SCA 4.1 Requirements Allocation, Objectives, and Verification Criteria

Working Document WINNF-16-P-0025-V1.0.0

17 March 2017

TERMS, CONDITIONS & NOTICES

This document has been prepared by the SCA 4.1 Compliancy Project to assist The Software Defined Radio Forum Inc. (or its successors or assigns, hereafter "the Forum"). It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the SCA 4.1 Compliancy Project.

Contributors to this document that have submitted copyrighted materials (the Submission) to the Forum for use in this document retain copyright ownership of their original work, while at the same time granting the Forum a non-exclusive, irrevocable, worldwide, perpetual, royalty-free license under the Submitter's copyrights in the Submission to reproduce, distribute, publish, display, perform, and create derivative works of the Submission based on that original work for the purpose of developing this document under the Forum's own copyright.

Permission is granted to the Forum's participants to copy any portion of this document for legitimate purposes of the Forum. Copying for monetary gain or for other non-Forum related purposes is prohibited.

THIS DOCUMENT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS DOCUMENT.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

This document was developed following the Forum's policy on restricted or controlled information (Policy 009) to ensure that that the document can be shared openly with other member organizations around the world. Additional Information on this policy can be found here: http://www.wirelessinnovation.org/page/Policies_and_Procedures

Although this document contains no restricted or controlled information, the specific implementation of concepts contain herein may be controlled under the laws of the country of origin for that implementation. Readers are encouraged, therefore, to consult with a cognizant authority prior to any further development.

Wireless Innovation Forum ™ and SDR Forum ™ are trademarks of the Software Defined Radio Forum Inc.

Table of Contents

List of Tables

SCA 4.1 Requirements Allocation, Objectives and Verification Criteria

1 Introduction

This document contains a table of the requirements contained in the Software Communications Architecture (SCA) version 4.1 [1] and associated appendices, an assigned unique requirement number and the requirement allocation to Operating Environment (OE), Application (AP) and to both OE and AP indicated as “Both”, the objectives and verification criteria for each requirement.
The Requirements Allocation, Objectives and Verification Criteria contained in Table 1 are intended to be used by developers of SCA 4.1 Products (e.g. OEs, Applications, and Tools), and SCA 4.1 Compliance verification activities.

2 Requirement Allocation

The use of “*” as part of the SCA 4.1 Requirement Number, e.g. SCA69*: indicates that the requirement is for SCA 2.2.2 backwards compatibility and is a replacement for the SCA69 SCA 4.1 baseline requirement.

The use of “N/A” as the SCA 4.1 Requirement Text, e.g. SCA69*: indicates that there is no SCA 2.2.2 backwards requirement text associated with the SCA 4.1 Requirement Number, and the SCA 4.1 requirement is not applicable while testing an SCA 2.2.2 component. This occurs when the SCA 4.1 baseline requirement is a new requirement.

3 Requirement Objective

The requirement objectives provide the purpose of the verification criteria.

4 Verification Criteria

The verification criteria are written as pass/fail statements. An implementation is compliant with the requirement if it passes the verification criteria. An implementation is noncompliant with the requirement if it fails the verification criteria.

Table 1 SCA 4.1 Requirements Allocation, Objectives and Verification Criteria

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA1	The OE and related file systems shall support a maximum filename length of 40 characters and a maximum pathname length of 1024 characters.	OE	3.1.1	Establishes the maximum filename and pathname length that a file system is required to support.	1. OE and related file systems accept filenames of 40 or fewer characters. 2. OE and related file systems accept pathnames of 1024 or fewer characters.
SCA451	The OE shall provide the functions and options designated as mandatory by a profile defined in Appendix B.	OE	3.1.1	Provide a standard set of Operating System functions and standard C library functions	Verify each of the mandatory functions listed in Appendix B is available in the selected profile.
SCA452	The OE shall provide a transfer mechanism that, at a minimum, provides the features specified in Appendix E for the specific platform technology implemented.	OE	3.1.2	Specifies a standard set of transport mechanisms provided by the OE	As specified by the transfer mechanism PSM listed in Appendix E, Section 9.2, for the platform technology and its profile
SCA453	The log service shall conform to the OMG Lightweight Log Service Specification [1].	OE	3.1.2.1	Establishes requirements for a SCA Log Service.	Logging functionality conforms to the OMG Lightweight Log Service Specification.
SCA3	The OE shall provide two standard event channels: Incoming Domain Management and Outgoing Domain Management.	OE	3.1.2.2.1	To ensure the OE provides standard event channels.	The OE provides the Incoming Domain Management and Outgoing Domain Management event channels (i.e. "IDM_Channel" and "ODM_Channel").

