

UNCLASSIFIED

SOFTWARE COMMUNICATIONS ARCHITECTURE SPECIFICATION

APPENDIX F: UNITS OF FUNCTIONALITY AND PROFILES



20 August 2015
Version: 4.1

Prepared by:

Joint Tactical Networking Center (JTNC)
33000 Nixie Way
San Diego, CA 92147-5110

Distribution Statement A - Approved for public release; distribution is unlimited (27 August 2015)

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
4.0.1	Incorporated transition to JTNC and applied SCA 4.0 Errata Sheet v1.0	01 October 2012
4.1<DRAFT>	Incorporated Naming Proposal Changes, Backwards Compatibility Changes, Scalable Components Changes, Scalable Managers Changes	31 December 2014
4.1	Added new Target OE UOF Relocated Backwards Compatible requirements to this appendix Update Normative References ICWG Approved	20 August 2015

TABLE OF CONTENTS

F.1	SCOPE	5
F.1.1	Overview	5
F.2	CONFORMANCE	5
F.2.1	UOF and Profile Conformance on the Part of an SCA Product	5
F.2.2	Sample Conformance Statement	5
F.3	CONVENTIONS.....	6
F.4	NORMATIVE REFERENCES	6
F.5	INFORMATIVE REFERENCES	6
F.6	UNITS OF FUNCTIONALITY - DEFINITIONS	6
F.6.1	Target Operating Environment Units of Functionality	6
F.6.2	BaseComponent Units of Functionality	7
F.6.3	Application Related Component Units of Functionality	9
F.6.4	DeviceComponent Units of Functionality	10
F.7	UNITS OF FUNCTIONALITY – IMPLEMENTATION.....	11
F.7.1	Application Backwards Compatible	11
F.7.1.1	Naming Service.....	11
F.7.1.2	ApplicationManager ReleaseObject Alternative Requirements	11
F.7.1.3	ApplicationFactory Create Alternative Requirements.....	11
F.7.1.4	InstallApplication Exception/Errors	12
F.7.1.5	ApplicationManager Alternative Requirements	12
F.7.1.6	ApplicationFactoryComponent Semantics	12
F.7.1.7	ApplicationFactory Alternative Requirements	12
F.7.1.8	DomainManager Semantics	14
F.8	SCA PROFILES.....	14
F.9	ATTACHMENTS	15

LIST OF FIGURES

Figure 1: BaseComponent Units of Functionality.....	9
Figure 2: Application Component Units of Functionality	10
Figure 3: Device Components Units of Functionality.....	11
Figure 4: SCA Profiles with OE Units of Functionality	15

APPENDIX F UNITS OF FUNCTIONALITY AND PROFILES

F.1 SCOPE

This appendix defines Units of Functionality (UOFs) and Profiles used to achieve scalable levels of conformance with the SCA defined in the main specification.

F.1.1 Overview

A UOF is characterized by a set of related SCA requirements. In many cases a UOF represents a collection of optional requirements to be incorporated or omitted as a whole from a component.

Individual UOFs often provides capabilities that are complementary to those of another UOF. Therefore, this appendix also includes defined profiles. A profile comprises a set of UOFs. The SCA profiles contain related UOFs that are aligned with common real world scenarios.

Profiles are applicable to a platform realization and its resident Framework Control Components. All SCA components and defined profiles may be augmented with additional UOFs. This appendix describes the relationships between implemented components, defined platforms and the primary UOFs. The entire collection of UOFs is identified in the attachment referenced in F.9.

F.2 CONFORMANCE

F.2.1 UOF and Profile Conformance on the Part of an SCA Product

The elements of this specification are not required to be used solely for a particular platform or application. SCA Products are developed in accordance with the requirements defined within the SCA specification. The SCA is written in a style that asserts all of its requirements as mandatory. This specification provides the means by which a conformant SCA Product may be relieved from the responsibility of implementing certain SCA requirements.

Conformance for an SCA Product is defined at a profile level as follows:

- An SCA Product needs to be conformant with all requirements identified by the Units of Functionality that comprise the Profile as defined within this appendix.

