Date: March 2007

POSIX Profiles Specification, v1.0

OMG Available Specification
formal/07-03-06

The PIM and PSM for Software Radio Components Specification (formal/07-03-01) is physically partitioned into 5 volumes:

Communication Channel and Equipment (formal/07-03-02)
Component Document Type Definitions (formal/07-03-03)
Component Framework (formal/07-03-04)
Common and Data Link Layer Facilities (formal/07-03-05)
POSIX Profiles (formal/07-03-06)

OBJECT MANAGEMENT GROUP

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

LICENSES

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 250 First Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™ , Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IIOP™ , MOF™ and OMG Interface Definition Language (IDL)™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

OMG’s Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page *http://www.omg.org*, under Documents, Report a Bug/Issue (*http://www.omg.org/technology/agreement.htm*).

Table of Contents

Preface

About the Object Management Group

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at *http://www.omg.org/*.

OMG Specifications

As noted, OMG specifications address middleware, modeling, and vertical domain frameworks. A Specifications Catalog is available from the OMG website at:

http://www.omg.org/technology/documents/spec_catalog.htm

Specifications within the Catalog are organized by the following categories:

OMG Modeling Specifications

- UML
- MOF
- XMI
- CWM
- Profile specifications.

OMG Middleware Specifications

- CORBA/IIOP
- IDL/Language Mappings
- Specialized CORBA specifications
- CORBA Component Model (CCM).

Platform Specific Model and Interface Specifications

- CORBAservices
- CORBAfacilities
- OMG Domain specifications
- OMG Embedded Intelligence specifications
- OMG Security specifications.

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
140 Kendrick Street
Building A, Suite 300
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
Email: pubs@omg.org

Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

Helvetica/Arial - 10 pt. Bold: OMG Interface Definition Language (OMG IDL) and syntax elements.

`Courier - 10 pt. Bold:` Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

Note – Terms that appear in *italics* are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Issues

The reader is encouraged to report any technical or editing issues/problems with this specification to *http://www.omg.org/technology/agreement.htm*.

1 Scope

This specification defines the application environment profiles for embedded constrained systems, based on Standardized Application Environment Profile - POSIX® Realtime Application Support (AEP), IEEE Std 1003.13-2003.

The specification includes two specific profiles, which are characterized as follows:

1. The application environment profile (AEP), which is for constrained embedded general purpose processing, is the preferred profile for embedded processing and its utilization is encouraged for all processing environments.

2. The lightweight application environment profile (LwAEP). LwAEP is more constrained than the AEP and is targeted towards environments with limited computing support. Examples of constrained embedded processors include Digital Signal Processors (DSPs), processor cores within Field Programmable Gate Arrays (FPGAs) and micro-controllers. Application of LwAEP should apply only to DSPs. Use of LwAEP in an FPGA-based processor core is encouraged but not required.

2 Conventions

Within this specification, the following abbreviations are used:

1. “MAN” indicates that the identified function or option is mandatory for the indicated profile.

2. “NRQ” indicates that the identified function or option is not required for the indicated profile.

3. “PRT” indicates that only a subset of the identified function or unit of functionality is required. The designation will be followed by a note or cross-reference indicating which elements are required.

4. “OPT” indicates that the identified function or option may be included within the environment.

3 Conformance

The real-time profiles defined in this standard requires only specific Units of Functionality of the required standards. The absence of particular elements of these standards introduces constraints on the use of some of the features of particular functions. These constraints must be observed by an application that conforms to the profile when using each of the required functions.

An Ada AEP has not been explicitly defined. Any Ada application shall be restricted to using the equivalent Ada functionality, as defined in POSIX Ada language binding (ISO/IEC 14519:2001), designated as mandatory by this profile or may use the C interface.

Key considerations in selection of functions for the embedded processor are as follows:

- Of late, DSP development environments include operating systems that offer a rich and scaleable feature set – preemptive multitasking, installable interrupt handlers and inter-process communications.

- Current DSP technology does not employ Memory Management Units (MMUs).

- Different DSP environments sometimes offer extensions or services that target specific market segments optimizations for video processing, power savings features and kernel support for real-time debugging.
- Current embedded state-of-the-art does not exploit loadable modules. Entire FPGA and DSP images containing infrastructure and application software are loaded simultaneously as part of waveform instantiation.

Ultimately the presence of a full-featured RTOS in the embedded processor is a relatively new practice and yet one that is recognized as offering life cycle software cost benefit. The state-of-the-art will continue to advance and this specification shall not disallow the migration of new design paradigms as they become matured and practiced.

The following standards are required in whole or in part by the AEP.

Table 2.1 - Required Standards

Standard	AEP	LwAEP
C Standard (ISO/IEC 9899:1999)	PRT	PRT
POSIX (ISO/IEC 9945:2003)	PRT	PRT

4 References

4.1 Normative References

4.1.1 POSIX Specifications

4.1.1.1 Application Environment Profile

Standardized Application Environment Profile - POSIX® Realtime Application Support (AEP),
IEEE Std 1003.13-2003
Institute of Electrical and Electronics Engineers, 2003
http://standards.ieee.org

4.1.1.2 C Standard

C Standard (ISO/IEC 9899:1999)
The Open Group, 1999
http://www.opengroup.org/

4.1.1.3 POSIX

Information technology Portable Operating System Interface
POSIX (ISO/IEC 9945:2003)
International Organization for Standardization, 2003
http://www.iso.org/

4.1.1.4 POSIX Ada Language Binding

Information Technology-POSIX Ada Language Interfaces
POSIX Ada Language Interfaces (ISO/IEC 14159 - 2001)
International Organization for Standardization, 2001
http://www.iso.org

5 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

Common Object Request Broker Architecture (CORBA)

An OMG distributed computing platform specification that is independent of implementation languages.

