JTRS Technology Laboratory

Operating Environment (OE) Test Report

for the

SCARI-Open

February 28, 2005

Prepared for:

Joint Tactical Radio System Joint Program Office

1700 N. Moore Street, Suite 1000, Arlington, VA 22209

Prepared by:

JTRS Technology Laboratory

Space and Naval Warfare Systems Center Charleston

P.O. Box 190022, North Charleston, SC 29419-9022

Approved for public release; distribution is unlimited.

Table of Contents

Executive Summary

Purpose

SCARI-Open is the java based SCA Core Framework that was developed by the Communication Research Centre. The goal of this CF is to make a JTeL certified available to community at large. The purpose of the testing is to provide the community a running example with source code that is performing as per specification.

SCA Compliance Testing Results

Core Framework (CF) Testing
< This section provides the Executive Summary of the Core Framework Testing>

CORBA Middleware Assessment
< This section provides the Executive Summary of the ORB assessment >

Operating System (OS) Assessment
< This section provides the Executive Summary of the Operating System assessment >

SCA Certification Recommendation by the JTeL

< This section provides the JTeL recommendations for certification. >

Resolution Plan
< This section discusses a recommended path for resolution of discrepancies and subsequent retesting. >

1 Scope

1.1 Identification

This report details the results of testing the SCARI2-Open. This Operating Environment was developed to meet the requirements of the Modular Software-programmable Radio Consortium Software Communications Architecture (SCA) Specification Version 2.2 [8].

1.1.1 Core Framework Assessment for the SCARI2-Open

This report details the results of testing the SCARI2-Open that was included with the Generic PC Platform. This Core Framework was developed to meet the requirements of the Modular Software-programmable Radio Consortium Software Communications Architecture (SCA) Specification Version 2.2 [8].

1.1.2 Middleware SCA Compliance Assessment for theJava IDL

The Joint Tactical Radio System (JTRS) Joint Program Office (JPO) tasks JTeL to perform SCA compliance testing and assessments on JTRS Operating Environments (OE). There are three areas of OE compliance testing and assessments: compliance of the Operating System (OS), Core Framework (CF), and Middleware Common Object Request Broker Agent (CORBA). Each area is tested or assessed against the requirements of the Software Communication Architecture [8]. The JTeL used the procedures detailed in their MinimumCORBA Specification Procedures, reference (b)***, for this assessment.

This report's purpose is to convey the findings of a JTeL assessment performed on the Java IDL. JTeL conducted this assessment using the requirements specified for the Middleware and Services of the SCA Operating Environment, in section 3.1.2 of reference [8].

This report includes a Summary of the JTeL findings, a Certification Recommendation, an Assessment Summary, the Assessment Details, References, and an annotated SCA v2.2 Middleware & Services Checksheet.

1.1.3 SCA V2.2 AEP POSIX® Compliance Assessment for the Linux Redhat Fedora Core 1

In accordance with JTRS JPO Technical Laboratory's (JTeL) POSIX® OE PSE SCA v2.2 Compliance V1.0 requirements [7], an assessment of the Linux Redhat Fedora Core 1 documentation, as supplied with the SCARI-Open, has been performed.

This assessment includes a Summary of the JTeL findings, Certification Recommendation, Assessment Summary, Assessment Details, References, and an annotated SCA v2.2 AEP POSIX Function Checksheet.

1.2 System Overview

The mission of the JTRS program is to acquire a family of affordable, high capacity, tactical radios to provide interoperable line-of-sight and beyond line-of-sight command, control, communications, computers and intelligence capabilities to the warfighter. To accomplish this goal of joint interoperability, the Department of Defense (DoD) defined a joint program to develop and field software defined radio systems that are based on waveforms realized as software applications. These waveforms will share a common architecture as defined by the SCA [8].

Adherence to the SCA allows standardization of the JTR set waveform application environment and is the basis for waveform portability. Portability of waveforms is one of the primary means of reducing development costs and providing for interoperability.

® POSIX is a registered trademark of the Institute of Electrical and Electronics Engineers, Inc.

Interoperability is achieved through the reuse of common core waveforms used in all JTR sets, and through an expanded Defense Information Systems Agency (DISA) Joint Interoperability Test Command (JITC) waveform certification process mandated for each waveform before operational fielding.

1.2.1 Core Framework Assessment

< This section provides a discussion of the results of the Core Framework testing. A discussion of specific areas of non-compliance is provided here. >

1.2.2 CORBA Middleware Assessment

< This section provides a discussion of the findings of the assessment of the Operating Environment's CORBA middleware. A discussion of specific areas of non-compliance is provided here. >

1.2.3 POSIX Compliance Assessment

< This section provides a discussion of the findings of the assessment of the Operating Environment's Operating System. A discussion of specific areas of non-compliance is provided here. >

The JTeL AEP compliance assessment can only attest to the basic existence of the required functions, arguments, and options by name. No tests are performed to verify the actual functional accuracy or compliance of any individual element.

1.3 Tested Configuration

The configuration consists of two PC: the tester and the target. They are interconnected by Ethernet. The tester is s Satellite 6100 Pro Toshiba laptop running Window XP SP2 OS and it is hosting the JTAP tool. The target is an HP desktop workstation wx3100 running Fedora Core 1 Linux. SCARI-Open is running on the target. Java IDL is the orb used on the target as well as TAO1.3 for the JTAP PseudoComponents (Devices, Resources, Services).

1.4 Document Overview

This test report conforms to the content guidelines of Data Item Description No. DI-IPSC-81440A, Software Test Report (STR) [Reference 1]. It is organized as follows:

Section 1: Scope

This section provides an identification and overview of the system and software to which this document applies.

Section 2: Referenced Documents

This section cites documents referenced by this report.

Section 3: Overview of Test Results

This section provides a summary of the test results, lists the remaining deficiencies, and suggests recommended improvements in the testing.

Section 4: Detailed Test Results

This section provides detailed results including any problems encountered and any deviations from the test procedures.

2 Referenced Documents

[1] Data Item Description No. DI-IPSC-81440A, Software Test Report (STR)
[2] General Unit Test Suite (GUTS) 2.0 Design Guide AX300132-001, 14 August 2003

[3] *Information technology – Portable Operating System Interface (POSIX®) – Part 1: System Application Program Interface (API) [C Language]*, ISO/IEC 9945-1:1996 aka ANSI/IEEE 1003.1, 1996 Edition
[4] JTAP Test Lessons Learned JTRS Step 2C Document No. R257A630 Rev B, 1 October 2003, BAE Systems
[5] JTAP Test Report JTRS Step 2C Document No. R257A629 Rev B, 1 October 2003, BAE Systems
[6] JTRS Technology Laboratory MinimumCORBA Specification Procedures V1.0, December 05, 2003
[7] *POSIX® OE PSE SCA v2.2 Compliance V1.0*, [JTeL-PSE SCA-v1.0], 31 Dec 2003, JTRS Technical Laboratory
[8] *Software Communications Architecture Specification*, MSRC-5000SCA, V2.2, November 17, 2001, Joint Tactical Radio System (JTRS) Joint Program Office
[9] *Software Communications Architecture Specification, Appendix B. SCA Application Environment Profile*, MSRC-5000SCA, Appendix B, rev. 2.2
[10] *Support and Rationale Document for the Software Communications Architecture Specification (V2.2)*, MSRC-5000SCA, V2.2, December 19, 2001, Joint Tactical Radio Office

3 Overview of Test Results

3.1 Overall Assessment of the Software Tested

3.1.1 Results by Test Case

# of Test Cases	% of Total Test Cases	Status
263	87.09%	Passed
24	7.95%	JTAP Issues
0	2.98%	Untestable
9	2.98%	Spec Issues
6	1.99%	Failed

3.1.2 Results by Requirements

# of SCA Requirements	% of Requirements	Status
495	86.39%	Passed
19	3.32%	Failed
40	6.98%	JTAP Issues
6	1.05%	SPEC Issues
12	2.09%	Not Tested

# of BHV Requirements	% of Requirements	Status
107	89.92%	Passed
0	0.00%	Failed
10	8.40%	JTAP Issues
0	0.00%	SPEC Issues
1	0.84%	Not Tested

3.1.3 Remaining Deficiencies

The remaining deficiencies are detailed in section 4.

3.2 Impact of Test Environment

< This paragraph describes any special considerations in the test environment that would affect the test results. >

3.3 Recommended Improvements

< This section discusses any recommended improvements to the tested software or the requirements. >

3.4 Assessment Summary

3.4.1 CORBA Middleware Assessment

< This section provides a summary of the ORB assessment >

3.4.2 POSIX Compliance Assessment

< This section provides a summary of the Operating System assessment >

3.5 Assessment Details

3.5.1 POSIX Compliance Assessment

< This section provides a discussion of specific items of non-compliance with the SCA specification. Discuss the reasons for non-compliance and the impact of these discrepancies >

4 Detailed Test Results

4.1 NamingService Tests

4.1.1 Summary of Test Results

NamingService Tests	Status
NamingService destroy	Passed
NamingService Functions	Passed

4.1.2 Problems Encountered

No problem encountered.

4.1.3 Deviations from Test Cases/Procedures

No Deviations required.

4.2 DTD Verification Tests

4.2.1 Summary of Test Results

DTD Verification Tests	Status
DTD Verification	Passed

4.2.2 Problems Encountered

No problem encountered.

4.2.3 Deviations from Test Cases/Procedures

No Deviations required.