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA454	The OE shall provide an event capability which implements the PushConsumer and PushSupplier interfaces of the CosEventComm module as described in OMG Event Service Specification [2] consistent with the IDL found in that specification.	OE	3.1.2.2.1	To ensure the OE provides the PushConsumer and PushSupplier interfaces of the OMG Event Service specification.	The OE provides implementation for the PushConsumer and PushSupplier interfaces as described in the OMG Event Service Specification IDL.
SCA386	The createComponent operation shall create a component if no component exists for the given componentId.	Both	3.1.3.1.1.1.5.1.3	To ensure a component is created by a componentFactory if it does not already exist.	1. A Component by the given componentId does not previously exist. 2. A new component is returned by the ComponentFactory for the componentId provided.
SCA387	The createComponent operation shall assign the given componentId to a new component.	Both	3.1.3.1.1.1.5.1.3	To ensure a new component can be identified by the given componentId.	The created component's ComponentType identifier field equals that of the componentId parameter in the create call.
SCA388	The createComponent operation shall return a CF::ComponentType structure.	Both	3.1.3.1.1.1.5.1.4	To ensure that ComponentFactory returns the basic elements of a created component.	1. The returned structure complies with CF::ComponentType structure declaration. 2. The content of its fields is consistent with the domain profile content.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA389	The createComponent operation shall raise the CreateComponentFailure exception when it cannot create the component or the component already exists.	Both	3.1.3.1.1.1.5.1.5	1. The created component implements the CF::ComponentIdentifier interface. 2. The created component's identifier field equals that of the componentId parameter in the create call.	1. The component already exists or an error occurs which prevents the creation of a new component. 2. The CreateComponentFailure exception is raised.
SCA420	A BaseComponent shall implement a 'configure' kind of property with a name of PRODUCER_LOG_LEVEL.	Both	3.1.3.1.2.1.3	To ensure that a BaseComponent that is a log producer can filter log records to be sent to a Log Service.	The BaseComponent that is a log producer implements a PRODUCER_LOG_LEVEL property with kindtype of 'configure'.
SCA421	A BaseComponent shall output only those log records to a log service that correspond to enabled log level values in the PRODUCER_LOG_LEVEL attribute.	Both	3.1.3.1.2.1.3	To ensure a BaseComponent only outputs log records with the enabled log level.	The BaseComponent only outputs log records containing log level values enabled by the PRODUCER_LOG_LEVEL property.
SCA423	A BaseComponent shall operate normally in the case where the connections to a log service are nil or an invalid reference.	Both	3.1.3.1.2.1.3	To ensure a BaseComponent does not demonstrate any erroneous or unexpected behavior when connections to a log service are not present or invalid.	1. A log service connection points to a nil or an invalid reference. 2. The component operates (aside from logging behavior) identically to when a logging reference is valid.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA424	A BaseComponent that produces events shall implement the CosEventComm::PushSupplier interface and use the CosEventComm::PushConsumer interface for generating the events.	Both	3.1.3.1.2.1.3	To provide a standard mechanism to push message(s) to the event channel.	1. The BaseComponent implements the CosEventComm::PushSupplier interface. 2. The BaseComponent invokes the PushConsumer's push operation.
SCA425	A producer BaseComponent shall not forward or raise any exceptions when the connection to a CosEventComm::PushConsumer is a nil or invalid reference.	Both	3.1.3.1.2.1.3	To define the behavior of an event producer BaseComponent when its event channel connection is nil or invalid.	A BaseComponent does not raise an exception when its connection to a CosEventComm::PushConsumer is nil or invalid.
SCA444	A BaseComponent (e.g., ManageableApplicationComponent, DomainManagerComponent, etc.) that consumes events shall implement the CosEventComm::PushConsumer interface.	Both	3.1.3.1.2.1.3	To provide a standard mechanism to receive/consume message(s) from the event channel.	A BaseComponent that consumes events implements the CosEventComm::PushConsumer interface.
SCA518	The releaseObject operation shall disconnect any ports that are still connected.	Both	3.1.3.1.2.1.3	Ensures a component that is being removed has all of its ports disconnected.	The releaseObject operation disconnects any remaining port connections.
SCA426	A BaseComponent shall realize the ComponentIdentifier interface.	Both	3.1.3.1.2.1.4	To ensure a BaseComponent provides an identifier.	The BaseComponent inherits the ComponentIdentifier interface.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA427	A BaseComponent shall be associated with a domain profile file.	Both	3.1.3.1.2.1.4	To ensure a domain profile file (i.e. SPD) exists for the BaseComponent.	An SPD exists and is associated with the BaseComponent.
SCA428	A BaseComponent shall provide a test implementation for all properties whose kindtype is "test" as defined in its descriptor files.	Both	3.1.3.1.2.1.4	To ensure each BaseComponent that is testable has a test implementation for each property with a kindtype value of "test".	1. Properties of kindtype value of "test" for a BaseComponent that is testable are defined in its descriptor files. 2. A implementation is provided in the runTest method for each property identified in step 1.
SCA429	A BaseComponent shall configure or retrieve query values for all properties whose kindtype is "configure" as defined in its domain profile.	Both	3.1.3.1.2.1.4	To provide capability to configure and query a component's properties.	1. The properties defined within the component's domain profile with a kindtype of "configure" can be accessed by the configure or query operations. 2. Properties with a mode of writeonly or readwrite can be modified via CF::PropertySet::configure. 3. Properties with a mode of readonly and readwrite can be queried via CF::PropertySet::query.
SCA430	A BaseComponent shall supply ports for all the ports defined in its domain profile.	Both	3.1.3.1.2.1.4	To ensure a BaseComponent has implemented the defined ports.	All component's ports specified in the SCD can be accessed via the CF::PortAccessor interface.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA432	A BaseComponent shall realize the LifeCycle interface.	Both	3.1.3.1.2.1.4	To ensure a BaseComponent provides a standard mechanism for its life cycle to be managed.	The BaseComponent inherits the LifeCycle interface.
SCA433	A BaseComponent shall realize the ControllableInterface interface to provide overall management control of the component.	Both	3.1.3.1.2.1.4	To ensure a BaseComponent provides a standard mechanism to control a component.	The BaseComponent inherits the ControllableInterface interface.
SCA545	A BaseComponent shall realize the PropertySet interface to configure and query its properties.	Both	3.1.3.1.2.1.4	To ensure a BaseComponent provides a standard mechanism for its properties to be configured and queried.	The BaseComponent inherits the PropertySet interface.
SCA546	A BaseComponent shall realize the TestableInterface interface to define and utilize its test properties.	Both	3.1.3.1.2.1.4	To ensure a BaseComponent provides a standard mechanism for it to be tested.	The BaseComponent inherits the TestableInterface interface.
SCA547	A BaseComponent shall realize the PortAccessor interface as a proxy for its uses and provides ports.	Both	3.1.3.1.2.1.4	To ensure a BaseComponent provides a standard mechanism for the BaseComponent to expose its uses and provides ports.	The BaseComponent inherits the PortAccessor interface.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA548	A BaseComponent shall implement its optional composition relationships via inheritance.	Both	3.1.3.1.2.1.4	To ensure a BaseComponent uses inheritance to realize its optional compositional interface(s).	The BaseComponent inherits all optional composition interfaces (i.e. SCA Figure 3-4) specified in its domain profile.
SCA540	Each BaseFactoryComponent shall support the mandatory Component Identifier execute parameter as described in section 3.1.3.3.1.3.5.1, in addition to their user-defined execute properties in the component's SPD.	Both	3.1.3.1.2.2.3	To ensure BaseFactoryComponents can accept all properly formatted execute parameters.	The FactoryComponent accepts the Component Identifier and any other execute parameters defined in the SPD.
SCA541	Each executable BaseFactoryComponent shall set its identifier attribute using the Component Identifier execute parameter.	Both	3.1.3.1.2.2.3	Defines a standard mechanism for setting the identifier of a BaseFactoryComponent that was executed by an ExecutableDevice.	BaseFactoryComponent identifier matches the value of the COMPONENT_IDENTIFIER execute parameter.
SCA574	The releaseObject operation shall release all component instances created by the BaseFactoryComponent.	Both	3.1.3.1.2.2.3	To ensure BaseFactoryComponent releases components that it created when it is released.	The BaseFactoryComponent's releaseObject operation releases each component created by the BaseFactoryComponent that has not been previously released.
SCA413	A BaseFactoryComponent shall realize the ComponentFactory interface.	Both	3.1.3.1.2.2.4	To ensure a BaseFactoryComponent provides a standard mechanism to create component(s).	The BaseFactoryComponent inherits the ComponentFactory interface.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA414	A BaseFactoryComponent shall fulfill the BaseComponent requirements.	Both	3.1.3.1.2.2.4	To ensure a BaseFactoryComponent satisfies the requirements of a BaseComponent.	Will be satisfied by the verification of BaseComponent requirements.
SCA549	A BaseFactoryComponent shall realize the LifeCycle interface.	Both	3.1.3.1.2.2.4	To ensure a BaseFactoryComponent provides a standard mechanism for its life cycle to be managed.	The BaseFactoryComponent inherits the LifeCycle interface.
SCA6	The readonly identifier attribute shall return the instance-unique identifier for a component.	Both	3.1.3.2.1.1.4.1	To provide a standard mechanism to retrieve the component's identifier.	The identifier attribute returns the component's identifier.
SCA7	The connectUsesPorts operation shall make the connection(s) to the component identified by its input portConnections parameter.	Both	3.1.3.2.1.2.5.1.3	To ensure a component connects its uses ports as specified in its input portConnections parameter.	Connections are established between an application's component uses ports and the ports specified in the input portConnections parameter.
SCA519	The connectUsesPorts operation shall disconnect any connections it formed if any connections in the input portConnections parameter cannot be successfully established.	Both	3.1.3.2.1.2.5.1.3	Ensure that all of a component's formed connections are disconnected (i.e. pre-connection state) if any connection in the input portConnections cannot be successfully made.	When one of a component's port connections specified in the input portConnections parameter cannot be established, all specified connections in the portConnections parameter re disconnected.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA8	The connectUsesPorts operation shall raise the InvalidPort exception when the input portConnections parameter provides an invalid connection for the specified port.	Both	3.1.3.2.1.2.5.1.5	To define a standard mechanism for reporting an error when the connectUsesPorts operation is unable to connect the specified port(s).	1. The portConnections input parameter contains invalid data (connectionId, portName, portReference) or the number of connections exceeds the maximum limit for the port. 2. The InvalidPort exception is raised.
SCA10	The disconnectPorts operation shall break the connection(s) to the component identified by the input portDisconnections parameter.	Both	3.1.3.2.1.2.5.2.3	To provide a standard mechanism to disconnect the specified connections from the component uses port(s).	The component uses port(s) connection(s) to the specified input portDisconnections parameter no longer exist.
SCA11	The disconnectPorts operation shall release all ports if the input portDisconnections parameter is a zero length sequence.	Both	3.1.3.2.1.2.5.2.3	To provide a standard mechanism to disconnect all connections from the component uses port(s).	1. The portDisconnections parameter is zero length. 2. All component uses port(s) connection(s) no longer exist.
SCA12	The disconnectPorts operation shall raise the InvalidPort exception when the input portDisconnections parameter provides an unknown connection to the PortAccessor's component.	Both	3.1.3.2.1.2.5.2.5	To ensure an exception is raised when the input parameter is an unknown connection.	The disconnectPorts operation raised the InvalidPort exception when the input portDisconnections parameter contained an unknown connection.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA13	The getProvidesPorts operation shall return the object references that are associated with the input port names and the connectionIds.	Both	3.1.3.2.1.2.5.3.4	Return the port references associated with the input parameters (i.e. portNames). The connectionIds are not needed.	The port(s) associated with the input port name(s) are populated within the reference(s) that correspond to the provides port object(s).