Component

A component can always be considered an autonomous unit within a system or subsystem. It has one or more ports, and its internals are hidden and inaccessible other than as provided by its interfaces. A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment. A component exposes a set of ports that define the component specification in terms of provided and required interfaces. As such, a component serves as a type, whose conformance is defined by these provided and required interfaces (encompassing both their static as well as dynamic semantics).

Platform

A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Request for Proposal (RFP)

A document requesting OMG members to submit proposals to the OMG's Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing task force.

6 Symbols and abbreviated terms

Abbreviation	Definition
CORBA	Common Object Request Broker Architecture
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
I/O	Input/Output
IDL	Interface Definition Language
IIOP	Internet Inter-ORB Protocol
ISO	International Standards Organization
MAN	Mandatory
NRQ	Not Required
OMG	Object Management Group
OPT	Optional
POSIX	Portable Operating System Interface
PRT	Partial
PSE52	Real-time Controller System Profile, defined in IEEE Std. 1003.13
PSM	Platform Specific Model
RFP	Request For Proposal
UML	Unified Modeling Language

7 Additional Information

7.1 Changes to Adopted OMG Specifications

The specifications contained in this document require no changes to adopted OMG specifications.

7.2 Guide to this Specification

Chapter 8 provides a set of tables that organize POSIX data types and library functions into logical categories. Each row in these tables contains a POSIX function name, the IEEE section numbers, and the designation of mandatory, not required, partially required, or optional.

7.3 Acknowledgements

The following organizations (listed in alphabetical order) contributed to this specification:

- BAE Systems
- The Boeing Company
- Blue Collar Objects
- Carleton University
- Communications Research Center Canada
- David Frankel Consulting
- École de Technologie Supérieure
- General Dynamics Decision Systems
- Harris
- ISR Technologies
- ITT Aerospace/Communications Division
- L-3 Communications Corporation
- Mercury Computer Systems
- The MITRE Corporation
- Mobile Smarts
- Northrup Grumman
- PrismTech
- Raytheon Corporation
- Rockwell Collins
- SCA Technica
- Space Coast Communication Systems
- Spectrum Signal Processing
- THALES
- Virginia Tech University
- Zeligsoft
- 88solutions

8 Operating System Profiles

8.1 Posix.1

The options, limits, and any other constraints on POSIX.1 shall be provided as described in Table 8.1.

Table 8.1 - POSIX.1 Option Requirements

Option	AEP	LwAEP
{_POSIX_ASYNCHRONOUS_IO}	MAN	NRQ
{_POSIX_CHOWN_RESTRICTED}	NRQ	NRQ
{_POSIX_CLOCK_SELECTION}	NRQ	NRQ
{_POSIX_FSYNC}	PRT[a]	NRQ
{_POSIX_MAPPED_FILES}	NRQ	NRQ
{_POSIX_MEMLOCK_RANGE}	MAN	NRQ
{_POSIX_MEMLOCK}	MAN	NRQ
{_POSIX_MEMORY_PROTECTION}	NRQ	NRQ
{_POSIX_MESSAGE_PASSING}	MAN	NRQ
{_POSIX_MONOTONIC_CLOCK}	NRQ	NRQ
{_POSIX_NO_TRUNC}	PRI	NRQ
{_POSIX_PRIORITIZED_IO}	NRQ	NRQ
{_POSIX_PRIORITY_SCHEDULING}	NRQ	NRQ
{_POSIX_REALTIME_SIGNALS}	MAN	NRQ
{_POSIX_SAVED_IDS}	NRQ	NRQ
{_POSIX_SEMAPHORES}	MAN	PRT
{_POSIX_SHARED_MEMORY_OBJECTS}	NRQ	NRQ
{_POSIX_SYNCHRONIZED_IO}	PRT[b]	NRQ
{_POSIX_THREAD_ATTR_STACKADDR}	MAN	NRQ
{_POSIX_THREAD_ATTR_STACKSIZE}	MAN	MAN
{_POSIX_THREAD_CPUTIME}	NRQ	NRQ
{_POSIX_THREAD_PRIO_INHERIT}	MAN	NRQ
{_POSIX_THREAD_PRIO_PROTECT}	MAN	NRQ
{_POSIX_THREAD_PRIORITY_SCHEDULING}	MAN	MAN
{_POSIX_THREAD_PROCESS_SHARED}	NRQ	NRQ
{_POSIX_THREAD_SAFE_FUNCTIONS}	PRT[c]	PRT[d]
{_POSIX_THREAD_SPORADIC_SERVER}	NRQ	NRQ

Table 8.1 - POSIX.1 Option Requirements

Option	AEP	LwAEP
{_POSIX_TIMEOUTS}	NRQ	NRQ
{_POSIX_TIMERS}	MAN	PRT
{_POSIX_TRACE_EVENT_FILTER}	NRQ	NRQ
{_POSIX_TRACE_LOG}	NRQ	NRQ
{_POSIX_TRACE}	NRQ	NRQ
{_POSIX_VDISABLE}	NRQ	NRQ

a. fsync not required

b. fdatasync not required

c. See Table 8.22 : POSIX_THREAD_SAFE_FUNCTIONS Functions

d. See Table 8.22 : POSIX_THREAD_SAFE_FUNCTIONS Functions

Note:

- PRI - The primary file system shall generate an error for pathname components longer than NAME_MAX. The user is responsible for semantics of other file systems that may be mounted.
- Embedded processor C/C++ run-time libraries typically do not support stdio.h or iostream.h
- Heavy weight processes are typically not supported in embedded operating systems. The mandatory POSIX.1 options can be implemented without the use of heavy weight signaling.

