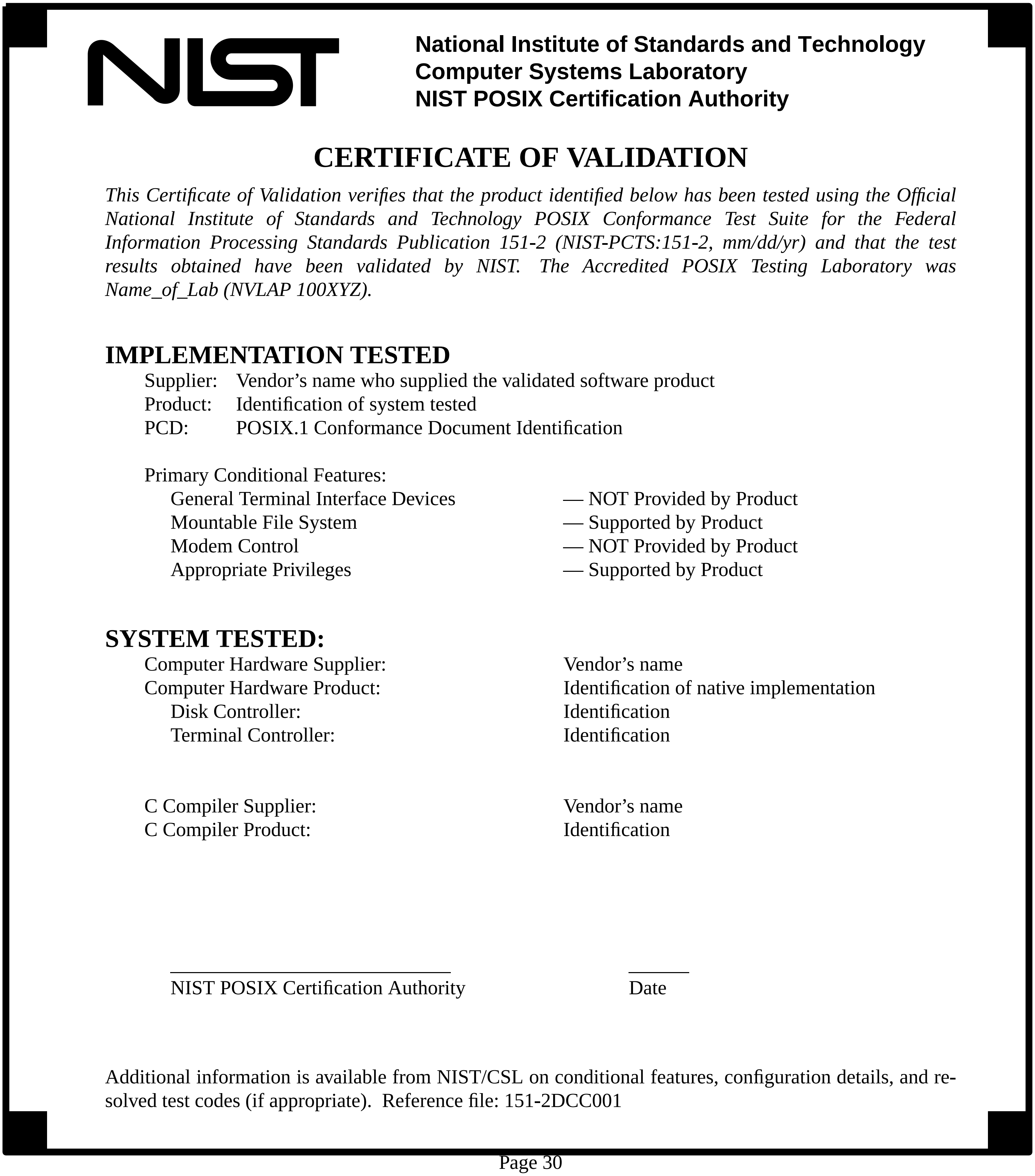
Compiler Information
C Compiler Supplier: Vendor’s name
C Compiler Product: Identification

APTL Name & Number: ____

NVLAP Signatory: ______________________________ Date: ________

[3] This Appendix represents a sample of the actual form. The actual form is provided in *./CERT_data/NPTP_appendixB* of the NIST-PCTS:151-2. The discussion of the items in this form are specified in §6.4.1.* of this document.