4.3 Domain Manager Tests

4.3.1 Summary of Test Results

Domain Manager Tests	Status

DomainManager applicationFactories Attribute	Passed
DomainManager applications Attribute	Passed
DomainManager configure	Passed
DomainManager configure InvalidConfiguration	Passed
DomainManager configure PartialConfiguration	Passed
DomainManager deviceManagers Attribute	Passed
DomainManager domainManagerProfile Attribute	Passed
DomainManager fileMgr Attribute and Components	Passed
DomainManager identifier Attribute	Passed
DomainManager installApplication ApplicationInstallationError	Passed
DomainManager installApplication InvalidFileName	JTAP issue
DomainManager installApplication InvalidProfile	Passed , JTAP issue (with log)
DomainManager installApplication uninstallApplication	Passed
DomainManager Naming Service Register	JTAP issue
DomainManager PRODUCER_LOG_LEVEL	Passed
DomainManager query	Passed
DomainManager query UnknownProperties	Passed
DomainManager registerDevice DeviceManagerNotRegistered	Passed
DomainManager registerDevice InvalidObjectReference	Passed
DomainManager registerDevice InvalidProfile	Passed
DomainManager registerDevice RegisterError	Passed
DomainManager registerDevice unregisterDevice	Passed
DomainManager registerDeviceManager InvalidObjectReference	Passed
DomainManager registerDeviceManager InvalidProfile	Passed
DomainManager registerDeviceManager RegisterError	Passed
DomainManager registerDeviceManager unregisterDeviceManager	JTAP issue
DomainManager registerService DeviceManagerNotRegistered	Passed
DomainManager registerService InvalidObjectReference	Passed
DomainManager registerService RegisterError	Passed
DomainManager registerService unregisterService	JTAP issue
DomainManager registerWithEventChannel AlreadyConnected	Passed
DomainManager registerWithEventChannel InvalidEventChannelName	Passed
DomainManager registerWithEventChannel InvalidObjectReference	Passed
DomainManager registerWithEventChannel unregisterFromEventChannel	Passed
DomainManager Restore ApplicationFactories, Setup	Passed
DomainManager Restore ApplicationFactories, Test	Passed
DomainManager uninstallApplication ApplicationUninstallationError	JTAP issue
DomainManager uninstallApplication InvalidIdentifier	Passed
DomainManager unregisterDevice InvalidObjectReference	JTAP issue

DomainManager unregisterDevice UnregisterError	Passed
DomainManager unregisterDeviceManager InvalidObjectReference	Passed
DomainManager unregisterDeviceManager UnregisterError	Passed
DomainManager unregisterFromEventChannel InvalidEventChannelName	Passed
DomainManager unregisterFromEventChannel NotConnected	Passed
DomainManager unregisterService InvalidObjectReference	Passed

4.3.2 Problems Encountered

4.3.2.1 DomainManager installApplication InvalidFileName

This test fails on a missing dtd file because it expected an InvalidFileName exception. The same scenario is done for the "DomainManager installApplication InvalidProfile" where an InvalidProfile is expected. The JTAP tool cannot expect different exception thrown for the same scenarion " non existant dtd". Due to the fact that the dtd is part of the xml file and that it is the parser that reports this error, I would be incline to categorize this failure as a InvalidProfile. An xml file (in the context of the SCA) that cannot be validated because of missing DTD should be invalid.

4.3.2.2 DomainManager installApplication InvalidProfile

Minor: The JTAP tool is having trouble matching the word tokens it is looking for even though it is in the log message

4.3.2.3 DomainManager Naming Service Register

Minor: the JTAP is misreading the Naming Service. The DomainManager is listed as expected by the JTAP.

4.3.2.4 DomainManager registerDeviceManager unregisterDeviceManager

JTAP Issue:

- Missing JTAP_SERVICE_1 component cause CreationApplicationError because of inability to establish connection to the service (1 occurrence)
- JTAPDevice error generation because file was unloaded and the refCount is 0 (3 occurrences)
- Missing JTAP_DEVICE_1 component cause CreationApplicationError because of inability to deploy the resource component (1 occurrence)

4.3.2.5 DomainManager registerService unregisterService

JTAP Issue:

- Missing JTAP_SERVICE_1 component cause CreationApplicationError because of inability to establish connection to the service (2 occurrences)
- JTAPDevice error generation because file was unloaded and the refCount is 0 (3 occurrences)
- Expecting ID information in the event object that is not provided during the service registration process.

4.3.2.6 DomainManager uninstallApplication ApplicationUninstallationError

JTAP tries to remove the application twice. Even though it failed the first time the application has been completely uninstalled. This is why an Invalid ID exception is thrown the second time around.

4.3.2.7 DomainManager unregisterDevice InvalidObjectReference

Minor: The JTAP tool is having trouble matching the word tokens it is looking for even though it is in the log message.

4.3.3 Deviations from Test Cases/Procedures

Loading the jtapValidApplication.exe onto the JTAP_DEVICE_1 causes a read error. The load implementation on the JTAP_DEVICE_1 does not do an iterative read (i.e. it attempts to read the whole 1M file which seems to fail). Workaround: create a script that points to the jtapValidApplication.exe on the test machine. Modify SPD accordingly.

4.4 Event Service Tests

4.4.1 Summary of Test Results

EventService Tests	Status
EventService disconnect_push_consumer	Passed
EventService disconnect_push_consumer CORBA::OBJECT_NOT_EXIST	Passed
EventService EventService Created Channel	Passed
EventService IDM_Channel	Passed
EventService ODM_Channel Consumer	Passed
EventService ODM_Channel Producer	Passed
EventService push CosEventComm::Disconnected	Passed

4.4.2 Problems Encountered

4.4.3 Deviations from Test Cases/Procedures

4.4.3.1 EventService disconnect_push_consumer

Had to modified the pd_I.cpp code and revert to old version of events extract since the >>= into a ulong failed which caused the test to fail. It was specified in the JTAP Issues 3.12 para 10.

4.5 File Tests

4.5.1 Summary of Test Results

File Tests	Status
File fileName Attribute	Passed
File filePointer Attribute	Passed
File read write close	Passed
File setFilePointer	Passed
File setFilePointer InvalidFilePointer	Passed
File sizeOf	Passed
File sizeOf FileException	Failed
File write IOException	Passed

4.5.2 Problems Encountered

4.5.2.1 File sizeOf FileException

This exception is not thrown even when it is deleted from the native filesystem.

4.5.3 Deviations from Test Cases/Procedures

4.6 FileSystem Tests

4.6.1 Summary of Test Results

FileSystem Tests	Status
FileSystem copy	JTAP issue
FileSystem copy FileException	Passed
FileSystem copy InvalidFileName	Passed
FileSystem create FileException	SPEC issue
FileSystem create InvalidFileName	SPEC issue
FileSystem create remove	Passed
FileSystem exists	Passed
FileSystem exists InvalidFileName	Passed
FileSystem list	Passed
FileSystem list InvalidFileName	Passed
FileSystem mkdir FileException	Passed
FileSystem mkdir InvalidFileName	Passed
FileSystem mkdir rmdir	JTAP issue
FileSystem open	Passed
FileSystem open FileException	SPEC issue
FileSystem open InvalidFileName	SPEC issue
FileSystem query	Passed
FileSystem query UnknownFileSystemProperties	Passed
FileSystem remove InvalidFileName	Passed
FileSystem rmdir InvalidFileName	Passed
FileSystem rmdir FileException	Passed

4.6.2 Problems Encountered

4.6.2.1 FileSystem copy

JTAP is attempting to copy directory structure. As Per SPEC 3.1.3.3.2.5.2.2 copy a file to another file.

4.6.2.2 FileSystem create FileException

Only non-native CORBA exceptions can provide a return value and exception information from the same CORBA method call. When the native exception is used no return value is provided. The SPEC should be changed to “if a return value is provided then it should be NIL” since it is an ORB-dependant requirement.

4.6.2.3 FileSystem create InvalidFileName

Same as 4.6.2.2

4.6.2.4 FileSystem mkdir rmdir

JTAP is not expecting an exception but it is trying to remove a directory that is not empty.
As per SPEC 3.1.3.3.2.8.3 rmdir should throw an exception if the directory is not empty.

4.6.2.5 FileSystem open FileException

Same as 4.6.2.2

4.6.2.6 FileSystem open InvalidFileName

Same as 4.6.2.2

4.6.3 Deviations from Test Cases/Procedures

4.7 FileManager Tests

4.7.1 Summary of Test Results

FileManager Tests	Status
FileManager copy	JTAP issue
FileManager copy FileException	SPEC issue
FileManager copy InvalidFileName	SPEC issue
FileManager create	Passed
FileManager create FileException	Passed
FileManager create InvalidFileName	Passed
FileManager Distributed FileSystem	Passed
FileManager exists	Passed
FileManager exists InvalidFileName	Passed
FileManager getMounts	Passed
FileManager list	Passed
FileManager list InvalidFileName	JTAP issue
FileManager mkdir	Passed
FileManager mkdir InvalidFileName	Passed
FileManager mkdir rmdir	Passed
FileManager mount InvalidFileName	Passed
FileManager mount InvalidFileSystem	Passed
FileManager mount MountPointAlreadyExists	Passed
FileManager mount unmount	Passed
FileManager open	Passed
FileManager open FileException	Passed
FileManager open InvalidFileName	Passed
FileManager query	Passed
FileManager query UnknownFileSystemProperties	Passed
FileManager remove	Passed
FileManager remove InvalidFileName	Passed
FileManager rmdir	Passed
FileManager rmdir FileException	Passed
FileManager rmdir InvalidFileName	Passed
FileManager unmount NonExistentMount	Passed

4.7.2 Problems Encountered

4.7.2.1 FileManager copy

JTAP is attempting to copy directory structure. As Per SPEC 3.1.3.3.2.5.2.2 copy a file to another file.

4.7.2.2 FileManager create FileException

Only non-native CORBA exceptions can provide a return value and exception information from the same CORBA method call. When the native exception is used no return value is provided. The SPEC should be changed to “if a return value is provided then it should be NIL”.

4.7.2.3 FileManager create InvalidFileName

Same as 4.7.2.1

4.7.2.4 FileManager list InvalidFileName

The JTAP fails to recognize the tab character

4.7.2.5 FileManager open FileException

Same as 4.7.2.1

4.7.2.6 FileManager open InvalidFileName

Same as 4.7.2.1

4.7.3 Deviations from Test Cases/Procedures

4.8 PseudoDevice Device Tests

4.8.1 Summary of Test Results

PseudoDevice Device Tests	**Status**
PseudoDevice adminState Attribute	Passed
PseudoDevice allocateCapacity	Passed
PseudoDevice allocateCapacity InvalidCapacity	Passed
PseudoDevice allocateCapacity InvalidState	Passed
PseudoDevice compositeDevice Attribute	Passed
PseudoDevice configure InvalidConfiguration	Passed
PseudoDevice configure PartialConfiguration	Passed
PseudoDevice configure query	Passed
PseudoDevice deallocateCapacity	Passed
PseudoDevice deallocateCapacity InvalidCapacity	Passed
PseudoDevice deallocateCapacity InvalidState	Passed
PseudoDevice getPort	Passed
PseudoDevice getPort UnknownPort	Passed
PseudoDevice identifier Attribute	Passed
PseudoDevice initialize	Passed
PseudoDevice initialize InitializeError	Passed
PseudoDevice label Attribute	Passed
PseudoDevice operationalState Attribute	Passed
PseudoDevice query UnknownProperties	Passed
PseudoDevice releaseObject	Passed
PseudoDevice releaseObject ReleaseError	Passed
PseudoDevice runTest	Passed
PseudoDevice runTest UnknownProperties	Passed
PseudoDevice runTest UnknownTest	Passed
PseudoDevice softwareProfile Attribute	Passed
PseudoDevice start	Passed
PseudoDevice start StartError	Passed
PseudoDevice stop	Passed

PseudoDevice stop StopError	Passed
PseudoDevice usageState Attribute	Passed

4.8.2 Problems Encountered

No problem encountered.