8.1.1 Single Process Function Behavior

The functions in Table 8.2 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.2 - POSIX_SINGLE_PROCESS Functions

Function	AEP	LwAEP
confstr()	NRQ	NRQ
environ	NRQ	NRQ
errno	NRQ	NRQ
getenv ( )	NRQ	NRQ
setenv()	NRQ	NRQ
sysconf ( )	NRQ	NRQ
uname( )	NRQ	NRQ
unsetenv()	NRQ	NRQ

8.1.2 Multi-Process Function Behavior

The functions listed in Table 8.3 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.3 - POSIX_MULTI_PROCESS Functions

Function	AEP	LwAEP
_exit ( )	NRQ	NRQ
_Exit()	NRQ	NRQ
assert ( )	NRQ	NRQ
atexit()	NRQ	NRQ
clock()	NRQ	NRQ
execl ( )	NRQ	NRQ
execle ( )	NRQ	NRQ
execlp ( )	NRQ	NRQ
execv ( )	NRQ	NRQ
execve ( )	NRQ	NRQ
execvp ( )	NRQ	NRQ
exit ( )	NRQ	NRQ
fork( )	NRQ	NRQ
getpgrp()	NRQ	NRQ
getpid ( )	NRQ	NRQ
getppid ( )	NRQ	NRQ
setsid()	NRQ	NRQ
sleep ( )	NRQ	NRQ
times ( )	NRQ	NRQ
wait( )	NRQ	NRQ
waitpid ( )	NRQ	NRQ

8.1.3 Job Control Function Behavior

The functions listed in Table 8.4 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.4 - POSIX_JOB_CONTROL Functions

Function*	AEP	LwAEP
setpgid()	NRQ	NRQ
tcgetpgrp()	NRQ	NRQ
tcsetpgrp()	NRQ	NRQ
*	NRQ	NRQ

Note:

- Further functionality is also defined here.

8.1.4 Signals Function Behavior

Operating systems on embedded processors typically support neither signaling nor exception handling. POSIX does not define behaviors associated with divide by zero or overflow / underflow. Signaling methods introduced as part of POSIX.1c are more consistent with the multi-threaded, single process model of a resource constrained processing environment.

The functions listed in Table 8.5 shall behave as described in the applicable clauses of the referenced POSIX specification, except for the following constraints:

1. An application that conforms to the AEP shall not result in abnormal termination of the process because this profile does not support multiple processes.
2. An application that conforms to the AEP shall not call the kill() function with a negative argument because this profile does not require process group functionality.

Table 8.5 - POSIX_SIGNALS Functions

Function	AEP	LwAEP
abort()	MAN	MAN
alarm()*	NRQ	NRQ
kill()	MAN	NRQ
pause()	MAN	NRQ
raise()	MAN	NRQ
sigaction()	MAN	NRQ
sigaddset()	MAN	NRQ
sigdelset()	MAN	NRQ

Function	AEP	LwAEP
sigemptyset()	MAN	NRQ
sigfillset()	MAN	NRQ
sigismember()	MAN	NRQ
signal()	MAN	NRQ
sigpending()	MAN	NRQ
sigprocmask()	MAN	NRQ
sigsupend()	MAN	NRQ
sigwait()	MAN	NRQ

Note:

- *Functionality provided through the POSIX timers.
- abort() is used to support assert() which is widely supported.

8.1.5 Signal Jump Function Behavior

The functions listed in Table 8.6 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.6 - POSIX_SIGNAL_JUMP Functions

Function	AEP	LwAEP
siglongjmp()	NRQ	NRQ
sigsetjmp()	NRQ	NRQ

8.1.6 User Group Function Behavior

The functions listed in Table 8.7 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.7 - POSIX_USER_GROUPS Functions

Function	AEP	LwAEP
getegid()	NRQ	NRQ
geteuid()	NRQ	NRQ
getgid()	NRQ	NRQ
getgroups()	NRQ	NRQ
getlogin()	NRQ	NRQ
getlogin_r()	NRQ	NRQ
getuid()	NRQ	NRQ
setegid()	NRQ	NRQ
seteuid()	NRQ	NRQ
setgid()	NRQ	NRQ
setuid()	NRQ	NRQ

8.1.7 File System Function Behavior

The functions listed in Table 8.8 shall behave as described in the applicable clauses of the referenced POSIX specification..