Conformance for an SCA Product is defined at a Unit of Functionality level as follows:

- An SCA Product needs to be conformant with all requirements identified by the Unit of Functionality defined within this appendix.

The capabilities of any SCA Product is often extended by incorporating additional Units of Functionality provided the Unit of Functionality conformance rules are adhered to.

F.2.2 Sample Conformance Statement

An SCA Product can be identified as being conformant to a specific version of the SCA and the specific technology that the product realizes.

- "Product A is an SCA conformant waveform ManageableApplicationComponent in accordance with the controllable and configurable UOFs."
- "Product B is an SCA conformant Operating Environment (OE) in accordance with the SCA Medium Profile containing an SCA Lightweight Application Environment Profile conforming POSIX layer and an SCA Full CORBA Profile transfer mechanism."

F.3 CONVENTIONS

N/A

F.4 NORMATIVE REFERENCES

The following documents contain provisions or requirements which by reference constitute requirements of this specification. Applicable versions are as stated.

- [1] OMG Document formal/05-02-02, Lightweight Log Service Specification, Version 1.1, February 2005.
- [2] OMG Document formal/04-10-02, Event Service Specification, Version 1.2, October 2004.
- [3] OMG Document formal/2004-10-03, OMG Naming Service Specification, Version 1.3, October 2004.
- [4] JTNC Software Communications Architecture Specification, Version 2.2.2, FINAL / 15 May 2006.

F.5 INFORMATIVE REFERENCES

N/A

F.6 UNITS OF FUNCTIONALITY - DEFINITIONS

F.6.1 Target Operating Environment Units of Functionality

A Target Operating Environment (OE) on a platform provides a core set of functionality that is available for use by the components that will be deployed on that platform. The Target OE may provide capabilities such as SCA services, support for the underlying RTOS or middleware support. The following UOFs are applicable to the Target OE:

- AEP Provider – provides the SCA Application Environment Profiles (AEP) capability
- AppDeploymentData – allows the application manager to return information about deployed application components
- DeviceMgrDeploymentData – allows the device manager component to return information about deployed platform components

- AppReleasable – provides application manager with a releasing capability
- Application Backwards Compatible – provides capability to manage SCA V2.2.2 compliant applications
- Deployment – provides capability to deploy BasePlatformComponents and ApplicationComponents
- Management Registration – provides the interfaces for registering components to domain and device manager components capability
- Management Un-Registration – provides the interfaces for unregistering components from domain and device manager components capability
- Management Releasable – provides the device manager releasing capability
- CORBA Provider – supports one or more SCA CORBA profiles
- Channel Extension – provides the concepts of platform channels and deployment of applications onto platform channels capability
- Event Channel – provides event channels and the event service capability in the SCA OE
- Log Capable – provides the log service capability within the SCA OE
- Application Installable – provides capability for dynamic application installation and un-installation
- Log Producer – produces logs using the *LogProducer* interface defined in [1]
- Nested Deployment – extends the deployment UOF by providing the capability to deploy installed nested applications and manage instantiated nested applications. Note: The domain profile (accardinality attribute) indicates whether the platform supports a single or multiple application controllers.
- PlatformComponentFactory Deployment – provides the capability of deploying components via a PlatformComponentFactoryComponent.

F.6.2 BaseComponent Units of Functionality

SCA components that are derived from BaseComponent such as, ApplicationManager-Component, DeviceManagerComponent, and DomainManagerComponent may comprise a wide range of UOFs as shown in Figure 1. The following UOFs are applicable to the components that derive from BaseComponent:

- Lifecycle
 - provides life cycle management capability via the *LifeCycle* interface
- AEP Compliant
 - adheres to one of the AEPs defined in SCA Appendix B
- Controllable
 - provides a control capability via the *ControllableInterface* interface
- CORBA Compliant
 - provides a CORBA communication capability that adheres to one of the CORBA profiles defined in SCA Appendix E
- Connectable
 - provides port connection management via the *PortAccessor* interface
- Configurable
 - provides configure and/or query functionality via the *PropertySet* interface, or defines configure and/or query properties
- Event Consumer
 - consumes events using the *PushConsumer* interface defined in [2]
- Event Producer
 - produces events using the *PushSupplier* interface defined in [2]
- Interrogable
 - provides interrogation capability via a component specified interface (i.e. *DeploymentAttributes*, *ComponentIdentifier* or *DeviceAttributes*)
- Log Producer
 - produces logs using the *LogProducer* interface defined in [1]
- Component Registration
 - provides a registration capability via the *ComponentRegistry* interface
- Component Un-Registration
 - provides an un-registration capability via the *FullComponentRegistry* interface
- Releasable
 - provides a release capability via the *Lifecycle* interface
- Testable
 - provides a testing capability via the *TestableInterface* interface and test properties



Figure 1: BaseComponent Units of Functionality

F.6.3 Application Related Component Units of Functionality

Application Related Components have additional optional UOFs beyond those provided by the BaseComponent as shown in Figure 2.

- ApplicationComponent extends BaseComponent UOFs with a Mandatory AEP UOF.
- ManageableApplicationComponent extends ApplicationComponent with a mandatory LifeCycle UOF.
- ApplicationControllerComponent extends ManageableApplicationComponent with a mandatory Controllable UOF.

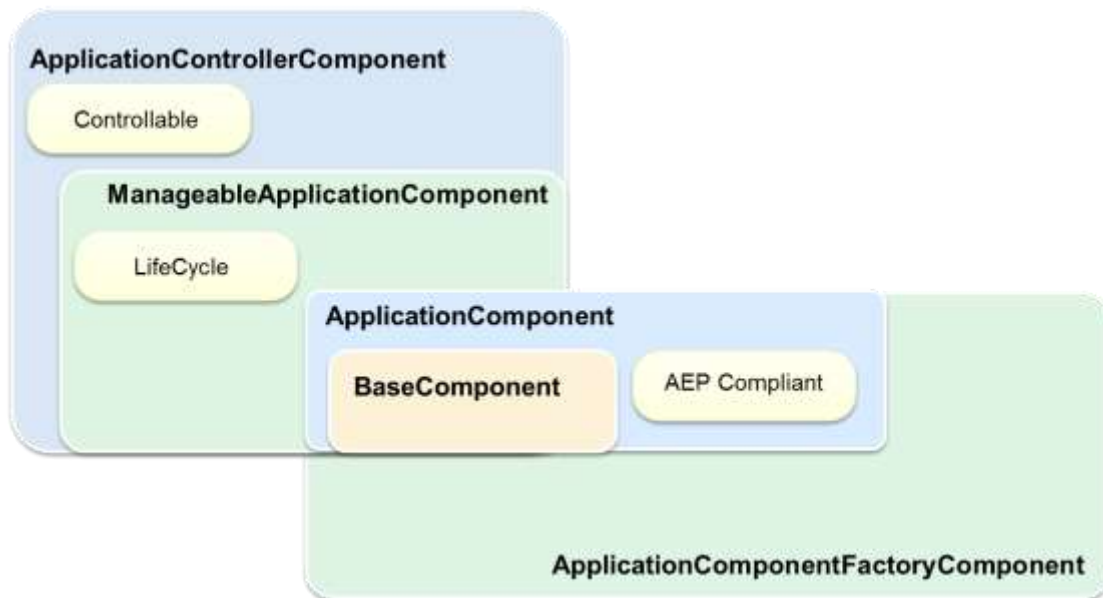


Figure 2: Application Component Units of Functionality

F.6.4 DeviceComponent Units of Functionality

All DeviceComponents have additional optional UOFs beyond those provided by BaseComponent as shown in Figure 3.

- DeviceComponent extends BaseComponent UOFs with a mandatory Lifecycle UOF.
- LoadableDeviceComponent definition extends DeviceComponent with a mandatory Loadable UOF.
- ExecutableDeviceComponent extends the DeviceComponent with mandatory Executable UOF.