SCA14	The getProvidesPorts operation shall raise an InvalidPort exception when the input portConnections parameter requests undefined connection(s).	Both	3.1.3.2.1.2.5.3.5	To ensure an exception is raised when the input port name(s) is/are undefined.	The InvalidPort exception is raised when an input port name is not defined in the SCD.
SCA15	The initialize operation shall raise an InitializeError exception when an initialization error occurs.	Both	3.1.3.2.1.3.5.1.5	To provide a standard mechanism for reporting errors encountered during the component initialize operation.	The InitializeError exception is raised when an error(s) occurs during the component initialize operation.
SCA16	The releaseObject operation shall release all internal memory allocated by the component during the life of the component.	Both	3.1.3.2.1.3.5.2.3	Release all memory allocated by the component.	All dynamic memory allocated by the component has been deallocated.
SCA17	The releaseObject operation shall tear down the component and release it from the operating environment.	Both	3.1.3.2.1.3.5.2.3	Ensures a component has been released from the operating environment.	All components have been deactivated.
SCA18	The releaseObject operation shall raise a ReleaseError exception when a release error occurs.	Both	3.1.3.2.1.3.5.2.5	To ensure an exception is raised when a release error occurs.	The releaseObject operation raises a ReleaseError exception when a release error occurs.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA19	The runTest operation shall use the input testId parameter to determine which of its predefined test implementations should be performed.	Both	3.1.3.2.1.4.5.1.3	To ensure each component test is discriminated by the input testId value.	1. Identify the testIds of propertyType Test in the component Domain Profile. 2. A test implementation is invoked for each unique input testId parameter of runTest.
SCA21	The runTest operation shall return the result(s) of the test in the testValues parameter.	Both	3.1.3.2.1.4.5.1.3	Ensures results of the runTest operation are returned using the testValues parameter.	The runTest operation returned results in the testValues parameter.
SCA23	The runTest operation shall raise the UnknownTest exception when there is no underlying test implementation that is associated with the input testId given.	Both	3.1.3.2.1.4.5.1.5	To ensure an exception is raised when an underlying test implementation does not exist for the input testId.	The runTest operation raises the UnknownTest exception when a test implementation does not exist for the input testId.
SCA24	The runTest operation shall raise the CF::UnknownProperties exception when the input parameter testValues contains any CF::DataTypes that are not known by the component's test implementation or any values that are out of range for the requested test.	Both	3.1.3.2.1.4.5.1.5	To ensure an exception is raised when input testValues contain CF::DataTypes not known by the component's test implementation or that are out of range.	The runTest operation raised the UnknownProperties exception when the input testValues contained CF::DataTypes not known by the component or values that are out of range.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA25	The exception parameter invalidProperties shall contain the invalid testValues properties id(s) that are not known by the component or the value(s) are out of range.	Both	3.1.3. 2.1.4. 5.1.5	Ensure CF::UnknownProperties exception raised contains the invalid parameter(s) that were not known by the component or value(s) that are out of range.	1. The runTest operation is invoked with an input testValues parameter that contained invalid property Ids or value(s) out of range. 2. UnknownProperties exception is raised and the invalidProperties parameter is populated with either the invalid testValues properties id(s) or the value(s) which is/are out of range.
SCA26	The configure operation shall assign values to the properties as indicated in the input configProperties parameter.	Both	3.1.3. 2.1.5. 5.1.3	To ensure assignment of configuration properties from the input configProperties parameter of the configure operation.	The configuration properties are modified by the corresponding properties specified in the input configProperties parameter via CF::PropertySet::configure.
SCA27	The configure operation shall raise a PartialConfiguration exception when some configuration properties were successfully set and some configuration properties were not successfully set.	Both	3.1.3. 2.1.5. 5.1.5	To ensure that a PartialConfiguration exception is raised when some but not all configuration properties were successfully set.	1. At least one but not all of the configuration properties can be successfully set from the properties specified in the input configProperties parameter. 2. The invocation of the configure operation raises the PartialConfiguration exception.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA28	The configure operation shall raise an InvalidConfiguration exception when a configuration error occurs and no configuration properties were successfully set.	Both	3.1.3. 2.1.5. 5.1.5	To ensure that an InvalidConfiguration exception is raised when none of the input configuration properties could be successfully set.	1. None of the configuration properties can be successfully set from the properties specified in the input configProperties parameter. 2. The invocation of the configure operation raises the InvalidConfiguration exception.
SCA29	The query operation shall return all component properties when the inout parameter configProperties is zero size.	Both	3.1.3. 2.1.5. 5.2.3	To return all queryable (i.e. readonly or readwrite properties of kindtype configure) properties of the component when the configProperties parameter is zero size.	1. The configProperties parameter is zero length. 2. The query operation returns all of the queryable properties (i.e. readonly or readwrite properties of kindtype configure) for the component.
SCA30	The query operation shall return only those id/value pairs specified in the configProperties parameter if the parameter is not zero size.	Both	3.1.3. 2.1.5. 5.2.3	To ensure the query operation returns only those properties (i.e. id/value pairs) of the component specified in the configProperties parameter.	The query operation returns the properties (i.e. id/value pairs) that correspond to valid input queryable properties specified in the input configProperties parameter.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA31	The query operation shall raise the CF::UnknownProperties exception when one or more properties being requested are not known by the component.	Both	3.1.3.2.1.5.5.2.5	To ensure the CF::UnknownProperties exception is raised when one or more property id(s) being requested are not known by the component.	A CF::UnknownProperties exception is raised when an id of one or more of the input properties is not known by the component.
SCA32	The readonly started attribute shall return the component's started value.	Both	3.1.3.2.1.6.4.1	To provide a standard mechanism to determine if a component has been started.	The component's "started" attribute value is returned.
SCA33	The start operation shall set the started attribute to a value of TRUE.	Both	3.1.3.2.1.6.5.1.3	To indicate the component is successfully started.	The started attribute return value is TRUE after the start operation is successfully invoked and the stop operation has not been invoked after the start operation.
SCA34	The start operation shall raise the StartError exception if an error occurs while starting the component.	Both	3.1.3.2.1.6.5.1.5	To ensure the StartError exception is raised when an error occurs while starting the component.	A StartError exception is raised when the start operation cannot complete with success.
SCA36	The stop operation shall set the started attribute to a value of FALSE.	Both	3.1.3.2.1.6.5.2.3	To indicate the component has been successfully stopped.	The started attribute return value is FALSE after the stop operation is successfully invoked and the start operation has not been invoked after the stop operation.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA37	The stop operation shall raise the StopError exception if an error occurs while stopping the component.	Both	3.1.3.2.1.6.5.2.5	To ensure the StopError exception is raised when an error occurs while stopping the component.	A StopError exception is raised when the stop operation cannot complete with success.
SCA169	Each ApplicationComponent shall be accompanied by an SPD file per section 3.1.3.6.	AP	3.1.3.2.2.1.4	To ensure an ApplicationComponent has its required domain profile file(s).	The ApplicationComponent has a SPD and all of its referenced domain profile file(s).
SCA173	An ApplicationComponent shall be limited to using the mandatory OS services designated in Appendix B as specified in the SPD.	AP	3.1.3.2.2.1.4	To ensure an ApplicationComponent is portable (i.e. limited to using the mandatory OS services designated in Appendix B) regarding the OS services it uses.	The ApplicationComponent does not use the “NRQ" features designated in the AEP.
SCA457	An ApplicationComponent shall be limited to using transfer mechanisms features specified in Appendix E for the specific platform technology implemented.	AP	3.1.3.2.2.1.4	To ensure an ApplicationComponent is portable regarding the transfer mechanisms it uses (i.e. as specified in Appendix E).	The ApplicationComponent uses a transport mechanism identified within SCA Appendix E.
SCA551	An ApplicationComponent shall fulfill the BaseComponent requirements.	AP	3.1.3.2.2.1.4	To ensure an ApplicationComponent satisfies the requirements of a BaseComponent.	Will be satisfied by the verification of BaseComponent requirements.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA168	Each executable ManageableApplicationComponent shall set its identifier attribute using the Component Identifier execute parameter.	AP	3.1.3.2.2.2.3	To provide a standard mechanism for setting the identifier of a ManageableApplicationComponent that was executed by an ExecutableDevice.	ManageableApplicationComponent identifier matches the value of the COMPONENT_IDENTIFIER execute parameter.
SCA455	Each ManageableApplicationComponent shall support the mandatory Component Identifier execute parameter as described in section 3.1.3.3.1.3.5.1, in addition to their user-defined execute properties in the component's SPD.	AP	3.1.3.2.2.2.3	To ensure the created ManageableApplicationComponent can accept all expected execute parameters (i.e. Component Identifier in addition to the user-defined execute properties in the component's SPD).	The ManageableApplicationComponent accepts the Component Identifier executable parameter (i.e. COMPONENT_IDENTIFIER as described in 3.1.3.3.2.2.3) and any other execute parameters defined in the component's SPD.
SCA456	Each executable ManageableApplicationComponent shall accept executable parameters as specified in section 3.1.3.4.1.6.5.1.3 (ExecutableInterface::execute).	AP	3.1.3.2.2.2.3	To ensure an executable ManageableApplicationComponent employs the specified format to obtain its execute parameters.	The ManageableApplicationComponent accepts the input executable parameters formatted as specified in section 3.1.3.4.1.6.5.1.3.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA82	A ManageableApplicationComponent shall register via the ComponentRegistry::registerComponent operation when a COMPONENT_REGISTRY_IOR parameter is supplied.	AP	3.1.3.2.2.2.3	To ensure a ManageableApplicationComponent registers when a COMPONENT_REGISTRY_IOR parameter is supplied.	1. COMPONENT_REGISTRY_IOR is supplied to the ManageableApplicationComponent. 2. The ManageableApplicationComponent registers with the ComponentRegistry::registerComponent operation using the supplied COMPONENT_REGISTRY_IOR.
SCA520	A ManageableApplicationComponent shall fulfill the ApplicationComponent requirements.	AP	3.1.3.2.2.2.4	To ensure an ManageableApplicationComponent satisfies the requirements of a ApplicationComponent.	Will be satisfied by the verification of ApplicationComponent requirements.
SCA166	A ManageableApplicationComponent shall perform file access through the FileSystem and File interfaces.	AP	3.1.3.2.2.2.4	To ensure ManageableApplicationComponent accesses files only through the SCA FileSystem and File interfaces rather than directly using underlying OS.	ManageableApplicationComponent accesses files only through the File or FileSystem interfaces.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA167	All ManageableApplicationComponent processes shall have a handler registered for the AEP SIGQUIT signal.	AP	3.1.3.2.2.2.4	Provides a standard mechanism for handling a SIGQUIT signal by the ManageableApplicationComponent processes.	Each ManageableApplicationComponent process has a signal handler registered for the AEP SIGQUIT signal.
SCA550	A ManageableApplicationComponent shall realize the LifeCycle interface.	AP	3.1.3.2.2.2.4	To ensure a ManageableApplicationComponent provides a standard mechanism for its life cycle to be managed.	The ManageableApplicationComponent inherits the LifeCycle interface.
SCA175	An ApplicationControllerComponent shall fulfill the ManageableApplicationComponent requirements.	AP	3.1.3.2.2.3.4	To ensure an ApplicationControllerComponent satisfies the requirements of a ManageableApplicationComponent.	Will be satisfied by the verification of ManageableApplicationComponent requirements.
SCA176	An ApplicationControllerComponent shall realize the ControllableInterface interface.	AP	3.1.3.2.2.3.4	To ensure an ApplicationControllerComponent provides a standard mechanism to control an application.	The ApplicationControllerComponent inherits the ControllableInterface interface.