Table 8.8 - POSIX_FILE_SYSTEM Functions

Function	AEP	LwAEP
access()	MAN	NRQ
chdir()	MAN	NRQ
closedir()	MAN	NRQ
creat()	MAN	NRQ
fpathconf()	MAN	NRQ
fstat()	MAN	NRQ
getcwd()	MAN	NRQ
link()	MAN	NRQ
mkdir()	MAN	NRQ
opendir()	MAN	NRQ
pathconf()	MAN	NRQ
readdir()	MAN	NRQ
readdir_r()	MAN	NRQ
remove()	MAN	NRQ
rename()	MAN	NRQ
rewinddir()	MAN	NRQ
rmdir()	MAN	NRQ
stat()	MAN	NRQ
tmpfile()	MAN	NRQ
tmpnam()	MAN	NRQ
unlink()	MAN	NRQ
utime()	MAN	NRQ

Note:

- POSIX file system not generally supported in embedded operating systems.

8.1.8 File Attributes Function Behavior

The functions listed in Table 8.9 shall behave as described in the applicable clauses of the referenced POSIX specification, except for the following constraint:

1. An application that conforms to the AEP shall be guaranteed that the file mode creation mask for any object created by any process is S_IRWXU; that is, the object shall be fully accessible to the creator.

Table 8.9 - POSIX_FILE_ATTRIBUTES Functions

Function	AEP	LwAEP
chmod()	NRQ	NRQ
chown()	NRQ	NRQ
fchmod()	NRQ	NRQ
fchown()	NRQ	NRQ
umask()	NRQ	NRQ

Note:

- POSIX file system not generally supported in embedded operating systems.

8.1.9 File and Directory Management Function Behavior

The functions listed in Table 8.10 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.10 - POSIX_FD_MGMT Functions

Function	AEP	LwAEP
dup()	NRQ	NRQ
dup2()	NRQ	NRQ
fcntl()	NRQ	NRQ
fgetpos()	NRQ	NRQ
fseek()	MAN	NRQ
fseeko()	MAN	NRQ
fsetpos()	NRQ	NRQ
ftell()	MAN	NRQ
ftello()	MAN	NRQ
ftruncate()	NRQ	NRQ
lseek()	MAN	NRQ
rewind()	MAN	NRQ

Note:

- POSIX file system not generally supported in embedded operating systems.

8.1.10 Device I/O Function Behavior

The functions listed in Table 8.11 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.11 - POSIX_DEVICE_IO Functions

Function	AEP	LwAEP
clearerr()	MAN	NRQ
close()	MAN	NRQ
fclose()	MAN	NRQ
fdopen()	MAN	NRQ
feof()	MAN	NRQ
ferror()	MAN	NRQ
fflush()	MAN	NRQ
fgetc()	MAN	NRQ
fgets()	MAN	NRQ
fileno()	MAN	NRQ
fopen()	MAN	NRQ
fprintf()	MAN	NRQ
fputc()	MAN	NRQ
fputs()	MAN	NRQ
fread()	MAN	NRQ
freopen()	MAN	NRQ
fscanf()	MAN	NRQ
fwrite()	MAN	NRQ
getc()	MAN	NRQ
getchar()	MAN	NRQ
gets()	MAN	NRQ
open()	MAN	NRQ
perror()	MAN	NRQ
printf()	MAN	NRQ
putc()	MAN	NRQ
putchar()	MAN	NRQ
puts()	MAN	NRQ
read()	MAN	NRQ
scanf()	MAN	NRQ
setbuf()	MAN	NRQ
setvbuf()	MAN	NRQ
stderr()	NRQ	NRQ
stdin()	NRQ	NRQ
stdout()	NRQ	NRQ
ungetc()	MAN	NRQ
vfprintf()	NRQ	NRQ
vfscanf()	NRQ	NRQ
vprintf()	NRQ	NRQ
vscanf()	NRQ	NRQ
write()	MAN	NRQ

Note:

- POSIX streams not generally supported in embedded operating systems.

8.1.11 Device-Specific Function Behavior

The functions listed in Table 8.12 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.12 - POSIX_DEVICE_SPECIFIC Functions

Function	AEP	LwAEP
cfgetispeed()	NRQ	NRQ
cfgetospeed()	NRQ	NRQ
cfsetispeed()	NRQ	NRQ
cfsetospeed()	NRQ	NRQ
ctermid()	NRQ	NRQ
isatty()	NRQ	NRQ
tcdrain()	NRQ	NRQ
tcflow()	NRQ	NRQ
tcflush()	NRQ	NRQ
tcgetattr()	NRQ	NRQ
tcsendbreak()	NRQ	NRQ
tcsetattr()	NRQ	NRQ
ttyname()	NRQ	NRQ
ttyname_r()	NRQ	NRQ

8.1.12 System Database Function Behavior

The functions listed in Table 8.13 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.13 - POSIX_SYSTEM_DATABASE Functions

Function	AEP	LwAEP
getgrgid()	NRQ	NRQ
getgrgid_r()	NRQ	NRQ
getgrnam()	NRQ	NRQ
getgrnam_r()	NRQ	NRQ
getpwnam()	NRQ	NRQ
getpwnam_r()	NRQ	NRQ
getpwuid()	NRQ	NRQ
getpwuid_r()	NRQ	NRQ

8.1.13 Pipe Function Behavior

The function listed in Table 8.14 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.14 - POSIX_PIPE_Function

Function	AEP	LwAEP
pipe()	NRQ	NRQ

8.1.14 FIFO Function Behavior

The functions listed in Table 8.15 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.15 - POSIX_FIFO Function