4.8.3 Deviations from Test Cases/Procedures

No Deviations required.

4.9 PseudoDevice LoadableDevice Tests

4.9.1 Summary of Test Results

PseudoDevice LoadableDevice Tests	**Status**
PseudoDevice load	Passed
PseudoDevice load InvalidFileName	Passed
PseudoDevice load InvalidLoadKind	Passed
PseudoDevice load InvalidState	Passed
PseudoDevice load LoadFail	Passed
PseudoDevice unload	Passed
PseudoDevice unload InvalidFileName	Passed
PseudoDevice unload InvalidState	Passed

4.9.2 Problems Encountered

No problem encountered.

4.9.3 Deviations from Test Cases/Procedures

No Deviations required.

4.10 PseudoDevice ExecutableDevice Tests

4.10.1 Summary of Test Results

PseudoDevice ExecutableDevice Tests	**Status**
PseudoDevice execute	Passed
PseudoDevice execute ExecuteFail	Passed
PseudoDevice execute InvalidFileName	Passed
PseudoDevice execute InvalidFunction	Passed
PseudoDevice execute InvalidOptions	Passed
PseudoDevice execute InvalidParameters	Passed
PseudoDevice execute InvalidState	Passed
PseudoDevice terminate	Passed
PseudoDevice terminate InvalidProcess	Passed
PseudoDevice terminate InvalidState	Passed

4.10.2 Problems Encountered

No problem encountered.

4.10.3 Deviations from Test Cases/Procedures

No Deviations required.

4.11 PseudoDevice AggregateDevice Tests

4.11.1 Summary of Test Results

PseudoDevice AggregateDevice Tests	Status
PseudoDevice AggregateDevice addDevice	Passed
PseudoDevice AggregateDevice addDevice InvalidObjectReference	Passed
PseudoDevice AggregateDevice devices Attribute	Passed
PseudoDevice AggregateDevice removeDevice	Passed
PseudoDevice AggregateDevice removeDevice InvalidObjectReference	Passed

4.11.2 Problems Encountered

No problem encountered.

4.11.3 Deviations from Test Cases/Procedures

No Deviations required.

4.12 PseudoWaveform Resource Tests

4.12.1 Summary of Test Results

PseudoWaveform Resource Tests	Status
PW Resource configure InvalidConfiguration	Passed
PW Resource configure PartialConfiguration	Passed
PW Resource configure query	Passed
PW Resource getPort	Passed
PW Resource getPort UnknownPort	Passed
PW Resource identifier Attribute	Passed
PW Resource initialize	Passed
PW Resource initialize InitializeError	Passed
PW Resource query UnknownProperties	Passed
PW Resource releaseObject	Passed
PW Resource releaseObject ReleaseError	Passed
PW Resource runTest	Passed
PW Resource runTest UnknownProperties	Passed
PW Resource runTest UnknownTest	Passed
PW Resource start	Passed
PW Resource start StartError	Passed
PW Resource stop	Passed
PW Resource stop StopError	Passed

4.12.2 Problems Encountered

No problem encountered.

4.12.3 Deviations from Test Cases/Procedures

The connections in the sad file were converted from deviceusedbythiscomponentref to componentref. The SCARI-Open does not support uses devices therefore the connection cannot be done using the deviceusedbythiscomponentref mechanisms.

4.13 DeviceManager Tests

4.13.1 Summary of Test Results

DeviceManager Tests	Status
DeviceManager configure InvalidConfiguration	Passed
DeviceManager configure PartialConfiguration	Passed
DeviceManager configure Producer LogLevelTypes	Passed
DeviceManager deviceConfigurationProfile Attribute	Passed
DeviceManager Execute Parameters For CompositeDevice	Failed
DeviceManager Execute Parameters For Device	Failed
DeviceManager Execute Parameters For Services	Passed
DeviceManager fileSys Attribute	Passed
DeviceManager getComponentImplementationId	Passed
DeviceManager getPort	Passed
DeviceManager getPort UnknownPort	Passed
DeviceManager identifier Attribute	Passed
DeviceManager label Attribute	Passed
DeviceManager query	Passed
DeviceManager query UnknownProperties	Passed
DeviceManager registerDevice InvalidObjectReference	Passed
DeviceManager registerDevice unregisterDevice	Passed
DeviceManager registeredDevices Attribute	Passed
DeviceManager registeredServices Attribute	Passed
DeviceManager registerService InvalidObjectReference	Passed
DeviceManager registerService unregisterService	Passed
DeviceManager shutdown	Passed
DeviceManager unregisterDevice InvalidObjectReference	Passed
DeviceManager unregisterService InvalidObjectReference	Passed

4.13.2 Problems Encountered

4.13.2.1 DeviceManager Execute Parameters For CompositeDevice

Java JVM does not directly support OS level functions : set stacksize, set priority

4.13.2.2 DeviceManager Execute Parameters For Device

Same as 4.13.2.1

4.13.2.3 DeviceManager shutdown

The JTAP tool complains that the DeviceManager is not waiting for the unregistration. I am not sure how the JTAP figures this out but it seems that when the shutdown terminates all devices have unregistered properly , including the corba return call. The both PseudoDevice show normal termination after their own internal wait. If I force the deviceManager to wait a long time then the test passes.

4.13.3 Deviations from Test Cases/Procedures

Modified the pd_I.cpp source code as per JTAP Issues 2.3.2. section 3.12 para 9

4.14 ApplicationFactory Tests

4.14.1 Summary of Test Results

ApplicationFactory Tests	Status
ApplicationFactory create CreateApplicationError	Passed
ApplicationFactory create CreateApplicationRequestError	Passed
ApplicationFactory create InvalidInitConfiguration	Passed
ApplicationFactory create PseudoWaveform	Failed
ApplicationFactory create with deviceAssignments	Passed
ApplicationFactory identifier Attribute	Passed
ApplicationFactory name Attribute	Passed
ApplicationFactory softwareProfile Attribute	Passed

4.14.2 Problems Encountered

4.14.2.1 ApplicationFactory create PseudoWaveform

Failures:

- SCA158 Failed:
  SCARI-Open does not support UsesDevice.
- SCA152 Failed:
  SCARI-Open is based on java that used a virtual machine that can run on any platform. Since the priority and stacksize is OS specific it is awkward to create a solution for all platform.

JTAP Issues:

- **BHV94 Failed :**
  The create operation will only configure the application's assemblycontroller component. Since all Resource must be configured, all configure items are sent to the assemblycontroller.

  This approach assumes that all property IDs are unique. If two resources built by different parties have the same propID it is impossible for any assembly controller to figure which one is the correct for the specific resource? Therefore it is not a good approach for a CF to bundle all config properties from all the components and feed it through the assembly controller.

  The spec is very elusive about how the application components (i.e. resource) get their initial config values.

- **SCA176 FAILED**
  The create operation shall configure an application's assemblycontroller component provided the assemblycontroller has configure readwrite or writeonly properties with values.

The CF provides all the assembly properties including the ones provided as initConfig in the create call. What the CF does not do is the property bundling of all the resources. This is covered in the explanation of failed BHV94

- **SCA174 UNTESTED**
  That is because of other REQ's like BHV94, SCA 176

- **SCA177 UNTESTED** should be a pass
  The create operation shall use the union of the input initConfiguration properties of the create operation and the assemblycontroller's componentinstantiation writeable "configure" properties that have values.
  The JTAP verified during test but never set it for passing req.

- **SCA179 FAILED**
  This requirement passes as per log. The JTAP test has bundled this req with SCA180. They should be treated separately.

- **BHV102/BHV103 UNTESTED**
  The createResource qualifier parameters shall only contain ResourceFactory factoryparam properties. This interpretation is not well supported in the SCA. BHV102 refers to req D.4.1.1.6 para 5 which state that the factoryparam are used by the ApplicationFactory during the resource creation by resource factory. What it fails to do is to identify the source of the factoryparams.

  BHV103 aka D.6.3.1.2 which describes the precedence order for components that are to be created by resourcefactory. Since only the resource create by Factory has the findcomponent/ componentresourcefactoryref with possible resourcefactoryproperties, it now becomes clear the the factoryparam eluded in D.4.1.1.6 should come from the resource profile NOT the resourceFactory profile.

- **SCA180 FAILED**
  The create operation, when creating a component from a ResourceFactory, shall pass the componentinstantiation componentresoursefactoryref element "factoryparam" properties that have values as qualifier parameters to the referenced ResourceFactory component's createResource operation.

  As per comment on BHV102, there is a conflict in requirements.

- **SCA181 FAILED**
  For connections established for a Log, the create operation shall create a unique producer log ID for each log producer.

  Comments: This requirement is in contradiction with 3.1.2.3.2.1.2 producerId (BHV2) where the unique identifier is the actual component's ID therefore the CF does not have to generate one.

- **SCA183 UNTESTED**
  For connections established for a CORBA Event Service's event channel, the create operation shall connect a COSEventComm PushConsumer or PushSupplier object to the event channel as specified in the SAD's domainfinder element

  Was connected but the JTAP failed to verify.

SPEC Issues:

- SCA182 FAILED
  The create operation shall invoke the PropertySet configure operation once, and only once, per log producer (as described by the SAD usesport element) in order to set its unique PRODUCER_LOG_ID (see section 3.1.3.3.5.5.1.2 for details).

  As per with 3.1.2.3.2.1.2 producerId (BHV2), since the component already receive the unique component id as a required execparam, it becomes redundant to call configure with information the component already has.

4.14.3 Deviations from Test Cases/Procedures

The connections in the sad file were converted from deviceusedbythiscomponentref to componentref. The SCARI-Open does not support uses devices therefore the connection cannot be done using the deviceusedbythiscomponentref mechanisms

4.15 ResourceFactory Tests

4.15.1 Summary of Test Results

ResourceFactory Tests	Status
ResourceFactory createResource CreateResourceFailure	SPEC Issues
ResourceFactory createResource releaseResource	Passed
ResourceFactory identifier Attribute	Passed
ResourceFactory releaseResource InvalidResourceId	Passed
ResourceFactory shutdown	Passed
ResourceFactory shutdown ShutdownFailure	Passed

4.15.2 Problems Encountered

4.15.2.1 ResourceFactory createResource CreateResourceFailure

Only non-native CORBA exceptions can provide a return value and exception information from the same CORBA method call. When the native exception is used no return value is provided. The SPEC should be changed to “if a return value is provided then it should be NIL” since it is an ORB-dependant requirement.