SCA415	The ApplicationComponentFactoryComponent shall only deploy ApplicationComponents.	AP	3.1.3.2.2.4.4	To ensure the ApplicationComponentFactoryComponent only deploys ApplicationComponents.	The components created by the ApplicationComponentFactoryComponent are ApplicationComponents.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA521	An ApplicationComponentFactoryComponent shall fulfill the BaseFactoryComponent requirements.	AP	3.1.3.2.2.4.4	To ensure an ApplicationComponentFactoryComponent satisfies the requirements of a BaseFactoryComponent.	Will be satisfied by the verification of BaseFactoryComponent requirements.
SCA522	An ApplicationComponentFactoryComponent shall fulfill the ApplicationComponent requirements.	AP	3.1.3.2.2.4.4	To ensure an ApplicationComponentFactoryComponent satisfies the requirements of an ApplicationComponent.	Will be satisfied by the verification of ApplicationComponent requirements.
SCA155	An AssemblyComponent shall be accompanied by the appropriate Domain Profile files per section 3.1.3.6.	AP	3.1.3.2.2.5	To ensure an AssemblyComponent has its required domain profile file(s).	The AssemblyComponent has a SAD and all of its referenced domain profile file(s).
SCA156	An AssemblyComponent shall have at least one ApplicationControllerComponent.	AP	3.1.3.2.2.5	To ensure a AssemblyComponent has a consistent interface to control the application.	An AssemblyComponent has at least one reference to an ApplicationControllerComponent within its Domain Profile files.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA41	The readonly name attribute shall return the name of the created application.	OE	3.1.3.3.1.1.4.1	To provide a standard mechanism to retrieve the instance name (as provided by the name parameter of the ApplicationFactory::create operation) of an application created by the ApplicationFactoryComponent.	1. A name for the application is provided by the name parameter in the create operation of the ApplicationFactoryComponent. 2. The name attribute of the application returned by the ApplicationFactoryComponent is equal to the name provided in the create operation.
SCA42	The ApplicationManager::releaseObject operation shall release each application component by utilizing the LifeCycle::releaseObject operation.	OE	3.1.3.3.1.1.5.1.3	To ensure the ApplicationManager releases each application component	ApplicationManager::releaseObject causes LifeCycle::releaseObject to be invoked on each application component.
SCA43	The ApplicationManager::releaseObject operation shall terminate the processes / tasks on allocated ExecutableDeviceComponents belonging to each application component.	OE	3.1.3.3.1.1.5.1.3	Terminate processes / tasks for application components allocated on executable Device components.	Processes/tasks/threads associated with the application components on a specified ExecutableDevice no longer exist.
SCA44	The ApplicationManager::releaseObject operation shall unload each application component instance from its allocated LoadableDeviceComponent.	OE	3.1.3.3.1.1.5.1.3	The ApplicationManager reclaims the resources that are consumed by an application component.	ApplicationManager::releaseObject contains a call to CF::LoadableInterface::unload for each application component instance.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA45	The ApplicationManager::releaseObject operation shall deallocate the DeviceComponent capacities that were allocated during application creation.	OE	3.1.3. 3.1.1. 5.1.3	Ensures DeviceComponent capacities that were allocated during Application creation are deallocated.	Implementation of ApplicationManager::releaseObject calls CF::CapacityManagement::deallocateCapacity on the DeviceComponents upon which its application components were deployed with properties that correspond to the identifiers and values of each component's allocation properties defined within its domain profile.
SCA46	The ApplicationManager::releaseObject operation shall release all object references to the components making up the application.	OE	3.1.3. 3.1.1. 5.1.3	Ensure all object references to the application's components are released.	The ApplicationManager releaseObject releases the object reference to the components it deployed as specified within the Application's domain profile.
SCA47	The ApplicationManager::releaseObject operation shall disconnect ports (including an Event Service's event channel consumers and producers) that were previously connected based upon the application's associated SAD.	OE	3.1.3. 3.1.1. 5.1.3	Ensures all of the ports that were connected as specified in the Application's SAD file are disconnected.	The releaseObject operation disconnects all ports associated with connections specified in the Application's SAD file.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA49	The ApplicationManager::releaseObject operation shall, upon successful application release, write an ADMINISTRATIVE_EVENT log record.	OE	3.1.3. 3.1.1. 5.1.3	To ensure the ApplicationManager::releaseObject operation that is successful is logged as an ADMINISTRATIVE_EVENT.	1. Log Service is supported. 2. DomainManagerComponent's log contains an ADMINISTRATIVE_EVENT log record when an ApplicationManager::releaseObject operation is successful.
SCA50	The ApplicationManager::releaseObject operation shall, upon unsuccessful application release, write a FAILURE_ALARM log record.	OE	3.1.3. 3.1.1. 5.1.3	To ensure the ApplicationManager::releaseObject operation that is unsuccessful is logged as a FAILURE_ALARM.	1. Log Service is supported. 2. DomainManagerComponent's log contains a FAILURE_ALARM log record when an ApplicationManager::releaseObject operation is unsuccessful.
SCA51	The ApplicationManager::releaseObject operation shall send a ComponentChangeEventType event to the Outgoing Domain Management event channel upon successful release of an application.	OE	3.1.3. 3.1.1. 5.1.3	To ensure the ApplicationManager sends a ComponentChangeEventType event to the Outgoing Domain Management event channel when an application is successfully released.	The ApplicationManager sends a ComponentChangeEventType event to the Outgoing Domain Management event channel upon successful release of an application. The ComponentChangeEventType contains following: 1. The producerId is the identifier attribute of the releasing ApplicationManagerComponent. 2. The ComponentChange is REMOVED. 3. The domainComponent is the released ApplicationManagerComponent's CF::ComponentType.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA53	The getProvidesPorts operation shall return the object references that are associated with the input provides port names for the application external ports as identified in the associated SAD (ApplicationManagerComponent's profile).	OE	3.1.3. 3.1.1. 5.2.4	Ensures the object references for application external provides ports can be obtained.	1. Provides ports names associated to application's external ports are provided to the ApplicationManager::getProvides Ports operation. 2. The ApplicationManager::getProvides Ports operation returns the requested ports.
SCA55	The connectUsesPorts operation shall make a connection to the application components by input portConnections parameter, which identifies the application external uses ports to be connected to.	OE	3.1.3. 3.1.1. 5.3.3	To ensure uses port connections can be established for application's external uses ports.	Connections are established between an application's external uses ports and the ports specified in the input portConnections parameter.
SCA523	The connectUsesPorts operation shall disconnect any connections it formed if any connections in the input portConnections parameter cannot be successfully established.	OE	3.1.3. 3.1.1. 5.3.3	Ensure that all of an application's formed connections are disconnected (i.e. pre-connection state) if any connection in the input portConnections cannot be successfully made.	When one of an application's external port connections specified in the input portConnections parameter cannot be established, all specified connections in the portConnections parameter are disconnected.
SCA58	The disconnectPorts operation shall break the connection(s) to the application external ports as identified by the connectionIds referenced in the input portDisconnections parameter.	OE	3.1.3. 3.1.1. 5.4.3	To provide a standard mechanism to disconnect the specified connections from the application external uses port(s).	The application external uses port(s) connection(s) to the specified input portDisconnections parameter no longer exist.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA59	The disconnectPorts operation shall release all external ports if the input portDisconnections parameter is a zero length sequence.	OE	3.1.3. 3.1.1. 5.4.3	To provide a standard mechanism to disconnect all connections from the application's external uses port(s).	1. The portDisconnections parameter is zero length. 2. All the application external uses port(s) connection(s) no longer exist.
SCA64	The deployedComponents attribute shall return the list of BaseComponents that have been successfully deployed or a sequence length of zero if no BaseComponents have been deployed.	OE	3.1.3. 3.1.2. 4.1	To ensure the DeviceManagerComponent provides a reference to a list of its deployed BaseComponent(s).	1. The deployedComponents list contains all the BaseComponents as specified in the DCD that have successfully deployed. 2. The deployedComponents list returns a zero length list when no BaseComponents have been deployed.
SCA65	The readonly name attribute shall return the name of the application instantiated by an application factory.	OE	3.1.3. 3.1.3. 4.1	To provide a standard mechanism to retrieve the name as specified in the SAD file of an application that an application factory creates.	The value of ApplicationFactory's name attribute is identical to the softwareassembly element name attribute of the application's SAD file.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA69	The create operation shall use the allocation property values contained in the input deploymentDependencies parameter over the application deploymentdependencies elements or components dependency allocation properties of application factory profile when they reference the same property.	OE	3.1.3. 3.1.3. 5.1.3	To establish the precedence relationships of the values given to allocationproperties defined in application components.	1. The deploymentDependencies input parameter provided to the create operation provides values for some allocation properties defined in the application factory's profile. 2. The values provided by the input parameter takes precedence over the values defined for these properties in the application factory's profile.
SCA70	The create operation shall pass the input deploymentDependencies parameter for nested assemblyinstantiation elements creation.	OE	3.1.3. 3.1.3. 5.1.3	To ensure that any deployment dependencies provided to the ApplicationFactory are also used for the creation of nested applications.	The ApplicationFactory component gives precedence to properties within the deploymentDependencies parameter over matching properties defined within the assemblyinstantion element when creating a nested application.
SCA74	The create operation shall deploy the ApplicationComponents as specified in the SAD.	OE	3.1.3. 3.1.3. 5.1.3	To ensure that the ApplicationComponents identified in the SAD are deployed.	The Application components identified in the SAD are deployed as specified.
SCA75	The create operation shall use each component's SPD implementation code's stacksize and priority elements, when specified, for the execute options parameters.	OE	3.1.3. 3.1.3. 5.1.3	To ensure the create operation uses the stacksize and priority values defined in the code child element of the implementation element in the SPD.	The stack size and priority values defined within the code element in the SPD are passed in to execute options parameters.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA84	The create operation shall, in order, initialize all ApplicationComponents, establish connections for those components, and finally configure ManageableApplicationComponent(s) as identified by the assemblycontroller element in the SAD.	OE	3.1.3. 3.1.3. 5.1.3	Ensures the standard sequence of operations are called for ApplicationComponents.	1. The create operation calls the initialize operation on each of an application's components. 2. The create operation establishes the application's connections. 3. The create operation calls the configure operation on the Components identified by the assemblycontroller element in the SAD.
SCA91	The create operation shall use the property values contained in the input initConfiguration parameter over the property values of the SAD's assemblycontroller element when they reference the same property.	OE	3.1.3. 3.1.3. 5.1.3	To establish the precedence of the input initConfiguration property values over matching properties in the Domain Profile of the assembly controller component.	1. The initConfiguration input parameter provided to the create operation provides values for some configuration properties of the assembly controller component. 2. The values provided by the input parameter takes precedence over the values defined for these properties in the assembly controller's profile.
SCA92	The create operation shall recognize application deployment channel preferences contained within an ADD file.	OE	3.1.3. 3.1.3. 5.1.3	To ensure the ApplicationFactoryComponent uses the channel preferences identified in the ADD file.	Do Not verify - Recommended for removal