The following additional UOFs are applicable to DeviceComponents:

- Loadable – provides the load management capability via the *LoadableInterface* interface
- Executable – provides an execution management capability via the *ExecutableInterface* interface
- Aggregatable – provides an aggregation capability via the *AggregateDeviceAttributes* interface; a parent composite device with children devices
- Allocatable – provides capacity management via the *CapacityManagement* interface and allocation properties that are managed along with usage state
- Manageable – provides administration capability via the *AdministratableInterface* interface and administrative state behavior

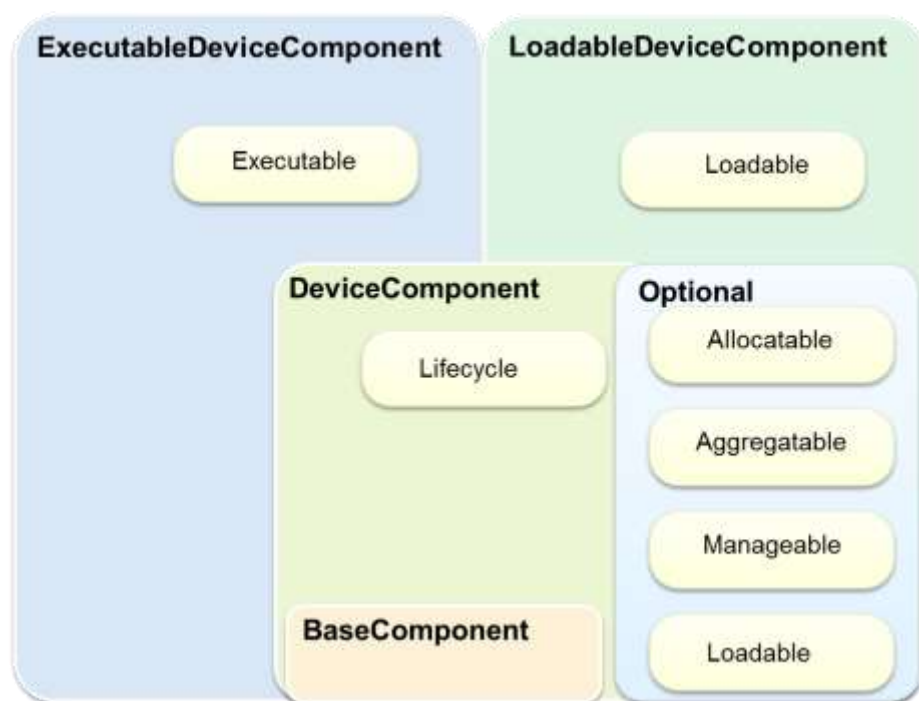


Figure 3: Device Components Units of Functionality

F.7 UNITS OF FUNCTIONALITY – IMPLEMENTATION

F.7.1 Application Backwards Compatible

This section contains the requirements and supporting text that provides the Framework with the ability to manage SCA V2.2.2 compliant applications.

F.7.1.1 Naming Service

SCA544 The OE shall provide a naming capability which implements the CosNaming module *NamingContext* interface operations: *bind*, *bind_new_context*, *unbind*, *destroy*, and *resolve* as defined in the OMG Naming Service Specification [3] using the IDL found in Appendix A of that reference.

F.7.1.2 ApplicationManager ReleaseObject Alternative Requirements

When an SCA V2.2.2 application is being released, the released component adheres to the SCA V2.2.2 [4] requirements.

F.7.1.3 ApplicationFactory Create Alternative Requirements

When an SCA V2.2.2 application is being created, the created application adheres to the SCA V2.2.2 [4] requirements. The following requirements are applicable in those instances in place of their corresponding base requirement:

SCA69* N/A.

SCA70* N/A.

SCA84* The *create* operation shall, in order, initialize all application resources, then establish connections for those resources, and finally configure the application component indicated by the *assemblycontroller* element in the SAD.

F.7.1.4 InstallApplication Exception/Errors

SCA552 The *installApplication* operation shall raise the `ApplicationInstallationError` exception when SCA V2.2.2 application installation is not supported.