4.15.3 Deviations from Test Cases/Procedures

No Devications required.

4.16 Application Tests

4.16.1 Summary of Test Results

Application Tests	Status
Application componentDevices Attribute	Failed
Application componentImplementations Attribute	Passed
Application componentNamingContexts Attribute	Passed

Application componentProcessIds Attribute	Failed
Application configure InvalidConfiguration	Passed
Application configure PartialConfiguration	Passed
Application configure query	Passed
Application getPort	Passed
Application getPort UnknownPort	Passed
Application initialize	Passed
Application name Attribute	Passed
Application profile Attribute	Passed
Application query empty set	Passed
Application query UnknownProperties	Passed
Application releaseObject	Failed
Application releaseObject ReleaseError	JTAP Issues
Application runTest	Passed
Application runTest UnknownProperties	Passed
Application runTest UnknownTest	Passed
Application start	Passed
Application start StartError	Passed
Application stop	Passed
Application stop StopError	Passed

4.16.2 Problems Encountered

4.16.2.1 Application componentDevices Attribute

The use device will not appear since it is not supported by CRC CF

4.16.2.2 Application componentProcessIds Attribute

Java exec limitation. . The Java Virtual Machine makes abstraction of OS processes, it allows you to create process but hides the low level real pid.. Since the OS pid is not easily available the SCARI-Open ExecutableDevice manages virtual pid that does not match the "real" pid from the OS. The terminate call still work with the virtual pid since java virtual machine knows how to kill the OS process.

4.16.2.3 Application releaseObject

The use device deallocation will not happen since it is not supported by CRC CF

4.16.2.4 Application releaseObject ReleaseError

- Issue 1: The JTAP looks for a error string but does cannot match it even though it is present in the message.

```
Expected message: PW ReleaseError Exception
Received message: [ApplicationFactory:release] : ReleaseError component
DCE:12345678-1000-1000-8000-00A0C9E780D8 error => PW ReleaseError
Exception
```

- Issue 2: The JTAP tools attempts to access the AssemblyController after the first release was called. All components are destroy by then and the Assembly Controller is no longer available which leads to the System Exception.

4.16.3 Deviations from Test Cases/Procedures

The connections in the sad file were converted from deviceusedbythiscomponentref to componentref. The SCARI-Open does not support uses devices therefore the connection cannot be done using the deviceusedbythiscomponentref mechanisms

4.17 Device Tests

4.17.1 Summary of Test Results

Device Tests	Status
Device adminState Attribute	Passed
Device allocateCapacity InvalidCapacity	Passed
Device allocateCapacity InvalidState	Passed
Device compositeDevice Attribute	Passed
Device configure	Passed
Device configure InvalidConfiguration	Passed
Device configure PartialConfiguration	Passed
Device deallocateCapacity InvalidCapacity	Passed
Device deallocateCapacity InvalidState	Passed
Device execute ExecuteFail	SPEC Issues
Device execute InvalidFileName InvalidFunction	SPEC Issues
Device execute InvalidOptions	Passed
Device execute InvalidParameters	SPEC Issues / JTAP Issues
Device execute InvalidState	SPEC Issues / JTAP Issues
Device execute terminate	See Device load execute terminate unload
Device getPort	Passed
Device getPort UnknownPort	Passed
Device identifier Attribute	Passed
Device label Attribute	Passed
Device load InvalidFileName	Passed
Device load InvalidLoadKind	SPEC Issues
Device load InvalidState	Passed
Device load LoadFail	JTAP Issues
Device load execute terminate unload	Failed / JTAP Issues
Device operationalState Attribute	Passed
Device query	Passed
Device query UnknownProperties	Passed
Device releaseObject	Passed
Device runtest UnknownProperties	Passed
Device runTest UnknownTest	Passed
Device softwareProfile Attribute	JTAP Issues
Device stop start	Passed
Device terminate InvalidProcess	Passed
Device terminate InvalidState	Passed
Device unload InvalidFileName	Passed
Device unload InvalidState	Passed
Device usageState Attribute	Passed

4.17.2 Problems Encountered

4.17.2.1 Device execute ExecuteFail

Only non-native CORBA exceptions can provide a return value and exception information from the same CORBA method call. When the native exception is used no return value is provided. The SPEC should be changed to “if a return value is provided then it should be -1” since it is an ORB-dependant requirement.

4.17.2.2 Device execute InvalidFileName InvalidFunction

Only non-native CORBA exceptions can provide a return value and exception information from the same CORBA method call. When the native exception is used no return value is provided. The SPEC should be changed to “if a return value is provided then it should be -1” since it is an ORB-dependant requirement.

4.17.2.3 Device execute InvalidOptions

JTAPIssues:

Seems like and internal error

```
INF: Verify the InvalidOptions Exception is received.
INF: Received InvalidOptions Exception exception as expected.
REQ: PRIMARY_REQUIREMENT: SCA454 PASSED
DBG: Verify the exception invalidOpts properties.
ERR: An exception that occured during test execution was not caught
```

4.17.2.4 Device execute InvalidParameters

SPEC Issues:

Only non-native CORBA exceptions can provide a return value and exception information from the same CORBA method call. When the native exception is used no return value is provided. The SPEC should be changed to “if a return value is provided then it should be -1” since it is an ORB-dependant requirement.

JTAP Issues:

Load , Exec and unload are not symmetrical which causes the unload to fail during the test teardown.

Loading with fullpath filename and call execute and unload with the just the filename

```
DBG: Calling load with
DBG:   fileName: /target2_2/JTAPLoadAndExecute/JTAPLoadAndExecute
DBG:   loadKind: EXECUTABLE

DBG: Calling execute with
DBG: functionName: JTAPLoadAndExecute

DBG: Calling unload with:
DBG:   fileName: JTAPLoadAndExecute
```

4.17.2.5 Device execute InvalidState

SPEC Issues:

Only non-native CORBA exceptions can provide a return value and exception information from the same CORBA method call. When the native exception is used no return value is provided. The SPEC should be changed to “if a return value is provided then it should be -1” since it is an ORB-dependant requirement.

JTAP Issues:

Load , Exec and unload are not symmetrical which causes the unload to fail during the test teardown.

Loading with fullpath filename and call execute and unload with the just the filename

```
DBG: Calling load with
DBG:   fileName: /target2_2/JTAPLoadAndExecute/JTAPLoadAndExecute
DBG:   loadKind: EXECUTABLE

DBG: Calling execute with
DBG: functionName: JTAPLoadAndExecute
```

```
DBG: Calling unload with:
DBG:   fileName: JTAPLoadAndExecute
```

4.17.2.6 Device load execute terminate unload

Failed:

The SCARI-Open accepts the option values but cannot use them because of the JVM limitation. This caused 4 requirements to fail SCA442, SCA443, SCA445, SCA448. SCA442 and SCA443 should pass since the Executable Device accepts the options. SCA445 and SCA 448 overlap but they both failed due to JVM limitation.

JTAP Issues:

Issue 1:

ERR: The process ID returned by execute (0x00000002) is not the actual OS process ID (0x000071a1)
REQ: PRIMARY_REQUIREMENT: SCA449 FAILED

This requirement as stated in the specification should pass. A unique pid is returned. The fact that the pid return is not the same as the OS pid is irrelevant. The JVM allows you to create process but hides the low level real pid.

Issue 2:

Load , Exec and unload are not symmetrical which causes the unload to fail during the test teardown.

Loading with fullpath filename and call execute and unload with the just the filename

```
DBG: Calling load with
DBG:   fileName: /target2_2/JTAPLoadAndExecute/JTAPLoadAndExecute
DBG:   loadKind: EXECUTABLE

DBG: Calling execute with
DBG: functionName: JTAPLoadAndExecute

DBG: Calling unload with:
DBG:   fileName: JTAPLoadAndExecute
```

4.17.2.7 Device load InvalidLoadKind

SPEC Issue:

Java IDL does not allow InvalidLoadKind by construction since the skelton cannot map it to a defined enum type. Therefore it throws bad param.

DBG: loadKind: INVALID
INF: Caught CORBA SystemException (BAD_PARAM): Minor code: 0; Completed status: No.
INF: Verify the InvalidLoadKind Exception exception is received.
INF: Received CORBA::SystemException when expecting InvalidLoadKind Exception.
REQ: PRIMARY_REQUIREMENT: SCA430 FAILED

4.17.2.8 Device load LoadFail

JTAP Issue:

The JTAP FS complains that there are too many files opened by throwing a SCA FileException

4.17.2.9 Device softwareProfile Attribute

JTAP Issue:

The JTAP truncates the ExcutableDevice.spd.xml too much. It prevents to find a match in the DCD

```
DBG: The XML passed to the XML utility will be validated against
deviceconfiguration.2.2.dtd.
DBG: Looking for a fileref that refers to the file xecutableDevice.spd.xml
ERR: The DeviceManager's DCD file does not contain a reference to a DPD file
for this device.
```

4.17.3 Deviations from Test Cases/Procedures

No deviation required

4.18 Port Tests

4.18.1 Summary of Test Results

Port Tests	**Status**
Port connectPort disconnectPort PseudoDevice	Passed
Port connectPort disconnectPort PseudoWaveform	Passed
Port connectPort InvalidPort PseudoDevice	Passed
Port connectPort InvalidPort PseudoWaveform	Passed
Port connectPort OccupiedPort PseudoDevice	Passed
Port connectPort OccupiedPort PseudoWaveform	Passed
Port disconnectPort InvalidPort PseudoDevice	Passed
Port disconnectPort InvalidPort PseudoWaveform	Passed
Port PortType	Passed

4.18.2 Problems Encountered

4.18.2.1 Port PortType

Issue 1:

There is a 10000 loop in the test that gives you the impression that the test is frozen.

Issue 2:

The WString subtest makes a non const copy of the tempString which will lead to a segmentation fault later when the method exits and attempts to release the sequence_var. The second delete on the same location cause a segmentation fault.

Modify PseudoWaveform_TestPort_i.cpp to make a copy of the data and not take ownership of the tempString pointer

4.18.3 Deviations from Test Cases/Procedures

Modified PseudoWaveform_TestPort_i.cpp to prevent the segfault.