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA93	The create operation shall recognize a deploymentDependencies property which is a CF::Properties type with an id of "DEPLOYMENT_CHANNEL" and a value that is a string sequence.	OE	3.1.3. 3.1.3. 5.1.3	To ensure the ApplicationFactoryComponent create operation accepts the "DEPLOYMENT_CHANNEL" deploymentDependencies property parameter.	The ApplicationFactoryComponent accepted the deploymentDependencies property parameter with an ID "DEPLOYMENT_CHANNEL" and a value of type string sequence.
SCA94	The create operation shall recognize channel preferences contained within a "DEPLOYMENT_CHANNEL" deploymentDependency property contained within the deploymentDependencies parameter.	OE	3.1.3. 3.1.3. 5.1.3	To ensure the ApplicationFactoryComponent uses the channel preferences identified in the "DEPLOYMENT_CHANNEL" property contained in the deploymentDependencies parameter.	Do Not verify - Recommended for removal

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA95	The create operation shall attempt to allocate an application to the PDD file channel alternatives provided within a "DEPLOYMENT_CHANNEL" property or an ADD file in a sequential manner.	OE	3.1.3. 3.1.3. 5.1.3	To provide a standard mechanism for deploying an application based on channel preferences.	1. The PDD file exists. 2. One of the following application deployments occurred: 2.1. The ApplicationFactoryComponent deployed the application on one of the deployment channels defined in the PDD and referenced by the "DEPLOYMENT_CHANNEL" property. 2.2. The ApplicationFactoryComponent deployed the application on one of the deployment channels defined in the PDD and referenced by the ADD.
SCA96	The create operation shall utilize channel preferences expressed within a "DEPLOYMENT_CHANNEL" property rather than those contained within an ADD file if both exist.	OE	3.1.3. 3.1.3. 5.1.3	To establish the precedence of the "DEPLOYMENT_CHANNEL" property values over that contained within ADD file when both exist.	1. The PDD file exists. 2. When the DEPLOYMENT_CHANNEL property exists in the input deploymentDependencies parameter and an ADD file exists, the channel preferences specified in input deploymentDependencies parameter is used.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA97	The create operation shall recognize a deployment option with a deployedname attribute value of "DEFAULT" which matches all application instance names that are not explicitly identified by a deployedname attribute value within the same descriptor file.	OE	3.1.3.3.1.3.5.1.3	To ensure the platform can use a default set of candidate channels for application deployment when a preference is not specified for a specific application instance.	The channel defined within the deploymentoption with deployedname DEFAULT is used when the application's instance name doesn't match any of the other deploymentoption's deployedname.
SCA575	The create operation shall use the affinity values contained in the input executionAffinityAssignments parameter prior to those specified by the ApplicationFactoryComponent profile's processcollocation attribute and/or coreaffinity element.	OE	3.1.3.3.1.3.5.1.3	To establish the precedence of the execution affinity value parameter over the process colocation and/or coreaffinity defined in the SAD.	When the affinity property exists in the input executionAffinityAssignments parameter and the SAD, the input executionAffinityAssignments parameter is used.
SCA102	The create operation shall return the created ApplicationManagerComponent's CF::ComponentType for the created application when the application is successfully created.	OE	3.1.3.3.1.3.5.1.4	To ensure the required information for the ApplicationManagerComponent is returned by the create operation when the application is successfully created.	The create operation returns a CF::ComponentType structure with the required information for the application that has been created.
SCA576	The create operation's returned CF::ComponentType's specializedInfo shall contain the application's deployed components as identified by COMPONENTS_ID and CF::Components type value.	OE	3.1.3.3.1.3.5.1.4	To ensure the list of deployed components for an application is returned from the create operation.	The returned CF::ComponentType's specializedInfo contains a list of deployed components as specified in the SAD.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA103	The create operation shall raise the CreateApplicationRequestError exception when the input deviceAssignments parameter contains one or more invalid application component to device assignment(s).	OE	3.1.3. 3.1.3. 5.1.5	To ensure an exception is raised when the input deviceAssignments parameter contains one or more invalid application component to device assignment(s).	1. The input deviceAssignments parameter contained one or more invalid application component to device assignment(s). 2. The create operation raised the CreateApplicationRequestError exception.
SCA104	The create operation shall raise the CreateApplicationError exception when the create request is valid but the application cannot be successfully instantiated due to internal processing error(s).	OE	3.1.3. 3.1.3. 5.1.5	To provide a standard mechanism for reporting errors when the create operation fails to instantiate an application due to internal processing error.	An internal processing error in the create operation occurs which prevents the instantiation of a new application and the CreateApplicationError exception is raised.
SCA105	The create operation shall raise the CreateApplicationError exception when the CF implementation provides enhanced deployment support via the use of a PDD file if the CF is not able to allocate the application to any of the provided channel alternatives.	OE	3.1.3. 3.1.3. 5.1.5	To provide a standard mechanism for reporting errors when the create operation cannot instantiate an application on any of the provided channel alternatives.	1. The CF implementation provides the deployment support via the use of a PDD file. 2. An application cannot be deployed to any of its specified channel alternatives. 3. The CreateApplication Error is raised.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA106	The create operation shall raise the CreateApplicationError exception when the CF implementation provides enhanced deployment support via the use of a PDD file and a domainfinder element "servicetype" connection to a ServiceComponent whose service type is provided by a service contained within a channel element servicelist cannot be established to a service identified within that list.	OE	3.1.3. 3.1.3. 5.1.5	To ensure a CreateApplicationError exception is raised when a connection to a service cannot be established to a service identified in the PDD.	1. A PDD File Exists. 2. domainfinder element "servicetype" connection to a ServiceComponent is to be established. 3. The service cannot be identified within the servicelist of a channel in the PDD. 4. CreateApplicationError exception is raised.
SCA107	The create operation shall raise the InvalidInitConfiguration exception when the input initConfiguration parameter contains properties that are unknown by a SAD's assemblycontroller element.	OE	3.1.3. 3.1.3. 5.1.5	To ensure an exception is raised when the input initConfiguration parameter contains properties that are unknown by an assembly controller of an application.	The create operation raises the InvalidInitConfiguration exception when the input initConfiguration parameter contains properties that are unknown by a SAD's assemblycontroller element.
SCA108	The InvalidInitConfiguration invalidProperties parameter shall identify the invalid properties.	OE	3.1.3. 3.1.3. 5.1.5	Ensure InvalidInitConfiguration exception contains the invalid properties when raised.	Verified by conformance to SCA107.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA570	The create operation shall raise the CreateApplicationError exception when an ApplicationManagerComponent already exists in the system with a CF::ComponentType identifier attribute value equal to that of the input name parameter.	OE	3.1.3. 3.1.3. 5.1.5	To ensure an exception is raised when an attempt is made to create an ApplicationManager with a name that already exists in the system.	The create operation raises the CreateApplicationError exception when an ApplicationManagerComponent already exists in the system with a CF::ComponentType identifier attribute value equal to that of the input name parameter.
SCA109	The readonly managers attribute shall return a list of DeviceManagerComponents that have registered with the DomainManagerComponent.	OE	3.1.3. 3.1.4. 4.1	To ensure the DomainManagerComponent provides a reference to a list of its registered DeviceManagerComponents.	The list of CF::ComponentTypes provided by the DomainManager's managers attribute contains the DeviceManagerComponents that have registered with the DomainManagerComponent.
SCA110	The readonly applications attribute shall return the list of ApplicationManagerComponents that have been instantiated.	OE	3.1.3. 3.1.4. 4.2	To ensure the DomainManager provides a list of the instantiated ApplicationManagerComponents.	The applications attribute provides a list of CF::ComponentTypes with appropriate information for each ApplicationManagerComponent that has been instantiated.
SCA435	The readonly applicationFactories attribute shall return a list with one ApplicationFactoryComponent per AssemblyComponent (SAD file and associated files) successfully installed (i.e. no exception raised).	OE	3.1.3. 3.1.4. 4.3	To ensure the DomainManager provides a list of the installed ApplicationFactoryComponents.	The applicationFactories attribute provides a list of CF::ComponentTypes with appropriate information for each AssemblyComponent that has been successfully installed.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA111	The readonly fileMgr attribute shall return the DomainManagerComponent's FileManagerComponent.	OE	3.1.3. 3.1.4. 4.4	To ensure the DomainManager provides a reference to its FileManagerComponent.	The fileMgr attribute contains a reference to the DomainManager's FileManager.
SCA112	The readonly domainManagerProfile attribute shall return the filename of the DomainManagerComponent's DMD or the DMD itself.	OE	3.1.3. 3.1.4. 4.5	To ensure the DomainManager provides access to the DomainManagerComponent's DMD content.	The domainManagerProfile attribute contains the path including filename of the domain's DMD or the content of the DMD itself.
SCA113	The installApplication operation shall verify the existence of the AssemblyComponent's SAD file and all files upon which the SAD depends, within the DomainManagerComponent's file manager.	OE	3.1.3. 3.1.5. 5.1.3	To ensure the DomainManagerComponent has access to all domain profile files for the application being installed.	1. All files required by an application are in the DomainManager's File Manager. 2. The DomainManager's installApplication operation is invoked and doesn't throw an exception.
SCA114	The installApplication operation shall write an ADMINISTRATIVE_EVENT log record to a DomainManagerComponent's log, upon successful application installation.	OE	3.1.3. 3.1.5. 5.1.3	To ensure a successful execution of the DomainInstallation::installApplication operation is logged in the DomainManagerComponent's log as an ADMINISTRATIVE_EVENT.	1. Log Service is supported. 2. DomainManagerComponent's log contains an ADMINISTRATIVE_EVENT log record for each successfully installed application.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA115	The installApplication operation shall, upon unsuccessful application installation, write a FAILURE_ALARM log record to a DomainManagerComponent's log.	OE	3.1.3. 3.1.5. 5.1.3	To ensure an unsuccessful execution of the DomainInstallation::installApplication operation is logged in the DomainManagerComponent's log as a FAILURE_ALARM.	1. Log Service is supported. 2. DomainManagerComponent's log contains a FAILURE_ALARM when an installApplication operation is unsuccessful.
SCA116	The installApplication operation shall send a ComponentChangeEventType event to the Outgoing Domain Management event channel, upon successful installation of an application.	OE	3.1.3. 3.1.5. 5.1.3	To ensure the DomainInstallation::installApplication operation sends a ComponentChangeEventType event to the Outgoing Domain Management event channel when an application is successfully installed.	The DomainInstallation::installApplication operation sends a ComponentChangeEventType event to the Outgoing Domain Management event channel upon successful installation of an application. The ComponentChangeEventType contains following: 1. The producerId is the identifier attribute of the DomainManagerComponent. 2. The ComponentChange is ADDED. 3. The domainComponent is the installed ApplicationFactoryComponent's CF::ComponentType.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA571	The installApplication operation shall return the installed ApplicationFactoryComponent's CF::ComponentType.	OE	3.1.3. 3.1.5. 5.1.4	To ensure the required information for the ApplicationFactoryComponent is returned by the installApplication operation when the application is successfully installed.	The installApplication operation returns a CF::ComponentType structure with the required information for the application that has been installed.
SCA117	The installApplication operation shall raise the ApplicationInstallationError exception when the installation of the application file(s) was not successfully completed.	OE	3.1.3. 3.1.5. 5.1.5	To ensure an exception is raised when the installation of the application file(s) is not successfully completed.	The installApplication operation raises the ApplicationInstallationError exception when the installation of the application file(s) was not successfully completed.