APPENDIX C



National Institute of Standards and Technology
Computer Systems Laboratory
NIST POSIX Certification Authority

CERTIFICATE OF VALIDATION

This Certificate of Validation verifies that the product identified below has been tested using the Official National Institute of Standards and Technology POSIX Conformance Test Suite for the Federal Information Processing Standards Publication 151-2 (NIST-PCTS:151-2, mm/dd/yr) and that the test results obtained have been validated by NIST. The Accredited POSIX Testing Laboratory was Name_of_Lab (NVLAP 100XYZ).

IMPLEMENTATION TESTED

Supplier:	Vendor’s name who supplied the validated software product
Product:	Identification of system tested
PCD:	POSIX.1 Conformance Document Identification

Primary Conditional Features:

General Terminal Interface Devices	— NOT Provided by Product
Mountable File System	— Supported by Product
Modem Control	— NOT Provided by Product
Appropriate Privileges	— Supported by Product

SYSTEM TESTED:

Computer Hardware Supplier:	Vendor’s name
Computer Hardware Product:	Identification of native implementation
Disk Controller:	Identification
Terminal Controller:	Identification
C Compiler Supplier:	Vendor’s name
C Compiler Product:	Identification

______________________________ ______
NIST POSIX Certification Authority Date

Additional information is available from NIST/CSL on conditional features, configuration details, and resolved test codes (if appropriate). Reference file: 151-2DCC001

APPENDIX D

Sources of Documents

American National Standards Institute (ANSI), 1430 Broadway, New York, NY 10018, telephone 212/354-3300, order FAX 212/302-1286, information FAX 212/398-0023.

- ISO/IEC 9899:1990 Information Technology—Programming Language — C

Institute of Electrical and Electronics Engineers (IEEE), 345 East 47th Street, New York, NY 10017-2394, telephone 212/705-7900, Standards Orders 800/678-4333.

- International Standard ISO/IEC 9945-1:1990, Information Technology— Portable Operating System Interface (POSIX)— Part 1: System Application Program Interface (API) [C Language]
- IEEE Standard for Information Technology—Test Methods for Measuring Conformance to POSIX, IEEE Std 1003.3-1991
- IEEE Standard for Information Technology—Test Methods for Measuring Conformance to POSIX.1, IEEE Std 2003.1-1992

National Institute of Standards and Technology (NIST), Computer Systems Laboratory, NIST POSIX Certification Authority, Bldg 225 Room B266, Gaithersburg, MD 20899, 301/975-3295, FAX 301/590-0932.

- NIST POSIX Testing Policy— General Information.
- NIST POSIX Testing Policy— Certificate of Validation Requirements for FIPS 151-2.
- NIST-PCTS:151-2 Distribution, contains: (1) NIST POSIX Testing Policy— General Information, (2) NIST POSIX Testing Policy— Certificate of Validation Requirements for FIPS 151-2, (3) *NIST-PCTS:151-2 — Installation and Testing Guide*, and (4) NIST-PCTS:151-2.

National Technical Information Service (NTIS), 5285 Port Royal Road, Springfield, VA 22161, telephone 703/487-4650, FAX 703/321-8547.

- Federal Information Processing Standards Publication 151-2 (FIPS PUB 151-2), Portable Operating System Interface (POSIX)— System Application Program Interface [C Language], 1993 May 12. (Supersedes FIPS PUB 151-1 — 1990, March 28)
- Federal Information Processing Standards Publication 160 (FIPS PUB 160), ’C’, March 13, 1991 with change #1, August 24, 1992.
- NIST/CSL Validated Processor List, NISTIR XXXX (published quarterly), Order number PB91-937300.

National Voluntary Laboratory Accreditation Program (NVLAP), National Institute of Standards and Technology, Bldg 411 Room A124, Gaithersburg, MD 20899, telephone 301/975-4016, FAX 301/975-3839.

- NVLAP Program Handbook— Computer Applications Testing— POSIX Conformance Testing, 1991.

CONTENTS