4.19 Log Tests (Optional)

4.19.1 Summary of Test Results

Log Tests	Status
Log clearLog	Passed
Log destroy	Passed
Log getAdministrativeState setAdministrativeState	Passed
Log getAvailabilityStatus	Passed
Log getCurrentSize getNumRecords	Passed
Log getLogFullAction setLogFullAction	JTAP Issues
Log getMaxSize setMaxSize	Passed
Log getOperationalState	Passed
Log getRecordIdFromTime	Passed
Log setMaxSize InvalidParam	Passed
Log writeRecords retrieveById	Passed

4.19.2 Problems Encountered

4.19.2.1 Log getLogFullAction setLogFullAction

JTAP Issue:

The JTAP tool uses the wrong index to look up the message. Wrap test goes as follows:

1. Write first message, goes to index 0
2. Write second message, goes to index 1
3. At this stage the log is full (i.e. next message will force a wrap)
4. Write third message (same size as first message), goes to index 2 and first message is overitten

The JTAP test retrieves the 3rd message and compares it with the first one. It passes but that does not mean much. The test should instead

1. Test should retrieve by id 0 and check if list is empty
2. Then retrieve by index 1 with count 2 and compare the message with the first one

The JTAP test retrieves the 3rd message again and tries to compare it with the second. It fails but should not! It should retrieve log message at index 1 in order for the comparison to match.

4.19.3 Deviations from Test Cases/Procedures

No deviation required

4.20 PseudoDeviceManager Tests (OPTIONAL)

4.20.1 Summary of test results

PseudoDeviceManager Tests	Status
PseudoDeviceManager configure InvalidConfiguration	Not Run
PseudoDeviceManager configure PartialConfiguration	Not Run
PseudoDeviceManager configure query	Not Run
PseudoDeviceManager deviceConfigurationProfile Attribute	Not Run

PseudoDeviceManager fileSys Attribute	Not Run
PseudoDeviceManager getComponentImplementationId	Not Run
PseudoDeviceManager getPort	Not Run
PseudoDeviceManager getPort UnknownPort	Not Run
PseudoDeviceManager identifier Attribute	Not Run
PseudoDeviceManager label Attribute	Not Run
PseudoDeviceManager query UnknownProperties	Not Run
PseudoDeviceManager registerDevice	Not Run
PseudoDeviceManager registerDevice InvalidObjectReference	Not Run
PseudoDeviceManager registeredDevices Attribute	Not Run
PseudoDeviceManager registeredServices Attribute	Not Run
PseudoDeviceManager registerService	Not Run
PseudoDeviceManager registerService InvalidObjectReference	Not Run
PseudoDeviceManager shutdown	Not Run
PseudoDeviceManager unregisterDevice	Not Run
PseudoDeviceManager unregisterDevice InvalidObjectReference	Not Run
PseudoDeviceManager unregisterService	Not Run
PseudoDeviceManager unregisterService InvalidObjectReference	Not Run

4.20.2 Problems encountered

Not Run

4.20.3 Deviations from test cases/procedures

No deviation required

5 Test log

The Test log output is available upon request from jtel-cm@spawar.navy.mil.

6 Notes.

6.1 Appendices

A. SCA Requirements Results
B. SCA V2.2 Middleware and Services Checksheet
C. SCA Middleware Compliance
D. SCA V2.2 AEP POSIX® Function Checksheet

Appendix A SCA Requirements Results

Tag	Status
SCA002	PASSED
SCA004	PASSED
SCA005	PASSED
SCA007	PASSED
SCA009	PASSED
SCA010	PASSED
SCA011	PASSED
SCA012	PASSED
SCA014	PASSED
SCA029	PASSED
SCA030	PASSED
SCA031	PASSED
SCA032	PASSED
SCA033	PASSED
SCA034	PASSED
SCA035	PASSED
SCA036	JTAP Issues
SCA037	PASSED
SCA038	PASSED
SCA039	PASSED
SCA040	PASSED
SCA041	PASSED
SCA042	PASSED
SCA043	PASSED
SCA044	JTAP Issues
SCA045	PASSED
SCA047	PASSED
SCA049	PASSED
SCA050	PASSED
SCA051	JTAP Issues
SCA052	PASSED
SCA053	PASSED
SCA054	PASSED
SCA055	PASSED
SCA056	PASSED

Tag	Status
SCA057	PASSED
SCA058	PASSED
SCA060	PASSED
SCA061	PASSED
SCA062	PASSED
SCA064	PASSED
SCA066	PASSED
SCA067	PASSED
SCA069	PASSED
SCA069 D1	PASSED
SCA069 D2	PASSED
SCA070 D1	PASSED
SCA070 D2	PASSED
SCA071 D1	PASSED
SCA072 D1	PASSED
SCA072 D2	PASSED
SCA073 D1	PASSED
SCA073 D2	PASSED
SCA074 D1	PASSED
SCA074 D2	PASSED
SCA076	PASSED
SCA076 D1	PASSED
SCA078	PASSED

Tag	Status
D1	
SCA078 D2	PASSED
SCA079 D1	PASSED
SCA080 D1	PASSED
SCA081 D1	PASSED
SCA085	PASSED
SCA085 D1	PASSED
SCA086	JTAP Issues
SCA086 D1	PASSED
SCA087	JTAP Issues
SCA087 D1	PASSED
SCA089	PASSED
SCA089 D1	PASSED
SCA089 D2	PASSED
SCA089 D3	PASSED
SCA090	PASSED
SCA090 D1	PASSED
SCA091	PASSED
SCA091 D1	PASSED
SCA091 D2	PASSED
SCA093	PASSED
SCA093	PASSED

Tag	Status
D1	
SCA093 D2	PASSED
SCA094	PASSED
SCA094 D1	PASSED
SCA094 D2	PASSED
SCA095	PASSED
SCA096	PASSED
SCA096 D1	PASSED
SCA096 D2	PASSED
SCA096 D3	PASSED
SCA097	PASSED
SCA098	PASSED
SCA098 D1	PASSED
SCA098 D2	PASSED
SCA099	PASSED
SCA099 D1	PASSED
SCA100	PASSED
SCA100 D1	PASSED
SCA101	PASSED
SCA101 D1	PASSED
SCA101 D2	PASSED
SCA102	NOTTESTED
SCA102 D1	PASSED
SCA103 D1	PASSED
SCA105 D1	PASSED
SCA108 D1	PASSED
SCA110 D1	PASSED
SCA110 D2	PASSED
SCA111 D1	PASSED
SCA112 D1	SPEC Issues
SCA115 D1	PASSED
SCA117 D1	PASSED

Tag	Status
SCA117 D2	PASSED
SCA118 D1	PASSED
SCA119 D1	PASSED
SCA121	PASSED
SCA122	PASSED
SCA123	
SCA124	FAILED
SCA125	FAILED
SCA126	PASSED
SCA127	PASSED
SCA128	PASSED
SCA129	PASSED
SCA131	PASSED
SCA132	PASSED
SCA133	PASSED
SCA134	NOTTESTED
SCA135	NOTTESTED
SCA136	FAILED
SCA138	PASSED
SCA139	PASSED
SCA140	PASSED
SCA141	PASSED
SCA142	PASSED
SCA143	JTAP Issues
SCA144	PASSED
SCA145	PASSED
SCA146	PASSED
SCA147	PASSED
SCA148	PASSED
SCA149	PASSED
SCA150	PASSED
SCA151	PASSED
SCA152	PASSED
SCA153	PASSED
SCA154	PASSED
SCA155	PASSED
SCA156	PASSED
SCA157	PASSED
SCA158	FAILED
SCA159	PASSED
SCA160	PASSED
SCA161	PASSED
SCA162	FAILED
SCA163	PASSED
SCA164	PASSED
SCA165	PASSED
SCA166	PASSED
SCA167	PASSED
SCA168	PASSED
SCA169	PASSED

Tag	Status
SCA170	PASSED
SCA171	PASSED
SCA172	PASSED
SCA173	PASSED
SCA174	NOTTESTED
SCA175	PASSED
SCA176	FAILED
SCA177	NOTTESTED
SCA178	PASSED
SCA179	FAILED
SCA180	FAILED
SCA181	FAILED
SCA182	FAILED
SCA183	JTAP Issues
SCA185	PASSED
SCA186	PASSED
SCA187	PASSED
SCA190	PASSED
SCA191	PASSED
SCA192	PASSED
SCA193	PASSED
SCA194	PASSED
SCA195	PASSED
SCA196	PASSED
SCA197	PASSED
SCA198	PASSED
SCA199	PASSED
SCA200	PASSED
SCA201	PASSED
SCA202	PASSED
SCA203	PASSED
SCA204	PASSED
SCA205	PASSED
SCA206	PASSED
SCA207	PASSED
SCA208	PASSED
SCA209	PASSED
SCA210	PASSED
SCA211	PASSED
SCA212	PASSED
SCA213	PASSED
SCA214	PASSED
SCA215	PASSED
SCA216	JTAP Issues
SCA219	PASSED
SCA220	PASSED
SCA221	PASSED
SCA222	PASSED
SCA223	PASSED
SCA224	PASSED
SCA225	PASSED
SCA226	PASSED
SCA227	JTAP Issues

Tag	Status
SCA228	PASSED
SCA229	PASSED
SCA230	PASSED
SCA231	PASSED
SCA232	PASSED
SCA233	PASSED
SCA234	PASSED
SCA235	PASSED
SCA236	PASSED
SCA237	PASSED
SCA238	PASSED
SCA239	PASSED
SCA240	PASSED
SCA241	PASSED
SCA242	PASSED
SCA243	PASSED
SCA244	PASSED
SCA245	PASSED
SCA246	PASSED
SCA247	PASSED
SCA248	PASSED
SCA249	PASSED
SCA250	PASSED
SCA251	PASSED
SCA252	PASSED
SCA253	PASSED
SCA254	PASSED
SCA255	PASSED
SCA256	PASSED
SCA257	PASSED
SCA258	PASSED
SCA259	PASSED
SCA260	PASSED
SCA261	PASSED
SCA262	PASSED
SCA263	PASSED
SCA264	PASSED
SCA265	PASSED
SCA266	PASSED
SCA267	PASSED
SCA268	PASSED
SCA269	JTAP Issues
SCA270	PASSED
SCA271	PASSED
SCA272	JTAP Issues
SCA273	PASSED
SCA274	JTAP Issues
SCA275	PASSED
SCA276	PASSED
SCA277	PASSED
SCA278	PASSED
SCA279	PASSED
SCA280	PASSED