SCA118	The installApplication operation shall raise the CF::InvalidFileName exception when the input SAD file or any of the SAD's referenced filenames do not exist in the file system identified by the absolute path of the input profileFileName parameter.	OE	3.1.3. 3.1.5. 5.1.5	To ensure an exception is raised when the input SAD file or any of the SAD's referenced filenames do not exist in the file system.	The installApplication operation raised the CF::InvalidFileName exception when the input SAD file or any of the SAD's referenced filenames did not exist in the file system identified by the absolute path of the input profileFileName parameter.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA119	The installApplication operation shall log a FAILURE_ALARM log record to a DomainManagerComponent's Log with a message consisting of "installApplication::invalid file is xxx", where "xxx" is the input or referenced filename, when the CF InvalidFileName exception occurs.	OE	3.1.3. 3.1.5. 5.1.5	To ensure an unsuccessful execution of the DomainInstallation::installApplication operation is logged in the DomainManagerComponent's log as a FAILURE_ALARM and provides the invalid file name.	1. Log Service is supported. 2. The CF InvalidFileName exception occurs. 3. DomainManagerComponent's log contains a FAILURE_ALARM log record with a message consisting of "installApplication::invalid file is xxx", where "xxx" is the input or referenced filename.
SCA120	The installApplication operation shall raise the CF::InvalidProfile exception when any referenced property definition is missing.	OE	3.1.3. 3.1.5. 5.1.5	To ensure an exception is raised when a XML element that is referenced by a refid attribute is missing when installing an application.	The installApplication operation raised the CF::InvalidProfile exception when a XML element that is referenced by a refid attribute is missing.
SCA121	The installApplication operation shall write a FAILURE_ALARM log record to a DomainManagerComponent's log when the CF::InvalidProfile exception is raised.	OE	3.1.3. 3.1.5. 5.1.5	To ensure the DomainInstallation::installApplication operation logs a FAILURE_ALARM when the CF::InvalidProfile exception is raised.	1. Log Service is supported. 2. The InvalidProfile exception is raised by the DomainInstallation::installApplication operation and DomainManagerComponent's log contains a FAILURE_AL ARM.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA122	The installApplication operation shall raise the ApplicationAlreadyInstalled exception when the softwareassembly element name attribute of the referenced application is the same as a previously registered application.	OE	3.1.3. 3.1.5. 5.1.5	To ensure an exception is raised when the name of the application to be installed is the same as a previously registered application.	The installApplication operation raised the ApplicationAlreadyInstalled exception when the softwareassembly element name attribute of the referenced application is the same as a previously registered application name.
SCA123	The uninstallApplication operation shall, upon successful uninstall of an application, write an ADMINISTRATIVE_EVENT log record to a DomainManagerComponent's log.	OE	3.1.3. 3.1.5. 5.2.3	To ensure a successful execution of the DomainInstallation::uninstallApplication operation is logged in the DomainManagerComponent's log as an ADMINISTRATIVE_EVENT.	1. Log Service is supported. 2. DomainManagerComponent's log contains an ADMINISTRATIVE_EVENT log record when a uninstallApplication operation is successful.
SCA124	The uninstallApplication operation shall, upon unsuccessful uninstall of an application, write a FAILURE_ALARM log record to a DomainManagerComponent's log.	OE	3.1.3. 3.1.5. 5.2.3	To ensure the DomainInstallation::uninstallApplication operation logs a FAILURE_ALARM when the operation fails.	1. Log Service is supported. 2. DomainManagerComponent's log contains a FAILURE_ALARM log record when an uninstallApplication operation is unsuccessful.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA125	The uninstallApplication operation shall send a ComponentChangeEventType event to the Outgoing Domain Management event channel, upon the successful uninstallation of an application.	OE	3.1.3. 3.1.5. 5.2.3	To ensure the DomainInstallation::uninstallApplication operation sends a ComponentChangeEventType event to the Outgoing Domain Management event channel when an application is successfully uninstalled.	The DomainInstallation::uninstallApplication operation sent a ComponentChangeEventType event to the Outgoing Domain Management event channel upon successful uninstall of an application. The ComponentChangeEventType contains following: 1. The producerId is the identifier attribute of the DomainManagerComponent. 2. The ComponentChange is REMOVED. 3. The domainComponent is the uninstalled ApplicationFactoryComponent's CF::ComponentType.
SCA436	The uninstallApplication operation shall make the ApplicationFactoryComponent unavailable from the DomainManagerComponent (i.e. its services no longer provided for the application).	OE	3.1.3. 3.1.5. 5.2.3	To ensure an instance of the application cannot be created after the application has been uninstalled.	The DomainManagerComponent does not provide a reference to the uninstalled ApplicationFactoryComponent.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA126	The uninstallApplication operation shall raise the InvalidIdentifier exception when the identifier parameter is invalid.	OE	3.1.3. 3.1.5. 5.2.5	To ensure the DomainInstallation::uninstallApplication operation raises an exception when the identifier parameter is invalid.	The DomainInstallation::uninstallApplication operation raised an InvalidIdentifier exception when the identifier parameter is invalid.
SCA127	The uninstallApplication operation shall raise the ApplicationUninstallationError exception when an internal error causes an unsuccessful uninstallation of the application.	OE	3.1.3. 3.1.5. 5.2.5	To ensure the DomainInstallation::uninstallApplication operation raises an exception when the operation is unsuccessful due to an internal error.	The DomainInstallation::uninstallApplication operation raised an ApplicationUninstallationError exception when the operation is unsuccessful due to an internal error.
SCA131	The registerComponent operation shall register the component indicated by the input registeringComponent parameter, if it does not already exist.	OE	3.1.3. 3.1.6. 5.1.3	To ensure all components specified by the input registeringComponent parameter are registered and only once	1. The component identified by the input parameter is added to the manager list (e.g. ApplicationFactoryComponent, DomainManagerComponent or DeviceManagerComponent) of registered components. 2. The component identified by the input parameter is not added to the manager list of deployed components if it is already registered.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA132	The registerComponent operation shall raise the CF::InvalidObjectReference when the input registeringComponent contains a nil componentObject object reference.	OE	3.1.3. 3.1.6. 5.1.5	To ensure the registerComponent operation raises an exception when the input registeringComponent contains a nil componentObject object reference.	The registerComponent operation raised a CF::InvalidObjectReference exception when the input registeringComponent contained a nil componentObject object reference.
SCA133	The registerComponent operation shall raise the RegisterError exception when registration is unsuccessful.	OE	3.1.3. 3.1.6. 5.1.5	To ensure the registerComponent operation raises an exception when registration is unsuccessful.	The registerComponent operation raised the RegisterError exception when registration was unsuccessful.
SCA134	The unregisterComponent operation shall unregister a registered component entry specified by the input identifier parameter.	OE	3.1.3. 3.1.7. 5.1.3	Unregister a specified registered component.	The component identified by the input parameter is unregistered and removed from the manager list of deployed components.
SCA135	The unregisterComponent operation shall raise the UnregisterError exception when unregistration is unsuccessful.	OE	3.1.3. 3.1.7. 5.1.5	To ensure the unregisterComponent operation raises an exception when unregistration is unsuccessful.	The unregisterComponent operation raised the UnregisterError exception when unregistration was unsuccessful.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA136	The registerWithEventChannel operation shall connect, with a connection named by the input registeringId parameter, the object contained within the input registeringObject parameter to an event channel specified by the input eventChannelName parameter.	OE	3.1.3. 3.1.8. 5.1.3	To ensure the registerWithEventChannel operation connects a registering component to the specified domain's event channel.	The registeringObject is connected to the event channel as identified in the input eventChannelName parameter and the connection name is the same as the input registeringId parameter.
SCA137	The registerWithEventChannel operation shall raise the CF::InvalidObjectReference exception when the input registeringObject parameter contains an invalid reference to a CosEventComm::PushConsumer interface.	OE	3.1.3. 3.1.8. 5.1.5	To ensure the registerWithEventChannel operation raises an exception when the input registeringObject parameter contains an invalid reference to a CosEventComm::PushConsumer interface.	The registerWithEventChannel operation raised the CF::InvalidObjectReference exception when the input registeringObject parameter contained an invalid reference to a CosEventComm::PushConsumer interface.
SCA138	The registerWithEventChannel operation shall raise the InvalidEventChannelName exception when the input eventChannelName parameter contains an invalid event channel name.	OE	3.1.3. 3.1.8. 5.1.5	To ensure the registerWithEventChannel operation raises an exception when the input eventChannelName parameter contains an invalid event channel name.	The registerWithEventChannel operation raised an InvalidEventChannelName exception when the input eventChannelName parameter contained an invalid event channel name.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA139	The registerWithEventChannel operation shall raise AlreadyConnected exception when the object contained within the input registeringObject parameter already contains a connection identified by the input registeringId parameter.	OE	3.1.3. 3.1.8. 5.1.5	To ensure the registerWithEventChannel operation raises an exception when the connection identified by the registeringId and eventChannelName parameters already exists.	The registerWithEventChannel operation raised an AlreadyConnected exception when the connection identified by the registeringId and eventChannelName parameters already existed.
SCA140	The unregisterFromEventChannel operation shall disconnect a registered component from the event channel as identified by the input parameters.	OE	3.1.3. 3.1.8. 5.2.3	To ensure the unregisterFromEventChannel operation disconnects an unregistering component from the specified domain's event channel.	The component as identified by the input unregisteringId parameter is disconnected from the event channel identified by the input eventChannelName parameter.
SCA141	The unregisterFromEventChannel operation shall raise the InvalidEventChannelName exception when the input eventChannelName parameter can't be located as a named event channel within the domain.	OE	3.1.3. 3.1.8. 5.2.5	To ensure the unregisterFromEventChannel operation raises an exception when the input eventChannelName parameter does not identify an existing event channel within the domain.	The unregisterFromEventChannel operation raised an InvalidEventChannelName exception when the input eventChannelName parameter was not identified as an event channel within the domain.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA142	The unregisterFromEventChannel operation shall raise the NotConnected exception when the input unregisteringId parameter is not associated with the input eventChannelName parameter.	OE	3.1.3. 3.1.8. 5.2.5	To ensure the unregisterFromEventChannel operation raises an exception when the input unregisteringId parameter is not registered to the event channel identified by the input eventChannelName parameter.	The unregisterFromEventChannel operation raised a NotConnected exception when the input unregisteringId parameter was not registered to the event channel identified by the input eventChannelName parameter.
SCA151	The shutdown operation shall unregister the manager from the domain.	OE	3.1.3. 3.1.9. 5.1.3	To ensure the shutting down manager is unregistered from the domain.	The manager that was shut down is not in the DomainManger managers attribute.
SCA152	The shutdown operation shall perform a releaseObject on all of the manager's registered components that support the LifeCycle interface.	OE	3.1.3. 3.1.9. 5.1.3	To ensure releaseObject is called on all of the manager's registered components that support the LifeCycle interface.	releaseObject operation is called for each registered component that supports the LifeCycle interface in the collection of deployed components.
SCA153	The shutdown operation shall terminate the execution of each component created as specified in the manager's profile after it has unregistered from the manager.	OE	3.1.3. 3.1.9. 5.1.3	To ensure threads / processes are terminated for each component specified in the managers' profile.	Processes/threads associated with the unregistered components no longer exist.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA437	The shutdown operation shall cause the manager to be unavailable (i.e. released from the operating environment and its process terminated on the OS), when all of the manager's registered components are unregistered and all created components are terminated.	OE	3.1.3. 3.1.9. 5.1.3	To ensure the manager is unavailable after shutdown.	1. All of the manager's components are unregistered. 2. The DeviceManager is dereferenced. 3. The device manager's process is terminated.
SCA158	An ApplicationManagerComponent shall delegate the runTest, start, stop, configure, and query operations to the ApplicationControllerComponent(s) as identified by the AssemblyComponent's SAD assemblycontroller element (application controller).	OE	3.1.3. 3.2.1. 3	To ensure the ApplicationManagerComponent delegates runTest, start, stop, configure, and query operations to the ApplicationControllerComponent(s) to allow Application specific behavior.	1. An external entity (e.g. HMI, nonCore Framework component) calls the ApplicationManagerComponent runTest, start, stop, configure, and query operations. 2. ApplicationManagerComponent delegates the corresponding runTest, start, stop, configure, and query operations to the ApplicationControllerComponent(s).