Function	AEP	LwAEP
mkfifo()	NRQ	NRQ

8.1.15 Language-Specific Support Services Function Behavior

The functions listed in Table 8.16 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.16 - POSIX_C_LANG_SUPPORT Functions

Function	AEP	LwAEP
abs()	MAN	MAN
asctime()	MAN	MAN
asctime_r()	MAN	MAN
atof()	MAN	NRQ
atoi()	MAN	NRQ
atol()	MAN	MAN
atoll()	MAN	MAN
bsearch()	MAN	NRQ
calloc()	MAN	MAN
ctime()	MAN	NRQ
ctime_r()	MAN	NRQ
difftime()	NRQ	NRQ
div()	NRQ	NRQ
feclearexcept()	NRQ	NRQ
fegetenv()	NRQ	NRQ
fegetexceptflag()	NRQ	NRQ
fegetround()	NRQ	NRQ
feholdexcept()	NRQ	NRQ
feraiseexcept()	NRQ	NRQ
fesetexceptflag()	NRQ	NRQ
fesetround()	NRQ	NRQ
fetestexcept()	NRQ	NRQ
feupdateenv()	NRQ	NRQ
free()	MAN	MAN
fesetenv()	NRQ	NRQ
gmtime()	MAN	NRQ
gmtime_r()	MAN	NRQ
imaxabs()	NRQ	NRQ

Table 8.16 - POSIX_C_LANG_SUPPORT Functions

Function	AEP	LwAEP
imaxdiv()	NRQ	NRQ
isalnum()	MAN	NRQ
isalpha()	MAN	MAN
isblank()	NRQ	NRQ
iscntrl()	MAN	NRQ
isdigit()	MAN	MAN
isgraph()	MAN	NRQ
islower()	MAN	NRQ
isprint()	MAN	MAN
ispunct()	MAN	NRQ
isspace()	MAN	NRQ
isupper()	MAN	NRQ
isxdigit()	MAN	MAN
labs()	NRQ	NRQ
ldiv()	NRQ	NRQ
llabs()	NRQ	NRQ
lldiv()	NRQ	NRQ
localeconv()	NRQ	NRQ
localtime()	MAN	NRQ
localtime_r()	MAN	NRQ
malloc()	MAN	MAN
memchr()	NRQ	NRQ
memcmp()	NRQ	NRQ
memcpy()	NRQ	NRQ
memmove()	NRQ	NRQ
memset()	NRQ	NRQ
mktime()	MAN	NRQ
qsort()	MAN	NRQ
rand()	MAN	MAN
rand_r()	MAN	MAN
realloc()	MAN	MAN
setlocale ( )	MAN	NRQ
snprintf()	NRQ	NRQ
sprintf()	MAN	NRQ
srand()	MAN	MAN
sscanf()	MAN	NRQ
strcat()	MAN	NRQ
strchr()	MAN	NRQ
strcmp()	MAN	NRQ
strcoll()	NRQ	NRQ
strcpy()	MAN	NRQ
strcspn()	MAN	NRQ
strerror()	NRQ	NRQ
strerror_r()	NRQ	NRQ
strftime()	MAN	NRQ
strlen()	MAN	NRQ
strncat()	MAN	NRQ
strncmp()	MAN	MAN
strncpy()	MAN	NRQ
strpbrk()	MAN	NRQ
strrchr()	MAN	NRQ
strspn()	MAN	NRQ
strstr()	MAN	NRQ
strtod()	NRQ	NRQ
strtof()	NRQ	NRQ
strtoimax()	NRQ	NRQ
strtok()	MAN	NRQ
strtok_r()	MAN	NRQ
strtol()	NRQ	NRQ
strtold()	NRQ	NRQ
strtoll()	NRQ	NRQ
strtoul()	NRQ	NRQ
strtoull()	NRQ	NRQ
strtoumax()	NRQ	NRQ
strxfrm()	NRQ	NRQ
time()	MAN	NRQ

Table 8.16 - POSIX_C_LANG_SUPPORT Functions

Function	AEP	LwAEP
tolower()	MAN	MAN
toupper()	MAN	NRQ
tzname,	NRQ	NRQ
tzset()	NRQ	NRQ
va_arg()	NRQ	NRQ
va_copy()	NRQ	NRQ
va_end()	NRQ	NRQ
va_start()	NRQ	NRQ
vsnprintf()	NRQ	NRQ
vsprintf()	NRQ	NRQ
vsscanf()	NRQ	NRQ

- A form of context switch used to support a nonlocal exit.