Tag	Status
SCA281	PASSED
SCA282	PASSED
SCA283	PASSED
SCA284	PASSED
SCA285	PASSED
SCA286	PASSED
SCA287	PASSED
SCA288	PASSED
SCA289	PASSED
SCA290	PASSED
SCA291	PASSED
SCA292	PASSED
SCA293	PASSED
SCA294	PASSED
SCA295	PASSED
SCA296	PASSED
SCA297	PASSED
SCA298	PASSED
SCA299	JTAP Issues
SCA300	PASSED
SCA301	PASSED
SCA303	PASSED
SCA304	PASSED
SCA305	PASSED
SCA306	PASSED
SCA307	PASSED
SCA308	PASSED
SCA309	PASSED
SCA310	PASSED
SCA311	PASSED
SCA312	PASSED
SCA313	PASSED
SCA314	NOTTESTED
SCA315	FAILED
SCA316	PASSED
SCA317	PASSED
SCA318	PASSED
SCA319	PASSED
SCA320	FAILED
SCA321	JTAP Issues
SCA322	JTAP Issues
SCA323	JTAP Issues
SCA324	PASSED
SCA325	PASSED
SCA326	PASSED
SCA327	JTAP Issues
SCA329	PASSED
SCA330	PASSED
SCA331	JTAP Issues
SCA332	JTAP Issues
SCA333	JTAP Issues
SCA334	JTAP Issues
SCA335	JTAP Issues

Tag	Status
SCA336	PASSED
SCA338	PASSED
SCA339	PASSED
SCA340	PASSED
SCA341	PASSED
SCA342	PASSED
SCA343	PASSED
SCA344	PASSED
SCA345	PASSED
SCA345 D1	PASSED
SCA345 D2	PASSED
SCA386	PASSED
SCA386 D1	PASSED
SCA386 D2	PASSED
SCA387	PASSED
SCA387 D1	PASSED
SCA388	PASSED
SCA390	JTAP Issues
SCA391	PASSED
SCA392	PASSED
SCA393	PASSED
SCA394	PASSED
SCA395	PASSED
SCA395 D1	PASSED
SCA395 D2	PASSED
SCA401	PASSED
SCA401 D1	PASSED
SCA401 D2	PASSED
SCA403	PASSED
SCA403 D1	PASSED
SCA403 D2	PASSED
SCA404	PASSED
SCA404 D1	PASSED
SCA404 D2	PASSED
SCA405 D1	PASSED
SCA405 D2	PASSED
SCA408 D1	PASSED

Tag	Status
SCA409	PASSED
SCA409 D1	PASSED
SCA409 D2	PASSED
SCA410	PASSED
SCA410 D1	PASSED
SCA410 D2	PASSED
SCA411 D1	PASSED
SCA411 D2	PASSED
SCA415	PASSED
SCA415 D1	PASSED
SCA415 D2	PASSED
SCA417	PASSED
SCA417 D1	PASSED
SCA417 D2	PASSED
SCA418	NOTTESTED
SCA419	FAILED
SCA420	PASSED
SCA421	NOTTESTED
SCA422	PASSED
SCA424	JTAP Issues
SCA424 D1	PASSED
SCA426	PASSED
SCA426 D1	PASSED
SCA426 D2	PASSED
SCA427	NOTTESTED
SCA428	NOTTESTED
SCA429	PASSED
SCA429 D1	PASSED
SCA429 D2	PASSED
SCA430	SPEC Issues
SCA430 D1	PASSED
SCA431	PASSED
SCA431 D1	PASSED
SCA431 D2	PASSED
SCA432	FAILED

Tag	Status
SCA432 D1	PASSED
SCA433	NOTTESTED
SCA433 D1	PASSED
SCA433 D2	PASSED
SCA434	JTAP Issues
SCA435	PASSED
SCA435 D1	PASSED
SCA435 D2	PASSED
SCA436	PASSED
SCA436 D1	PASSED
SCA436 D2	PASSED
SCA438	PASSED
SCA438 D1	PASSED
SCA442	FAILED
SCA443	FAILED
SCA444	PASSED
SCA444 D1	PASSED
SCA445	PASSED
SCA445 D1	PASSED
SCA445 D2	PASSED
SCA446	PASSED
SCA447	PASSED
SCA448	FAILED
SCA449	SPEC Issues
SCA449 D1	PASSED
SCA450	PASSED
SCA450 D1	PASSED
SCA450 D2	PASSED
SCA451	SPEC Issues
SCA451 D1	PASSED
SCA452	PASSED
SCA452 D1	PASSED
SCA452 D2	PASSED
SCA453	JTAP Issues
SCA453 D1	PASSED

Tag	Status
SCA453 D2	PASSED
SCA454	JTAP Issues
SCA454 D1	PASSED
SCA454 D2	PASSED
SCA455	PASSED
SCA455 D1	PASSED
SCA456	JTAP Issues
SCA456 D1	PASSED
SCA456 D2	PASSED
SCA457	JTAP Issues
SCA457 D1	PASSED
SCA457 D2	PASSED
SCA458	PASSED
SCA458 D1	PASSED
SCA459 D1	PASSED
SCA459 D2	PASSED
SCA460 D1	PASSED
SCA460 D2	PASSED
SCA462 D1	PASSED
SCA462 D2	PASSED
SCA463 D1	PASSED
SCA463 D2	PASSED
SCA465 D1	PASSED
SCA465 D2	PASSED
SCA466	PASSED
SCA467	PASSED
SCA467 D1	NOTTESTED
SCA467 D2	NOTTESTED
SCA468	PASSED
SCA468 D1	NOTTESTED
SCA468 D2	NOTTESTED

Tag	Status
SCA469	PASSED
SCA469 D1	NOTTESTED
SCA469 D2	NOTTESTED
SCA470	PASSED
SCA470 D1	NOTTESTED
SCA470 D2	NOTTESTED
SCA471	PASSED
SCA471 D1	NOTTESTED
SCA471 D2	NOTTESTED
SCA472	PASSED
SCA472 D1	NOTTESTED
SCA472 D2	NOTTESTED
SCA474. 1	PASSED
SCA474. 2	PASSED
SCA474. 5	PASSED
SCA474. 6	PASSED
SCA474. 7	PASSED
SCA475	PASSED
SCA476	NOTTESTED
SCA478. A	PASSED
SCA478. B	PASSED
SCA478. C	PASSED
SCA478. D	PASSED
SCA478. E	PASSED
SCA478. F	PASSED
SCA479	PASSED
SCA480	PASSED
SCA481	FAILED
SCA482	PASSED
SCA483	FAILED
SCA484. A	PASSED
SCA484. B	PASSED

Tag	Status
SCA485	PASSED
SCA485 D1	NOTTESTED
SCA485 D2	NOTTESTED
SCA486	PASSED
SCA487	PASSED
SCA488	PASSED
SCA488 D1	NOTTESTED
SCA488 D2	NOTTESTED
SCA489	PASSED
SCA489 D1	NOTTESTED
SCA489 D2	NOTTESTED
SCA490	PASSED
SCA491	PASSED
SCA492	PASSED
SCA492 D1	NOTTESTED
SCA492 D2	NOTTESTED
SCA493	PASSED
SCA493 D1	NOTTESTED
SCA493 D2	NOTTESTED
SCA494	PASSED
SCA495	PASSED
SCA496	PASSED
SCA496 D1	NOTTESTED
SCA496 D2	NOTTESTED
SCA497	PASSED
SCA497 D1	NOTTESTED
SCA497 D2	NOTTESTED
SCA498	PASSED
SCA499	PASSED
SCA500	PASSED
SCA500 D1	NOTTESTED
SCA500 D2	NOTTESTED
SCA501	JTAP Issues
SCA501 D1	NOTTESTED
SCA502	JTAP Issues
SCA503	JTAP Issues

Tag	Status
SCA504	PASSED
SCA504 D1	NOTTESTED
SCA504 D2	NOTTESTED
SCA504 D3	NOTTESTED
SCA505	JTAP Issues
SCA507	PASSED
SCA509	PASSED
SCA510	PASSED
SCA511	PASSED
SCA512	PASSED
SCA513	PASSED
SCA514	PASSED
SCA516	PASSED
SCA517	PASSED
SCA518	PASSED
SCA519	PASSED
SCA520	PASSED
SCA521	JTAP Issues
SCA523	PASSED
SCA525	PASSED
SCA527	PASSED
SCA528	PASSED
SCA532	PASSED
SCA533	PASSED
SCA534	PASSED
SCA535	JTAP Issues
SCA536	PASSED
SCA537	PASSED
SCA538	PASSED
SCA539	PASSED
SCA540	PASSED
SCA541	PASSED
SCA542	PASSED
SCA543	PASSED
SCA545	JTAP Issues
SCA546	PASSED
SCA547	PASSED
SCA548	PASSED
SCA549	SPEC Issues
SCA550	PASSED
SCA551	PASSED
SCA552	PASSED
SCA555	PASSED
SCA556	PASSED
SCA557	SPEC Issues
SCA558	PASSED
SCA559	PASSED
SCA560	PASSED
SCA561	PASSED
SCA562	PASSED

Tag	Status
SCA563	PASSED
SCA564	PASSED
SCA565	PASSED
SCA566	PASSED
SCA567	PASSED
SCA568	PASSED
SCA569	PASSED

Tag	Status
SCA570	PASSED
SCA573	PASSED
SCA574	PASSED
SCA575	PASSED
SCA576	PASSED
SCA577	PASSED
SCA578	PASSED

Tag	Status
SCA579	PASSED
SCA580	PASSED
SCA581	PASSED
SCA582	PASSED
SCA583	PASSED
SCA584	PASSED

Appendix B SCA v2.2 MIDDLEWARE and SERVICES Checksheet

Form revised 26 Jan 2004

	CORBA Naming Service		√
1	*Bind*		√
2	*Bind_new_context*		√
3	*Unbind*		√
4	*Destroy*		√
5	*Resolve*		√
	Log Service		
6	*Producer_log-level*		√
7	*LogLevelSequence*		√
8	*ProducerLogRecord*		√
9	*LogLevelType*		√
10	*ProducerLogRecordType*		√
11	*LogLevelSequence*		√
12	*InvalidParam*		√
13	*LogTimeType*		√
14	*OperationStateType*		√
15	*AdministrativeStatusType*		√
16	*AvailabilityStatusType*		√
17	*LogFullActionType*		√
18	*RecordIDType*		√
19	*LogRecordType*		√
20	*LogRecordSequence*		√
21	*ProducerLogRecordSequence Type*		√
	Operations		
22	*getMaxSize*		√
23	*setMaxSize*		√
24	*getCurrentSize*		√
25	*getNumRecords*		√
26	*getLogFullAction*		√
27	*setLogFullAction*		√
28	*getAvailabilityStatus*		√

29	*getAdministrativeStatus*		√
30	*setAdminitrativeStatus*		√
31	*getOperationalStatus*		√
32	*writeRecords*		√
33	*getRecordIDFromTime*		√
34	*retrieveById*		√
35	*ClearLog*		√
36	*Destroy*		√
	Event Service and Standard Events		
37	*PushConsumer*		√
38	*PushSupplier*		√
39	*IDM_Channel*		√
40	*ODM_Channel*		√
41	*StateChangeCategoryType*		√
42	*StateChangeType*		√
43	*StateChangeEventType*		√
44	*SourceCategoryType*		√
45	*DomainManagementObjectRemovedEventType*		√
46	*DomainManagementObjectAddedEventType*		√

√ - Function present
? - Inadequate documentation
N - Nonconforming function
X - Function or option not present

Appendix C SCA MIDDLEWARE COMPLIANCE

The purpose of this document is to show that JAVA IDL meets the SCA requirements for middleware compliance.