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA159	The ApplicationManagerComponent shall propagate exceptions raised by the AssemblyComponent's ApplicationControllerComponent(s).	OE	3.1.3.3.2.1.3	To ensure the ApplicationManagerComponent propagates exceptions raised by an application.	1. An external entity (e.g. HMI, nonCore Framework component) calls the ApplicationManagerComponent runTest, start, stop, configure, and query operations. 2. ApplicationManagerComponent propagates exceptions raised by runTest, start, stop, configure, and query operations of the ApplicationControllerComponent(s).
SCA160	The ApplicationManagerComponent shall not delegate the initialize operation to its ApplicationComponentFactoryComponent(s), ManageableApplicationComponent(s) or ApplicationControllerComponent(s).	OE	3.1.3.3.2.1.3	Ensures the ApplicationManagerComponent does not cause reinitialization of the application.	The ApplicationManagerComponent's initialize operation does not call the initialize operation of the ApplicationComponentFactoryComponent(s), ManageableApplicationComponent(s) or ApplicationControllerComponent(s).

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA161	The ApplicationManagerComponent shall delegate the runTest operation to all component(s) as identified by the AssemblyComponent's SAD assemblycontroller element (application controller) which have matching test IDs.	OE	3.1.3.3.2.1.3	To ensure the ApplicationManagerComponent delegates the runTest operation to ApplicationControllerComponent(s) which have the matching test IDs.	The ApplicationManagerComponent invoked the runTest operation for each ApplicationControllerComponent with a matching test ID.
SCA162	The ApplicationManagerComponent shall delegate configure and query operations to all ManageableApplicationComponent(s) as identified by the AssemblyComponent's SAD assemblycontroller element (application controller), which have matching property IDs.	OE	3.1.3.3.2.1.3	Configure and query operations are routed to the ApplicationControllerComponents for properties that have matching configure / query property IDs of those identified in the components Domain Profile.	The ApplicationManagerComponent calls configure/query operation on the appropriate ApplicationControllerComponent(s).
SCA163	The ApplicationManagerComponent shall raise the configure operation's InvalidConfiguration exception when the input configProperties parameter contains unknown properties.	OE	3.1.3.3.2.1.3	To ensure the ApplicationManagerComponent configure operation raises an exception when the exception when the input configProperties parameter contains unknown properties.	The ApplicationManagerComponent configure operation raised an InvalidConfiguration exception when the input configProperties parameter contained unknown properties.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA543	The ApplicationManagerComponent shall raise the query operation's UnknownProperties exception when the input configProperties parameter contains unknown properties.	OE	3.1.3.3.2.1.3	To ensure the ApplicationManagerComponent query operation raises an exception when the input configProperties parameter contains unknown properties.	The ApplicationManagerComponent query operation raised an UnknownProperties exception when the input configProperties parameter contained unknown properties.
SCA164	An ApplicationManagerComponent shall realize the ApplicationManager interface.	OE	3.1.3.3.2.1.4	To ensure an ApplicationManagerComponent provides standard mechanisms to manage an instantiated application.	The ApplicationManagerComponent inherits the ApplicationManager interface.
SCA165	An ApplicationManagerComponent shall fulfill the BaseComponent requirements.	OE	3.1.3.3.2.1.4	To ensure an ApplicationManagerComponent satisfies the requirements of a BaseComponent.	Will be satisfied by the verification of BaseComponent requirements.
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 DeviceComponent, for those ManageableApplicationComponent properties whose kindtype is "allocation" and whose action element is not "external".	OE	3.1.3.3.2.2.3	To ensure a standard mechanism to identify the DeviceComponents which provide the necessary resources on which each ManageableApplicationComponent can be deployed.	After the create operation finished, the ApplicationComponents have been deployed on DeviceComponents for which the properties whose kindtype is "allocation" and whose action element is not "external" matches the allocation properties of the DeviceComponents.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA71	The create operation shall allocate capacities to candidate DeviceComponents of the ApplicationComponent properties whose kindtype is "allocation" and whose action element is "external".	OE	3.1.3.3.2.2.3	To ensure capacities are allocated on the DeviceComponents that are candidates for the deployment of the ApplicationComponents.	1. The ApplicationComponent has properties with a kindtype of "allocation" and the action element has the value "external". 2. Device capacities are allocated when the device has available capacity.
SCA72	The create operation shall deallocate any capacity allocations on DeviceComponents that do not satisfy the ApplicationComponent's allocation requirements or that are not utilized due to an unsuccessful application creation.	OE	3.1.3.3.2.2.3	To ensure the ApplicationFactory releases allocated capacities when an application is not created successfully.	1. The application is not created successfully. 2. Device capacity is deallocated for those DeviceComponents where allocation was successful.
SCA73	The create operation shall load application modules onto DeviceComponents that have been granted successful capacity allocations and satisfy the ApplicationComponent's allocation requirements.	OE	3.1.3.3.2.2.3	To ensure ApplicationComponents are loaded on the LoadableDeviceComponents where the capacities have been granted and that satisfy the allocation requirements.	1. LoadableDeviceComponents satisfies the ApplicationComponent's allocation requirements. 2. Capacity allocations have been granted on the LoadableDeviceComponents. 3. The ApplicationComponents are loaded on the LoadableDeviceComponents.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA76	When the create operation deploys an ApplicationComponent via an ExecutableDeviceComponent, it shall include a Component Identifier, as defined in this section, in the parameters parameter of the ExecutableInterface::execute operation.	OE	3.1.3.3.2.2.3	To ensure a created ApplicationComponent has a unique Component Identifier.	The create operation passes the Component Identifier in the execute parameters in the call to ExecutableInterface::execute operation.
SCA77	When the create operation deploys an ApplicationComponent via an ApplicationComponentFactoryComponent, it shall provide the Component Identifier parameter as defined in this section.	OE	3.1.3.3.2.2.3	To ensure a created ApplicationComponent has a unique Component Identifier.	The create operation passes the Component Identifier as a parameter in the call to ComponentFactory::createComponent operation.
SCA81	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 when an ApplicationComponent is deployed via an ExecutableDeviceComponent.	OE	3.1.3.3.2.2.3	Provide standard mechanism for ApplicationComponent deployment using an ExecutableDeviceComponent.	The CF::ApplicationFactory::create calls CF::ExecutableInterface::execute supplying the execparam parameters of the componentinstantiation specified in the domain profile.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA83	The create operation, when creating an ApplicationComponent from an ApplicationComponentFactoryComponent, shall pass the componentinstantiation componentfactoryref element properties whose kindtype element is "factoryparam" as the qualifiers parameter to the referenced ApplicationComponentFactoryComponent's createComponent operation.	OE	3.1.3.3.2.2.3	Provide standard mechanism for creating ApplicationComponents using a ApplicationComponentFactoryComponent.	The CF::ApplicationFactory::create operation calls ApplicationComponentFactoryComponent's createComponent operation with the "factoryparam" parameters of the componentinstantiation specified in the domain profile.
SCA85	The create operation shall establish connections for an AssemblyComponent which are specified in the SAD connections element.	OE	3.1.3.3.2.2.3	To ensure that the ApplicationFactory establishes all of an application's connections.	All connections specified in the SAD's connections element are established.
SCA86	The create operation shall use the SAD connectinterface element id attribute as part of the unique identifier for a specific connection when provided.	OE	3.1.3.3.2.2.3	To ensure the unique identifier of a connection is constructed in a standard manner when the connectinterface element has an id attribute specified.	For each connectinterface element id in the SAD where an id attribute is specified, a unique identifier is associated with each connection instance which contains the id attribute.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA87	The create operation shall create a unique identifier and use it to designate a connection when no SAD connectinterface element id attribute is specified.	OE	3.1.3.3.2.2.3	To ensure a unique identifier of a connection is constructed when no SAD connectinterface element id attribute is specified.	For each connectinterface element in the SAD where an id attribute is not specified, a unique identifier has been created and associated with each connection instance.
SCA88	For connections to an event channel, the create operation shall connect a CosEventComm::PushConsumer or CosEventComm::PushSupplier object to the event channel as specified in the SAD's domainfinder element.	OE	3.1.3.3.2.2.3	To ensure the create operation connects CosEventComm::Push Consumer or CosEventComm::Push Supplier to a specified event channel.	All created components' event producer or event consumer ports are connected to the event channel(s) as specified in the SAD's domainfinder element.