F.7.1.5 ApplicationManager Alternative Requirements

When an SCA V2.2.2 application is being released, the released component adheres to the SCA V2.2.2 [4] requirements. The following requirements are applicable in those instances in place of their corresponding base requirement:

SCA158* An `ApplicationManagerComponent` shall delegate the *runTest*, *start*, *stop*, *configure*, and *query* operations to the application's assembly controller as identified by the `AssemblyComponent`'s SAD *assemblycontroller* element.

SCA159* The `ApplicationManagerComponent` shall propagate exceptions raised by the `AssemblyComponent`'s assembly controller.

SCA160* N/A.

SCA161* N/A.

SCA162* N/A.

SCA554 For components that are registered with Naming Service, the *releaseObject* operation shall unbind those components and destroy the associated naming contexts as necessary from the Naming Service.

F.7.1.6 ApplicationFactoryComponent Semantics

SCA555 The *create* operation shall instantiate a SCA V2.2.2 compliant application if the SAD does not have a *sca_version* element. The create operation maps the SCA V2.2.2 application values into the created `ApplicationManagerComponent`.

F.7.1.7 ApplicationFactory Alternative Requirements

When an SCA V2.2.2 application is being created, the created application adheres to the SCA V2.2.2 [4] requirements. The following requirements are applicable in those instances in place of their corresponding base requirement:

SCA68* The *create* operation shall identify valid component-device associations for the application by matching the allocation properties of the application to those of each candidate device, for those application component properties whose *kindtype* is *allocation* and whose *action* element is not *external*.

SCA71* The *create* operation shall allocate capacities to candidate devices of the application component properties whose *kindtype* is *allocation* and whose *action* element is *external*.

SCA72* The *create* operation shall deallocate any capacity allocations on devices that do not satisfy the application component's allocation requirements or that are not utilized due to an unsuccessful application creation.

SCA73* The *create* operation shall load application modules onto devices that have been granted successful capacity allocations and that satisfy the application component's allocation requirements.

SCA76* The *create* operation shall include the mandatory execute parameters Naming Context IOR, Name Binding, and Component Identifier, as described in this section, in the parameters parameter of the ExecutableInterface::execute operation when the CORBA instance's componentinstantiation element of the SAD contains a findcomponent element with a namingservice sub-element.

The execute parameter for the Naming Context IOR is a CF *Properties* type with an id element set to "NAMING_CONTEXT_IOR" and a value element set to the stringified IOR of the naming context to which the component will bind. SCA556 The *create* operation shall create any naming contexts that do not exist but which are required for successful binding to the Naming Context IOR. The structure of the naming context path is "/ DomainName / [optional naming context sequences]". In the naming context path, each "slash" (/) represents a separate naming context.

The Name Binding execute parameter is a CF *Properties* type with an id element set to "NAME_BINDING" and a value element set to a string in the format of "ComponentName_UniqueIdentifier". The ComponentName value is the SAD *componentinstantiation findcomponent namingservice* element's *name* attribute. The UniqueIdentifier is determined by the implementation. The Name Binding parameter is used by the component to bind its object reference to the Naming Context IOR parameter.

SCA542* N/A.

SCA77* N/A.

SCA557 Upon execution of a software module by the *create* operation, a *Resource* or a *ResourceFactory* component shall register with the Naming Service. The *create* operation uses "ComponentName_UniqueIdentifier" to retrieve the component's CORBA object reference from the Naming Context IOR.

The *create* operation obtains a resource in accordance with the SAD via the CORBA Naming Service or a resource factory. The *ResourceFactory* object reference is obtained by using the CORBA Naming Service. SCA83* The *create* operation, when creating a resource from a resource factory, shall pass the *componentinstantiation componentresourcefactoryref* element properties whose *kindtype* element is *factoryparam* as the *qualifiers* parameter to the referenced *ResourceFactory* component's *createResource* operation.

SCA524* N/A.

SCA81* When an application component is created via an executable device, the *create* operation shall pass the values of the *execparam* properties of the *componentinstantiation* *componentproperties* element contained in the SAD, as parameters to the *execute* operation.

SCA85* The *create* operation shall establish connections for an application which are specified in the SAD *domainfinder* element. SCA86* N/A. SCA87* N/A.