The relevant requirements may be found in section 3.1.2 of the SCA specification (MSRC-5000SCA V2.2). They are tabulated below:

Note: The Spectrum SCA BSP implements only those hardware specific portions of the Core Framework required to map a generic Core Framework onto Spectrum Hardware. As such, not all SCA requirements are directly applicable to the Spectrum's SCA BSP and this document needs to be read in conjunction with the equivalent compliance document for the generic Core Framework implementation with which the BSP is to be used.

Tag No.	SCA Para. No.	Requirement	Applies to OE Yes/No	Compliance
SCA_3	3.1.2.1	The OE SHALL use middleware that, at a minimum, provides the services and capabilities of minimumCORBA as specified by the OMG Document orbos/98-05-13, May 19, 1998.	Yes	.
SCA_4	3.1.2.2.1	A CORBA Naming Service SHALL be provided in the OE.	Yes	
SCA_5	3.1.2.2.1	A CORBA Naming Service supplied by an OE SHALL support the CosNaming CORBA module and its NamingContext interface's operations: bind, bind_new_context, unbind, destroy, and resolve.	Yes	
SCA_6	3.1.2.2.1	These operations SHALL meet the requirements of OMG Document formal/00-11/01: Interoperable Naming Service Specification.	Yes	
SCA_7	3.1.2.2.1	A Naming Service's NameComponent structure is made up of an id-and-kind pair. The "id" element of each NameComponent is a string value that uniquely identifies a NameComponent. The "kind" element of each NameComponent SHALL	Yes	

		be “” (null string).		
SCA_8	3.1.2.3.1	Log producers SHALL implement a configure property with an ID of “PRODUCER_LOG_LEVEL”. The PRODUCER_LOG_LEVEL configure property provides the ability to “filter” the log message output of a log producer.	Yes	
SCA_9	3.1.2.3.1	The type of this property SHALL be a LogLevelSequence. The configure property LogLevelSequence will contain all log levels that are enabled.	Yes	
SCA_10	3.1.2.3.1	Only the messages that contain an enabled log level SHALL be sent by a log producer to a Log. Log levels that are not in the LogLevelSequence are disabled.	Yes	
SCA_11	3.1.2.3.1	Log producers SHALL use their component identifier in the producerId field of the ProducerLogRecord.	Yes	
SCA_12	3.1.2.3.1	Log producers SHALL operate normally in the case where the connections to a Log are nil or an invalid reference.	Yes	
SCA_13	3.1.2.3.1	Log producers SHALL output only those log records that correspond to enabled LogLevelType values.	Yes	
SCA_14 to SCA_44	3.1.2.3.3	All Log service requirements	No	
SCA_45	3.1.2.4.1	A CORBA Event Service (e.g., OMG’s Event Service) SHALL be provided in the OE.	Yes	

SCA_46	3.1.2.4.1	The CORBA Event Service decouples the communication between consumer and producer objects, where consumer components are unaware of producer components, and vice versa. Consumer components process event data that are produced by producer components. The CORBA Event Service is based upon the Push Model approach where producers push events to consumer. The CORBA Event Service SHALL support Push interfaces (PushConsumer and PushSupplier) of the CosEventComm CORBA module as described in OMG Document formal/01-03-01: Event Service, v1.1. The compilable IDL for the CosEventComm is in the OMG Document formal/01-03-02: Event Service IDL, v1.1.	Yes	
SCA_47	3.1.2.4.1	The CosEventComm CORBA Module is used by consumers for receiving events and by producers for generating events. A component (e.g., Resource, DomainManager, etc.) that consumes events SHALL implement the CosEventComm PushConsumer interface.	Yes	
SCA_48	3.1.2.4.1	A component (e.g., Resource, Device, DomainManager, etc.) that produces events SHALL implement the CosEventComm PushSupplier interface and use the CosEventComm PushConsumer interface for generating the events.	Yes	

SCA_49	3.1.2.4.1	A producer component SHALL handle all cases, without raising any exceptions outside of the producer component, due to the connections to a CosEventComm PushConsumer being nil or an invalid reference. The CORBA Event Service will have the capability to create event channels. An event channel allows multiple suppliers to communicate with multiple consumers asynchronously. An event channel is both a consumer and a producer of events. For Example, event channels can be standard CORBA objects and communication with an event channel is accomplished using standard CORBA requests.	Yes	
SCA_50	3.1.2.4.1	The OE provides two standard event channels: Incoming Domain Management and Outgoing Domain Management. The Incoming Domain Management Channel name SHALL be "IDM_Channel".	No	
SCA_51	3.1.2.4.1	The Outgoing Domain Management Channel name SHALL be "ODM_Channel". The Incoming Domain Management event channel is used by components (e.g., Device state change event) within the domain to generate events that are consumed by domain management functions (e.g., ApplicationFactory, Application, DomainManager, etc.). The Outgoing Domain Management Channel is used by domain clients (e.g., HCI) to receive events (e.g., additions or removals from the domain) generated from domain management functions (e.g., ApplicationFactory, Application, DomainManager, etc.). Besides these two standard event channels, the OE allows other event channels to be set up by application developers.	No	

Appendix D SCA v2.2 AEP POSIX® Function Checksheet

Form revised 26 Jan 2004

	POSIX.1 Option Requirements		**√**
	unistd.h {_POSIX_NO_TRUNC}		
	POSIX_SINGLE_PROCESS Functions		
1	*time.h*	time( ) [1]	√
	POSIX_MULTI_PROCESS Functions		
2	*locale.h*	setlocale( )	√
	POSIX_SIGNALS Functions		
3	*signal.h*	kill( )	√
4	*unistd.h*	pause( )	√
5	*signal.h*	sigaction( )	√
6	*signal.h*	sigaddset( )	√
7	*signal.h*	sigdelset( )	√
8	*signal.h*	sigemptyset( )	√
9	*signal.h*	sigfillset( )	√
10	*signal.h*	sigismember( )	√
11	*signal.h*	sigpending( )	√
12	*signal.h*	sigprocmask( )	√
13	*signal.h*	sigsuspend( )	√
14	*stdlib.h*	abort( )	√
	POSIX_FILE_SYSTEM Functions		
15	*unistd.h*	access( )	√
16	*unistd.h*	chdir( )	√
17	*dirent.h*	closedir( )	√
18	*fcntl.h*	creat( )	√
19	*unistd.h*	fpathconf( )	√
20	*sys/stat.h*	fstat( )	√
21	*unistd.h*	getcwd( )	√
22	*unistd.h*	link( )	√
23	*sys/stat.h*	mkdir( )	√
24	*dirent.h*	opendir( )	√
25	*unistd.h*	pathconf( )	√
26	*dirent.h*	readdir( )	√
27	*stdio.h*	rename( )	√
28	*dirent.h*	rewinddir( )	√
29	*unistd.h*	rmdir( )	√
30	*sys/stat.h*	stat( )	√
31	*unistd.h*	unlink( )	√

32	*utime.h*	utime( )	√
33	*stdio.h*	remove( )	√
34	*stdio.h*	tmpfile( )	√
35	*stdio.h*	tmpnam( )	√
	POSIX_FD_MGMT Functions		
36	*unistd.h*	lseek( )	√
37	*stdio.h*	fseek( )	√
38	*stdio.h*	ftell( )	√
39	*stdio.h*	rewind( )	√
	POSIX_DEVICE_IO Functions		
40	*unistd.h*	close( )	√
41	*fcntl.h*	open( )	√
42	*unistd.h*	read( )	√
43	*unistd.h*	write( )	√
44	*stdio.h*	clearerr( )	√
45	*stdio.h*	fclose( )	√
46	*stdio.h*	fdopen( )	√
47	*stdio.h*	feof( )	√
48	*stdio.h*	ferror( )	√
49	*stdio.h*	fflush( )	√
50	*stdio.h*	fgetc( )	√
51	*stdio.h*	fileno( )	√
52	*stdio.h*	fgets( )	√
53	*stdio.h*	fopen( )	√
54	*stdio.h*	fprintf( )	√
55	*stdio.h*	fputc( )	√
56	*stdio.h*	fputs( )	√
57	*stdio.h*	fread( )	√
58	*stdio.h*	freopen( )	√
59	*stdio.h*	fscanf( )	√
60	*stdio.h*	fwrite( )	√
61	*stdio.h*	getc( )	√
62	*stdio.h*	getchar( )	√
63	*stdio.h*	gets( )	√
64	*stdio.h*	perror( )	√
65	*stdio.h*	printf( )	√
66	*stdio.h*	putc( )	√
67	*stdio.h*	putchar( )	√
68	*stdio.h*	puts( )	√
69	*stdio.h*	scanf( )	√
70	*stdio.h*	setbuf( )	√
71	*stdio.h*	sprintf( )	√
72	*stdio.h*	sscanf( )	√
73	*stdio.h*	ungetc( )	√
	POSIX_C_LANG_SUPPORT Character Handling Functions		

