

SOFTWARE COMMUNICATIONS ARCHITECTURE SPECIFICATION

APPENDIX E-3: PLATFORM SPECIFIC MODEL (PSM) - OBJECT MANAGEMENT GROUP INTERFACE DEFINITION LANGUAGE



28 February 2012
Version: 4.0

Prepared by:

JTRS Standards
Joint Program Executive Office (JPEO) for the Joint Tactical Radio System (JTRS)
33000 Nixie Way
San Diego, CA 92147-5110

Statement A - Approved for public release; distribution is unlimited (01 April 2012)

REVISION SUMMARY

Version	Revision	Date
Next <Draft>	Initial Draft Release	30 November 2010
Candidate Release	Initial Release	27 December 2011
4.0	ICWG Approved Release	28 February 2012

TABLE OF CONTENTS

APPENDIX E-3	PSM – OMG IDL	4
E-3.1	SCOPE	4
E-3.1	CONFORMANCE	4
E-3.2	CONVENTIONS	4
E-3.3	NORMATIVE REFERENCES	4
E-3.4	INFORMATIVE REFERENCES	4
E-3.5	UML TO CORBA IDL PSM MAPPING	4
E-3.6	ATTACHMENTS	6

LIST OF TABLES

TABLE 1: SCA UML TO OMG IDL PSM MAPPING

5

APPENDIX E-3 PSM – OMG IDL

E-3.1 SCOPE

This appendix defines the platform specific transport and technology model using the Object Management Group (OMG) Interface Definition Language (IDL) [1].

E-3.1 CONFORMANCE

Conformance with the IDL PSM representation of the SCA can be achieved by using the IDL documented in SCA Appendix C. See SCA Appendix E for additional information.

E-3.2 CONVENTIONS

N/A

E-3.3 NORMATIVE REFERENCES

The following documents contain provisions or requirements which by reference constitute requirements of this appendix (required mappings are identified in section E-3.5). Applicable versions are as stated.

- [1] Common Object Request Broker Architecture (CORBA) Specification Part 1: CORBA Interfaces, Version 3.2/ formal/2011-11-01, November 2011.

E-3.4 INFORMATIVE REFERENCES

- [2] UML™ Profile for CORBA™ Specification, Version 1.0 formal/2002-04-01, April 2002.
- [3] Common Object Request Broker Architecture (CORBA) for embedded Specification, Version 1.0 formal/2008-11-06, November 2008.

E-3.5 UML TO CORBA IDL PSM MAPPING

This section provides a mapping of the UML representation documented in the SCA PIM to its equivalent IDL representation.

The SCA technology specific mapping to OMG IDL consists of IDL that is based upon the UML Profile for CORBA [2]. The UML to IDL transformation rules are not universal rules for creating *any* technology specific mapping, but only for those within this appendix. This section defines a non-normative reference mapping of the UML constructs used within the SCA, a summary of which is represented in Table 1. The rule set for transforming UML packages, interfaces, types, and exceptions into IDL constructs are as follows:

1. UML interfaces and interface extensions map to a <<CORBAInterface>>. The <<CORBAInterface>> names, and subsequently the interface name in the IDL, are equivalent to the interface names within the SCA.

28 February 2012

2. UML attributes with readonly and readwrite map to <<CORBAInterface>> attributes and interface attributes within the IDL.
3. UML classes without operations that are not stereotyped and used as type definitions map to a <<CORBAStruct>> in a <<CORBAInterface>> or <<CORBAModule>> and a struct within the generated IDL. The designated classes name does not get translated into a CORBA type, the class attributes are added as fields within the IDL struct definition.
4. UML <<datatype>> maps to CORBA basic types. Primitive types are mapped to the corresponding IDL primitive types and primitive sequence types are mapped to a <<CORBATypedef>> and IDL typedef of primitive sequence types.
5. UML exceptions and exception extensions map to a <<CORBAException>> and an IDL exception. There is no specialization of exceptions in IDL so the SystemException definition does not appear in the generated IDL interfaces but all the specialized exceptions of SystemException are in the CORBA object module.
6. UML attributes that have a cardinality of many [*] map to a <<CORBATypedef>> of an IDL sequence type.
7. UML operations map to operations in the <<CORBAInterface>> definition and within the IDL interfaces.
8. Transformations are only performed for concrete classes, not for template classes.
9. For interfaces that reference a component stereotype for a type, the “component” qualifier is removed from the name within the IDL definition. For example, the FileManagerComponent would become FileManager as the type for the parameter or attribute.
10. UML attributes with constant stereotype map to a <<CORBAConstant>> in a <<CORBAInterface>> and within the generated IDL interface.
11. Basic types (e.g., Any, Object) map to an equivalent CORBA and IDL type.

The SCA UML model maps to a <<CORBAModule>> and IDL module named CF (Core Framework). The CF IDL module is broken up into multiple files in order to allow for a minimal memory foot print size for the embedded environment.

Table 1: SCA UML to OMG IDL PSM Mapping

UML Representation	OMG IDL Representation
Interface	Interface
exception	exception
Object	Object
Primitive Types (integer, Boolean, string, unlimited natural); others are constructed by layering formatting or constraints on top of primitives	void, boolean, char, double, long, float, long double, long long, short, octet, string, unsigned long long, unsigned short, wchar, wstring

Any	Any data type
Struct	Struct
Sequence	Sequence definition
Type	Typedef
Package	Module
No return value from an operation	void
Enumeration	enum
A constrained integer data type that corresponds to an octet	octet
Attribute	attribute

E-3.6 ATTACHMENTS

N/A