8.1.16 C Language Specific Mathematical Function Behavior

The function listed in Table 8.17 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.17 - POSIX_C_LANG_MATH Functions

Function	AEP	LwAEP
acos()	MAN	NRQ
acosf()	NRQ	NRQ
acosh()	NRQ	NRQ
acoshf()	NRQ	NRQ
acoshl()	NRQ	NRQ
acosl()	NRQ	NRQ
asin()	MAN	NRQ
asinf()	NRQ	NRQ
asinh()	NRQ	NRQ
asinhf()	NRQ	NRQ
asinhl()	NRQ	NRQ
asinl()	NRQ	NRQ
cacos()	NRQ	NRQ
cacosf()	NRQ	NRQ
cacosh()	NRQ	NRQ
cacoshf()	NRQ	NRQ
cacoshl()	NRQ	NRQ
cacosl()	NRQ	NRQ
atan()	MAN	NRQ
atan2()	MAN	NRQ
atan2f()	NRQ	NRQ
atan2l()	NRQ	NRQ
atanf()	NRQ	NRQ
atanh()	NRQ	NRQ
atanhf()	NRQ	NRQ
atanhl()	NRQ	NRQ
atanl()	NRQ	NRQ
cabs()	NRQ	NRQ
cabsf()	NRQ	NRQ
cabsl()	NRQ	NRQ
cexp()	NRQ	NRQ
cexpf()	NRQ	NRQ
cexpl()	NRQ	NRQ
Cimag()	NRQ	NRQ
cimagf()	NRQ	NRQ
cimagl()	NRQ	NRQ

Table 8.17 - POSIX_C_LANG_MATH Functions

Function	AEP	LwAEP
carg()	NRQ	NRQ
cargf()	NRQ	NRQ
cargl()	NRQ	NRQ
casin()	NRQ	NRQ
casinf()	NRQ	NRQ
casinh()	NRQ	NRQ
casinhf()	NRQ	NRQ
casinhl()	NRQ	NRQ
casinl()	NRQ	NRQ
catan()	NRQ	NRQ
catanf()	NRQ	NRQ
catanh()	NRQ	NRQ
catanhf()	NRQ	NRQ
catanhl()	NRQ	NRQ
catanl()	NRQ	NRQ
cbrt()	NRQ	NRQ
cbrtf()	NRQ	NRQ
cbrtl()	NRQ	NRQ
ccos()	NRQ	NRQ
ccosf()	NRQ	NRQ
Ccosh()	NRQ	NRQ
ccoshf()	NRQ	NRQ
ccoshl()	NRQ	NRQ
ccosl()	NRQ	NRQ
ceil()	MAN	NRQ
ceilf()	NRQ	NRQ
ceill()	NRQ	NRQ
csinhf()	NRQ	NRQ
csinhl()	NRQ	NRQ
csinl()	NRQ	NRQ
csqrt()	NRQ	NRQ
csqrtf()	NRQ	NRQ
csqrtl()	NRQ	NRQ
clog()	NRQ	NRQ
clogf()	NRQ	NRQ
clogl()	NRQ	NRQ
conj()	NRQ	NRQ
conjf()	NRQ	NRQ
conjl()	NRQ	NRQ
copysign()	NRQ	NRQ
copysignf()	NRQ	NRQ
copysignl()	NRQ	NRQ
cos()	MAN	NRQ
cosf()	NRQ	NRQ
cosh()	MAN	NRQ
coshf()	NRQ	NRQ
coshl()	NRQ	NRQ
cosl()	NRQ	NRQ
cpow()	NRQ	NRQ
cpowf()	NRQ	NRQ
cpowl()	NRQ	NRQ
cproj()	NRQ	NRQ
cprojf()	NRQ	NRQ
cprojl()	NRQ	NRQ
creal()	NRQ	NRQ
crealf()	NRQ	NRQ
creall()	NRQ	NRQ
csin()	NRQ	NRQ
csinf()	NRQ	NRQ
csinh()	NRQ	NRQ
floor()	MAN	NRQ
floorf()	NRQ	NRQ
floorl()	NRQ	NRQ
fma()	NRQ	NRQ
fmaf()	NRQ	NRQ
fmal()	NRQ	NRQ

Table 8.17 - POSIX_C_LANG_MATH Functions

Function	AEP	LwAEP
ctan()	NRQ	NRQ
ctanf()	NRQ	NRQ
ctanh()	NRQ	NRQ
ctanhf()	NRQ	NRQ
ctanhl()	NRQ	NRQ
ctanl()	NRQ	NRQ
erf()	NRQ	NRQ
erfc()	NRQ	NRQ
erfcf()	NRQ	NRQ
erfcl()	NRQ	NRQ
erff()	NRQ	NRQ
erfl()	NRQ	NRQ
exp()	MAN	NRQ
exp2()	NRQ	NRQ
exp2f()	NRQ	NRQ
exp2l()	NRQ	NRQ
expf()	NRQ	NRQ
expl()	NRQ	NRQ
expm1()	NRQ	NRQ
expm1f()	NRQ	NRQ
expm1l()	NRQ	NRQ
fabs()	MAN	NRQ
fabsf()	NRQ	NRQ
fabsl()	NRQ	NRQ
fdim()	NRQ	NRQ
fdimf()	NRQ	NRQ
fdiml()	NRQ	NRQ
isnormal()	NRQ	NRQ
isunordered()	NRQ	NRQ
ldexp()	MAN	NRQ
ldexpf()	NRQ	NRQ
ldexpl()	NRQ	NRQ
lgamma()	NRQ	NRQ
fmax()	NRQ	NRQ
fmaxf()	NRQ	NRQ
fmaxl()	NRQ	NRQ
fmin()	NRQ	NRQ
fminf()	NRQ	NRQ
fminl()	NRQ	NRQ
fmod()	MAN	NRQ
fmodf()	NRQ	NRQ
fmodl()	NRQ	NRQ
fpclassify()	NRQ	NRQ
frexp()	MAN	NRQ
frexpf()	NRQ	NRQ
frexpl()	NRQ	NRQ
hypot()	NRQ	NRQ
hypot()	NRQ	NRQ
hypotl()	NRQ	NRQ
ilogb()	NRQ	NRQ
ilogbf()	NRQ	NRQ
ilogbl()	NRQ	NRQ
isfinite()	NRQ	NRQ
isgreater()	NRQ	NRQ
isgreaterequal()	NRQ	NRQ
isinf()	NRQ	NRQ
isless()	NRQ	NRQ
islessequal()	NRQ	NRQ
islessgreater()	NRQ	NRQ
isnan()	NRQ	NRQ
lroundf()	NRQ	NRQ
lroundl()	NRQ	NRQ
modf()	MAN	NRQ
modff()	NRQ	NRQ
modfl()	NRQ	NRQ
nan()	NRQ	NRQ