74	*ctype.h*	isalnum( )	√
75	*ctype.h*	isalpha( )	√
76	*ctype.h*	iscntrl( )	√
77	*ctype.h*	isdigit( )	√
78	*ctype.h*	isgraph( )	√
79	*ctype.h*	islower( )	√
80	*ctype.h*	isprint( )	√
81	*ctype.h*	ispunct( )	√
82	*ctype.h*	isspace( )	√
83	*ctype.h*	isupper( )	√
84	*ctype.h*	isxdigit( )	√
85	*ctype.h*	tolower( )	√
86	*ctype.h*	toupper( )	√
	POSIX_C_LANG_SUPPORT Mathematical Functions		
87	*math.h*	acos( )	√
88	*math.h*	asin( )	√
89	*math.h*	atan( )	√
90	*math.h*	atan2( )	√
91	*math.h*	ceil( )	√
92	*math.h*	cos( )	√
93	*math.h*	cosh( )	√
94	*math.h*	exp( )	√
95	*math.h*	fabs( )	√
96	*math.h*	floor( )	√
97	*math.h*	fmod( )	√
98	*math.h*	frexp( )	√
99	*math.h*	ldexp( )	√
100	*math.h*	log( )	√
101	*math.h*	log10( )	√
102	*math.h*	modf( )	√
103	*math.h*	pow( )	√
104	*math.h*	sin( )	√
105	*math.h*	sinh( )	√
106	*math.h*	sqrt( )	√
107	*math.h*	tan( )	√
108	*math.h*	tanh( )	√
	POSIX_C_LANG_SUPPORT Non-Local Jump Functions		
109	*setjmp.h*	longjmp( )	√
110	*setjmp.h*	setjmp( )	√
	POSIX_C_LANG_SUPPORT General Functions		
111	*stdlib.h*	abs( )	√
112	*stdlib.h*	atof( )	√
113	*stdlib.h*	atoi( )	√

114	*stdlib.h*	atol( )	√
115	*stdlib.h*	bsearch( )	√
116	*stdlib.h*	calloc( )	√
117	*stdlib.h*	free( )	√
118	*stdlib.h*	malloc( )	√
119	*stdlib.h*	qsort( )	√
120	*stdlib.h*	rand( )	√
121	*stdlib.h*	realloc( )	√
122	*stdlib.h*	srand( )	√
	POSIX_C_LANG_SUPPORT **String Handling Functions**		
123	*string.h*	strcat( )	√
124	*string.h*	strchr( )	√
125	*string.h*	strcmp( )	√
126	*string.h*	strcpy( )	√
127	*string.h*	strcspn( )	√
128	*string.h*	strlen( )	√
129	*string.h*	strncpy( )	√
130	*string.h*	strncat( )	√
131	*string.h*	strncmp( )	√
132	*string.h*	strpbkr( )	√
133	*string.h*	strrchr( )	√
134	*string.h*	strspn( )	√
135	*string.h*	strstr( )	√
136	*string.h*	strtok( )	√
	POSIX_C_LANG_SUPPORT **Data and Time Functions**	Affects: time( )	
137	*time.h*	asctime( )	√
138	*time.h*	ctime( )	√
139	*time.h*	gmtime( )	√
140	*time.h*	localtime( )	√
141	*time.h*	mktime( )	√
142	*time.h*	strftime( )	√
	POSIX.1b Option Requirements		
	{_POSIX_ASYNCHRONOUS_IO}		
143	*aio.h*	aio_read( ) [2]	√
144	*aio.h*	aio_write( ) [2]	√
145	*aio.h*	lio_listio( ) [2]	√
146	*aio.h*	aio_error( )	√
147	*aio.h*	aio_return( )	√
148	*aio.h*	aio_cancel( )	√
149	*aio.h*	aio_suspend( )	√
150	*aio.h*	aio_fsync( ) [2]	√
	{_POSIX_MEMLOCK}		
151	sys/mman.h	mlockall( )	

152	sys/mman.h	munlockall( )	√
	{_POSIX_MEMLOCK_RANGE}		
153	sys/mman.h	mlock( )	√
154	sys/mman.h	munlock( )	√
	{_POSIX_MESSAGE_PASSING}		
155	*mqueue.h*	mq_open( )	√
156	*mqueue.h*	mq_close( )	√
157	*mqueue.h*	mq_unlink( )	√
158	*mqueue.h*	mq_send( )	√
159	*mqueue.h*	mq_receive( )	√
160	*mqueue.h*	mq_notify( ) [3]	√
161	*mqueue.h*	mq_setattr( )	√
162	*mqueue.h*	mq_getattr( )	√
	{_POSIX_REALTIME_SIGNALS}	Affects: mq_notify( )	√
163	signal.h	sigwaitinfo( )	√
164	signal.h	sigtimedwait( )	√
165	signal.h	sigqueue( )	√
	{_POSIX_SEMAPHORES}		
166	*semaphore.h*	sem_init( )	√
167	*semaphore.h*	sem_destroy( )	√
168	*semaphore.h*	sem_open( )	√
169	*semaphore.h*	sem_close( )	√
170	*semaphore.h*	sem_unlink( )	√
171	*semaphore.h*	sem_wait( )	√
172	*semaphore.h*	sem_trywait( )	√
173	*semaphore.h*	sem_post( )	√
174	*semaphore.h*	sem_getvalue( )	√
	{_POSIX_SYNCHRONIZED_IO}	Affects: aio_read, aio_write, lio_listio, aio_fsync	
	{_POSIX_TIMERS}		
175	*time.h*	clock_settime( )	√
176	*time.h*	clock_gettime( )	√
177	*time.h*	clock_getres( )	√
178	*time.h, signal.h*	timer_create( )	√
179	*time.h*	timer_delete( )	√
180	*time.h*	timer_settime( )	√
181	*time.h*	timer_gettime( )	√
182	*time.h*	timer_getoverrun( )	√
183	*time.h*	nanosleep( )	√
	{_POSIX_FSYNC}	(fsync( ) not required)	

	POSIX.1c Option Requirements		
	{_POSIX_THREADS}		
184	*sched.h*	sched_yield( )	√
185	*pthread.h, sys/types.h*	pthread_atfork( )	√
186	*pthread.h, signal.h*	phthread_kill( )	√
187	*pthread.h*	pthread_sigmask( )	√
188	*pthread.h*	pthread_attr_init( )	√
189	*pthread.h*	pthread_attr_destroy( )	√
190	*pthread.h*	pthread_attr_setdetachstate( )	√
191	*pthread.h*	pthread_attr_getdetachstate( )	√
192	*pthread.h*	pthread_attr_getschedparam( )	√
193	*pthread.h*	pthread_attr_setschedparam( )	√
194	*pthread.h*	pthread_create( )	√
195	*pthread.h*	pthread_join( )	√
196	*pthread.h*	pthread_detach( )	√
197	*pthread.h*	pthread_exit( )	√
198	*pthread.h*	pthread_self( )	√
199	*pthread.h*	pthread_equal( )	√
200	*pthread.h*	pthread_once( )	√
201	*pthread.h*	pthread_key_create( )	√
202	*pthread.h*	pthread_setspecific( )	
203	*pthread.h*	pthread_getspecific( )	
204	*pthread.h*	pthread_key_delete( )	
205	*pthread.h*	pthread_cancel( )	
206	*pthread.h*	pthread_setcancelstate( )	
207	*pthread.h*	pthread_setcanceltype( )	
208	*pthread.h*	pthread_testcancel( )	
209	*pthread.h*	pthread_cleanup_push( )	
210	*pthread.h*	pthread_cleanup_pop( )	
211	*pthread.h*	pthread_mutexattr_init( )	
212	*pthread.h*	pthread_mutexattr_destroy( )	
213	*pthread.h*	pthread_mutex_init( )	
214	*pthread.h*	pthread_mutex_destroy( )	
215	*pthread.h*	pthread_mutex_lock( )	
216	*pthread.h*	pthread_mutex_unlock( )	
217	*pthread.h*	pthread_mutex_trylock( )	
218	*pthread.h*	pthread_condattr_init( )	
219	*pthread.h*	pthread_condattr_destroy( )	
220	*pthread.h*	pthread_cond_init( )	
221	*pthread.h*	pthread_cond_destroy( )	
222	*pthread.h*	pthread_cond_signal( )	
223	*pthread.h*	pthread_cond_broadcast( )	
224	*pthread.h*	pthread_cond_wait( )	
225	*pthread.h*	pthread_cond_timedwait( )	
	{_POSIX_THREAD_ATTR_STACKADDR}		
226	*pthread.h*	pthread_attr_setstackaddr( ) [4]	

227	*pthread.h*	pthread_attr_getstackaddr( ) [4]	
	{_POSIX_THREAD_ATTR_STACKSIZE}		
228	*pthread.h*	pthread_attr_setstacksize( ) [4]	
229	*pthread.h*	pthread_attr_getstacksize( ) [4]	
	{_POSIX_THREAD_PRIO_INHERIT}		
230	*pthread.h*	pthread_mutexattr_setprotocol( ) [5]	
231	*pthread.h*	pthread_mutexattr_getprotocol( ) [5]	
232	*pthread.h*	pthread_mutexattr_setprioceiling( ) [5]	
233	*pthread.h*	pthread_mutexattr_getprioceiling( ) [5]	
	{_POSIX_THREAD_PRIO_PROTECT} *pthread.h*		
234	*pthread.h*	pthread_mutex_getprioceiling( )	
235	*pthread.h*	pthread_mutex_setprioceiling( )	
	{_POSIX_THREAD_PRIORITY_SCHEDULING}		
236	*pthread.h*	pthread_attr_setscope( )	
237	*pthread.h*	pthread_attr_getscope( )	
238	*pthread.h*	pthread_attr_setinheritsched( )	
239	*pthread.h*	pthread_attr_getinheritsched( )	
240	*pthread.h*	pthread_attr_setschedpolicy( )	
241	*pthread.h*	pthread_attr_getschedpolicy( )	
242	*pthread.h*	pthread_setschedparam( )	
243	*pthread.h*	pthread_getschedparam( )	
	POSIX_C_LANG_SUPPORT_R Functions		
	{_POSIX_THREAD_SAFE_FUNCTIONS}		
244	*time.h*	asctime_r( )	√
245	*time.h*	ctime_r( )	√
246	*time.h*	gmtime_r( )	√
247	*time.h*	localtime_r( )	√
248	*stdlib.h*	rand_r( )	√
249	*string.h*	strtok_r( )	√

√ - Function present
? - Inadequate documentation
N - Nonconforming function
X - Function or option not present

[1] See also POSIX_C_LANG_SUPPORT Date and Time Functions
[2] Affected by {_POSIX_SYNCHRONIZED_IO}
[3] Affected by {_POSIX_REALTIME_SIGNALS}
[4] Part of {_POSIX_THREADS}

[5] Or {_POSIX_THREAD_PRIO_PROTECT}