SCA90* The *create* operation shall configure the application component indicated by the *assemblycontroller* element in the SAD if that component has properties with a *kindtype* of "configure" and a *mode* of "readwrite" or "writeonly".

SCA98* N/A.

F.7.1.8 DomainManager Semantics

SCA558 The *installApplication* operation shall install a SCA V2.2.2 [4] compliant application.

F.8 SCA PROFILES

In Figure 4 three profiles are shown which are applicable to implementations of the SCA Framework Control Components that are not implemented by an application provider (i.e. DomainManagerComponent, DeviceManagerComponent and ApplicationManagerComponent), they are defined as follows are:

SCA Lightweight Profile

- Supports the AEP Provider and Deployment UOFs.
- Is suited for radio platforms where the hardware modules have a static configuration.
- Provides a minimum set of functionality which is applicable for resource (e.g. SWAP) constrained platforms.

SCA Medium Profile

- Supports the additional Management Registration UOF.
- Is suited for radio platforms where the hardware modules are plug-and-play but cannot be removed.
- Lightweight, but it introduces a configurable, dynamic aspect to the platform.
- May be the most flexible profile in that it provides the lightest weight realization of a platform that supports the deployment model introduced in earlier SCA versions.

SCA Full Profile

- Supports the additional Management Un-Registration and Management Releasable UOFs.
- Is suited for radio platforms where the hardware modules are plug-and-play and can be removed.
- Provides the full breadth of SCA deployment and management capabilities.
- Aligned to support prime power, multi-channel sets.

The SCA profile hierarchy, core profile capabilities and associated optional capabilities are depicted in Figure 4.

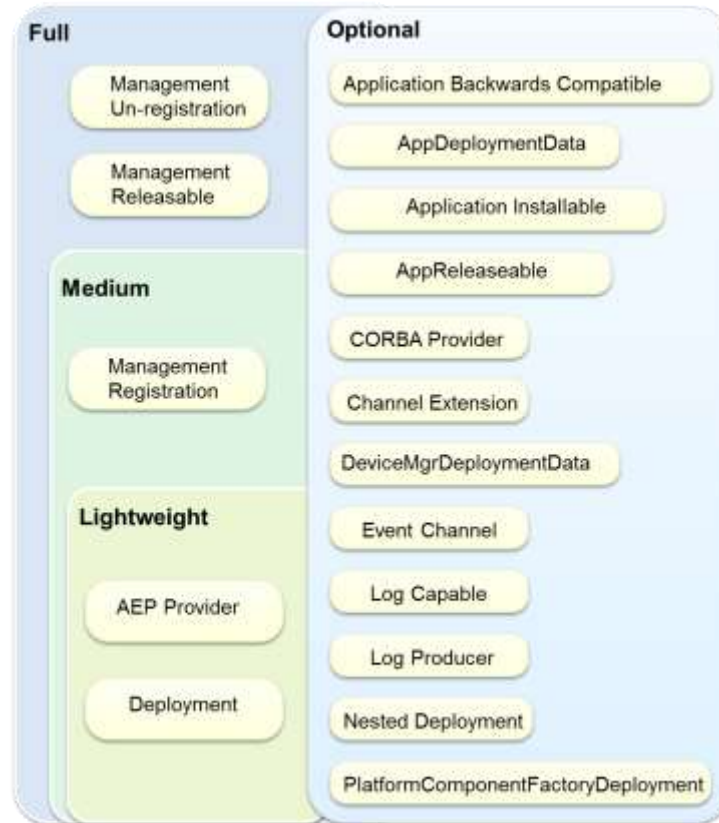


Figure 4: SCA Profiles with OE Units of Functionality

F.9 ATTACHMENTS

This appendix includes the following SCA requirements mapping:

- Appendix F Attachment 1: SCA Conformance Mapping

This attachment includes all the SCA requirements mapped to UOFs and component(s). When a requirement is mapped to UOFs (one or more) that requirement is only applicable if all of the referenced UOFs are implemented. When a requirement is not mapped to any UOFs that requirement is always applicable.