Table 8.17 - POSIX_C_LANG_MATH Functions

Function	AEP	LwAEP
lgammaf()	NRQ	NRQ
lgammal()	NRQ	NRQ
llrint()	NRQ	NRQ
llrintf()	NRQ	NRQ
llrintl()	NRQ	NRQ
llround()	NRQ	NRQ
llroundf()	NRQ	NRQ
llroundl()	NRQ	NRQ
log()	MAN	NRQ
log10()	MAN	NRQ
log10f()	NRQ	NRQ
log10l()	NRQ	NRQ
log1p()	NRQ	NRQ
log1pf()	NRQ	NRQ
log1pl()	NRQ	NRQ
log2()	NRQ	NRQ
log2f()	NRQ	NRQ
log2l()	NRQ	NRQ
logb()	NRQ	NRQ
logbf()	NRQ	NRQ
logbl()	NRQ	NRQ
logf()	NRQ	NRQ
logl()	NRQ	NRQ
lrint()	NRQ	NRQ
lrintf()	NRQ	NRQ
lrintl()	NRQ	NRQ
lround()	NRQ	NRQ
scalblnf()	NRQ	NRQ
scalblnl()	NRQ	NRQ
scalbn()	NRQ	NRQ
scalbnf()	NRQ	NRQ
scalbnl()	NRQ	NRQ
signbit()	NRQ	NRQ
nanf()	NRQ	NRQ
nanl()	NRQ	NRQ
nearbyint()	NRQ	NRQ
nearbyintf()	NRQ	NRQ
nearbyintl()	NRQ	NRQ
nextafter()	NRQ	NRQ
nextafterf()	NRQ	NRQ
nextafterl()	NRQ	NRQ
nexttoward()	NRQ	NRQ
nexttowardf()	NRQ	NRQ
nexttowardl()	NRQ	NRQ
pow()	MAN	NRQ
powf()	NRQ	NRQ
powl()	NRQ	NRQ
remainder()	NRQ	NRQ
remainderf()	NRQ	NRQ
remainderl()	NRQ	NRQ
remquo()	NRQ	NRQ
remquof()	NRQ	NRQ
remquol()	NRQ	NRQ
rint()	NRQ	NRQ
rintf()	NRQ	NRQ
rintl()	NRQ	NRQ
round()	NRQ	NRQ
roundf()	NRQ	NRQ
roundl()	NRQ	NRQ
scalbln()	NRQ	NRQ
sqrt()	MAN	NRQ
sqrtf()	NRQ	NRQ
sqrtl()	NRQ	NRQ
tan()	MAN	NRQ
tanf()	NRQ	NRQ
tanh()	MAN	NRQ

Table 8.17 - POSIX_C_LANG_MATH Functions

Function	AEP	LwAEP
sin()	MAN	NRQ
sinf()	NRQ	NRQ
sinh()	MAN	NRQ
sinhf()	NRQ	NRQ
sinhl()	NRQ	NRQ
sinl()	NRQ	NRQ
tanhf()	NRQ	NRQ
tanhl()	NRQ	NRQ
tanl()	NRQ	NRQ
tgamma()	NRQ	NRQ
tgammaf()	NRQ	NRQ
tgammal()	NRQ	NRQ
trunc()	NRQ	NRQ

8.1.17 C Language-Specific Nonlocal Jump Function Behavior

The functions listed in Table 8.18 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.18 - POSIX_C_LANG_JUMP Functions

Function*	AEP	LwAEP
longjmp()	MAN	NRQ
setjmp()	MAN	NRQ

Note:

- This is a form of context switch used to support a nonlocal exit.

8.1.18 POSIX Semaphores

The functions listed in Table 8.19 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.19 - POSIX.1 Semaphore Requirements

Function	AEP	LwAEP
sem_close()	MAN	NRQ
sem_destroy()	MAN	MAN
sem_getvalue()	MAN	MAN
sem_init()	MAN	MAN
sem_open()	MAN	NRQ
sem_post()	MAN	MAN

Table 8.19 - POSIX.1 Semaphore Requirements

Function	AEP	LwAEP
sem_trywait()	MAN	MAN
sem_unlink()	MAN	NRQ
sem_wait()	MAN	MAN

8.1.19 POSIX Timers

The functions listed in Table 8.20 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.20 - POSIX.1 Timer Functions

Function	AEP	LwAEP
clock_getres()	MAN	MAN
clock_gettime()	MAN	MAN
clock_settime()	MAN	MAN
nanosleep()	MAN	MAN
timer_create()	MAN	MAN
timer_delete()	MAN	MAN
timer_getoverrun()	MAN	NRQ
timer_gettime()	MAN	MAN
timer_settime()	MAN	MAN

8.1.20 POSIX Threads

The functions listed in Table 8.21 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.21 - POSIX.1 Thread Functions

Function	AEP	LwAEP
pthread_atfork()	NRQ	NRQ
pthread_attr_xxx()	MAN	PRT
pthread_cancel()	MAN	MAN
pthread_cleanup_xxx()	MAN	NRQ
pthread_cond_xxx()	MAN	NRQ
pthread_condattr_xxx()	MAN	NRQ
pthread_create()	MAN	MAN

Table 8.21 - POSIX.1 Thread Functions

Function	AEP	LwAEP
pthread_detach()	MAN	NRQ
pthread_equal()	MAN	MAN
pthread_exit()	MAN	MAN
pthread_getschedparam()	MAN	MAN
pthread_getspecific()	MAN	NRQ
pthread_join()	MAN	MAN
pthread_key_xxx()	MAN	NRQ
pthread_kill()	MAN	NRQ
pthread_mutex_xxx()	MAN	NRQ
pthread_mutexattr_xxx()	MAN	NRQ
pthread_once()	MAN	NRQ
pthread_self()	MAN	MAN
pthread_setcancelstate()	MAN	NRQ
pthread_setcanceltype()	MAN	NRQ
pthread_setschedparam()	MAN	MAN
pthread_setspecific()	MAN	NRQ
pthread_sigmask()	MAN	NRQ
pthread_testcancel()	MAN	NRQ

Note:

- PRT - Partial, only the following subset functionality is required: pthread_attr_getschedparam();pthread_attr_getstacksize();pthread_attr_init();pthread_attr_setschedparam(); pthread_attr_setstacksize(). And to implement these mandatory stack and schedule functions, it is necessary to adequately define the unsigned integer type size_t and the struct sched_param.

8.1.21 POSIX Thread Safe Option Requirements Behavior

The functions listed in Table 8.22 shall behave as described in the applicable clauses of the referenced POSIX specification.

Table 8.22 - POSIX_THREAD_SAFE_FUNCTIONS Functions

Function	AEP	LwAEP
asctime_r()	MAN	NRQ
ctime_r()	MAN	NRQ
flockfile()	NRQ	NRQ

Table 8.22 - POSIX_THREAD_SAFE_FUNCTIONS Functions

Function	AEP	LwAEP
ftrylockfile()	NRQ	NRQ
funlockfile()	NRQ	NRQ
getc_unlocked()	NRQ	NRQ
getchar_unlocked()	NRQ	NRQ
getgrgid_r()	NRQ	NRQ
getgrnam_r()	NRQ	NRQ
getlogin_r()	NRQ	NRQ
getpwnam_r()	NRQ	NRQ
getpwuid_r()	NRQ	NRQ
gmtime_r()	MAN	NRQ
localtime_r()	MAN	NRQ
putc_unlocked()	NRQ	NRQ
putchar_unlocked()	NRQ	NRQ
rand_r()	MAN	MAN
readdir_r()	MAN	NRQ
strerror_r()	NRQ	NRQ
strtok_r()	MAN	NRQ
ttyname_r()	NRQ	NRQ

Annex A   Software Radio Reference Sheet

The Software Radio specification responds to the requirements set by “Request for Proposals for a Platform Independent Model (PIM) and CORBA Platform Specific Model (PSM)” (swradio/02-06-02). The original specification (dtc/05-10-02) has been reorganized into 5 volumes, as follows:

Volume 1. Communication Channel and Equipment

This specification describes a UML profile for communication channel. The profile provides definitions for creating communication channel and communication equipment definitions. The specification also provides radio control facilities and physical layer facilities PIM for defining interfaces and components for managing communication channels and equipment for a radio set or radio system. Along with the profile and facilities is a platform specific model transformation rule set for transforming the communication channel into an XML representation and CORBA interfaces for the radio control facilities.

Volume 2. Component Document Type Definitions

This specification defines the content of a standard set of Data Type Definition (DTD) files for applications, components, and domain and device management. The complete DTD set is contained in Section 7, Document Type Definitions. XML files that are compliant with these DTD files will contain information about the service components to be started up when a platform is power on and information for deploying installed applications.

Volume 3. Component Framework

This specification describes a UML profile for component framework. The profile provides definitions for applications, components (properties, ports, interfaces, etc.), services, artifacts, logical devices, and infrastructure domain management components. In the profile are also library packages that contain interfaces for application, service, logical device, and infrastructure domain management components. Along with the profile is a platform specific model transformation rule set for transforming the profile model library interfaces into CORBA interfaces.

Volume 4. Common and Data Link Layer Facilities

This specification describes a set of facilities PIM for application and component definitions. The set of facilities are common layer facilities and data link layer facilities that can be utilized in developing waveforms and platform components, which promote the portability of waveforms across Software Defined Radios (SDR). Along with the facilities PIM is a platform specific model transformation rule set for transforming the facilities into CORBA interfaces.

Volume 5. POSIX

This specification defines the application environment profiles for embedded constraint systems, based on Standardized Application Environment Profile - POSIX® Realtime Application Support (AEP), IEEE Std 1003.13-2003 and PSE%2:2003.

INDEX