SCA89	The create operation shall create the specified event channel if the event channel does not exist.	OE	3.1.3.3.2.2.3	To ensure an identified event channel is created if it does not already exist.	1a. A specified event channel does not exist prior to application creation. 2a. After application creation, a specified event channel exists. 1b. A specified event channel exists prior to application creation. 2b. After application creation, a specified event channel is not duplicated.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA90	The create operation shall configure the ManageableApplicationComponent(s) indicated by the assemblycontroller element in the SAD that have properties with a kindtype of "configure" and a mode of "readwrite" or "writeonly" along with the union of properties contained in the input initConfiguration parameter of the create operation.	OE	3.1.3.3.2.2.3	To ensure each application's ManageableApplicationComponent(s) indicated by assemblycontroller element is configured with the union of the input initConfiguration parameter and the configure properties as specified in the Domain Profile.	The create operation calls the CF::PropertySet::configure for each ApplicationControllerComponent with union of properties contained in the input initConfiguration parameter and all the component's configurable properties, considering the overriding precedence specified in SCA91.
SCA98	For domainfinder element "servicetype" connections to a ServiceComponent whose service type is provided by a service contained within a channel element servicelist, the create operation shall only attempt to establish connections to services within the list.	OE	3.1.3.3.2.2.3	To ensure connections to services are restricted to a channel when channels are defined.	1. Identify the services a within channel element servicelist as defined in the PDD. 2. Identify the "servicetype" application connections for the ServiceComponent as defined in the SAD. 3. For application connections identified in step 2 all connections are established to the ServiceComponents identified in step 1.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA99	The create operation shall, upon successful application creation, write an ADMINISTRATIVE_EVENT log record.	OE	3.1.3.3.2.2.3	To ensure a successful ApplicationFactory::create operation is logged as an ADMINISTRATIVE_EVENT.	1. Log Service is supported. 2. DomainManagerComponent's log contains an ADMINISTRATIVE_EVENT log record when an ApplicationFactory::create operation is successful.
SCA100	The create operation shall, upon unsuccessful application creation, write a FAILURE_ALARM log record.	OE	3.1.3.3.2.2.3	To ensure an unsuccessful ApplicationFactory::create operation is logged as a FAILURE_ALARM.	1. Log Service is supported. 2. DomainManagerComponent's log contains a FAILURE_ALARM log record when an ApplicationFactory::create operation is unsuccessful.
SCA101	The create operation shall send a ComponentChangeEventType event to the Outgoing Domain Management event channel upon successful creation of an application.	OE	3.1.3.3.2.2.3	To ensure the ApplicationFactory sends a ComponentChangeEventType event to the Outgoing Domain Management event channel when an application is successfully created.	The ApplicationFactory::create operation sent a ComponentChangeEventType event to the Outgoing Domain Management event channel upon successful creation of an application. The ComponentChangeEventType contains the following: 1. The producerId is the identifier attribute of the ApplicationFactoryComponent. 2. The ComponentChange is ADDED. 3. The domainComponent is the returned ApplicationManagerComponent's CF::ComponentType.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA542	When the create operation deploys an ApplicationComponent via an ExecutableDeviceComponent, it shall include a ComponentRegistry IOR, as defined in this section, in the parameters parameter of the ExecutableInterface::execute operation when the SAD componentinstantiation stringifiedobjectref element is null value.	OE	3.1.3.3.2.2.3	To provide a standard mechanism for obtaining the reference of an ApplicationComponent.	1. The SAD componentinstantiation stringifiedobjectref element is null value. 2. The create operation passes the ComponentRegistry IOR in the execute parameter with the identifier COMPONENT_REGISTRY_IOR in the call to ExecutableInterface::execute operation.
SCA174	An ApplicationFactoryComponent shall realize the ApplicationFactory interface.	OE	3.1.3.3.2.2.4	To ensure an ApplicationFactoryComponent provides standard mechanisms to create applications.	The ApplicationFactoryComponent inherits the ApplicationFactory interface.
SCA144	The registerComponent operation shall register all of the components identified within the registeringComponent's specializedInfo field when the registeringComponent's type field is DEVICE_MANAGER_COMPONENT.	OE	3.1.3.3.2.3.3	To ensure a Domain Manager registers all components that have registered with a device manager when a device manager registers to the Domain Manager.	All components specified in the registering ComponentType's specializedInfo field are registered with the DomainManagerComponent when the ComponentType field is DEVICE_MANAGER_COMPONENT.
SCA149	The unregisterComponent operation shall unregister all of the associated (i.e. registered) components from the component being unregistered when its type field is a DEVICE_MANAGER_COMPONENT.	OE	3.1.3.3.2.3.3	Unregister all components that registered with the device manager.	Each registered component associated with the unregistering device manager component is removed from the collection of deployed components of the DomainManagerComponent.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA177	The DomainManagerComponent identifier shall be identical to the domainmanagerconfiguration element id attribute of the DMD file.	OE	3.1.3.3.2.3.3	To specify the value of the DomainManagerComponent identifier.	DomainManagerComponent identifier value matches the DMD file domainmanagerconfiguration element id attribute.
SCA178	A DomainManagerComponent shall define its utilized ServiceComponents in the DMD.	OE	3.1.3.3.2.3.3	To ensure the DMD contains the ServiceComponent definitions to be used by the DomainManagerComponent.	The DMD contains the service instances that are utilized by the DomainManagerComponent.
SCA179	A DomainManagerComponent shall write an ADMINISTRATIVE_EVENT log record to a DomainManagerComponent's log, when the managers attribute is obtained by a client.	OE	3.1.3.3.2.3.3	To ensure each access of a DomainManagerComponent's managers attribute is logged as an ADMINISTRATIVE_EVENT in the DomainManagerComponent's log.	1. Log Service is supported. 2. DomainManagerComponent's log contained an ADMINISTRATIVE_EVENT for each time the DomainManagerComponent's managers attribute was obtained by a client.
SCA180	A DomainManagerComponent shall write an ADMINISTRATIVE_EVENT log record to a DomainManagerComponent's log, when the applications attribute is obtained by a client.	OE	3.1.3.3.2.3.3	To ensure each access of a DomainManagerComponent's applications attribute is logged as an ADMINISTRATIVE_EVENT in the DomainManagerComponent's log.	1. Log Service is supported. 2. DomainManagerComponent's log contained an ADMINISTRATIVE_EVENT for each time the DomainManagerComponent's applications attribute was obtained by a client.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA181	A DomainManagerComponent shall write an ADMINISTRATIVE_EVENT log record to a DomainManagerComponent's log, when the applicationFactories attribute is obtained by a client.	OE	3.1.3.3.2.3.3	To ensure each access of a DomainManagerComponent's applicationFactories attribute is logged as an ADMINISTRATIVE_EVENT in the DomainManagerComponent's log.	1. Log Service is supported. 2. DomainManagerComponent's log contained an ADMINISTRATIVE_EVENT for each time the DomainManagerComponent's applicationFactories attribute was obtained by a client.
SCA182	A DomainManagerComponent shall write an ADMINISTRATIVE_EVENT log record to a DomainManagerComponent's log, when the fileMgr attribute is obtained by a client.	OE	3.1.3.3.2.3.3	To ensure each access of a DomainManagerComponent's fileMgr attribute is logged as an ADMINISTRATIVE_EVENT in the DomainManagerComponent's log.	1. Log Service is supported. 2. DomainManagerComponent's log contained an ADMINISTRATIVE_EVENT for each time the DomainManagerComponent's fileMgr attribute was obtained by a client.
SCA184	A DomainManagerComponent shall create its own FileManagerComponent that consists of all registered DeviceManagerComponent's FileSystemComponents.	OE	3.1.3.3.2.3.3	To ensure the DomainManager provides a FileManagerComponent with access to all FileSystems.	To ensure the DomainManager provides a FileManagerComponent with access to all FileSystems.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA185	Upon system startup, a DomainManagerComponent shall restore ApplicationFactoryComponents for AssemblyComponents that were previously installed by the DomainManager::installApplication operation.	OE	3.1.3.3.2.3.3	To ensure that the ApplicationFactories for the Applications that were previously installed are restored upon system startup.	Verified by SCA186
SCA186	A DomainManagerComponent shall add the restored application factories to the DomainManager interface applicationFactories attribute.	OE	3.1.3.3.2.3.3	To ensure that the references for the restored ApplicationFactories can be obtained.	Upon system startup, the DomainManager's applicationFactories attribute contains the restored ApplicationFactoryComponents for all previously installed applications.
SCA187	A DomainManagerComponent shall create the Incoming Domain Management and Outgoing Domain Management event channels.	OE	3.1.3.3.2.3.3	To ensure the DomainManagerComponent creates the Incoming Domain Management and Outgoing Domain Management event channels.	Incoming Domain Management and Outgoing Domain Management event channels exist.
SCA189	The registerComponent operation shall write an ADMINISTRATIVE_EVENT log record to a DomainManagerComponent log upon successful component registration.	OE	3.1.3.3.2.3.3	To ensure a successful registerComponent operation is logged as an ADMINISTRATIVE_EVENT in the DomainManagerComponent's log.	1. Log Service is supported. 2. DomainManagerComponent's log contains an ADMINISTRATIVE_EVENT for each registerComponent operation that was successful.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA191	The registerComponent operation shall write a FAILURE_ALARM log record to a DomainManagerComponent log upon unsuccessful component registration.	OE	3.1.3.3.2.3.3	To ensure an unsuccessful registerComponent operation is logged as an FAILURE_ALARM in the DomainManagerComponent's log.	1. Log Service is supported. 2. DomainManagerComponent's log contains a FAILURE_ALARM for each registerComponent operation that was unsuccessful.
SCA193	The registerComponent operation shall send a ComponentChangeEventType event to the Outgoing Domain Management event channel, upon successful registration of a component.	OE	3.1.3.3.2.3.3	To ensure the DomainManagerComponent sends a ComponentChangeEventType event to the Outgoing Domain Management event channel upon successful registration of a component.	The DomainManagerComponent's ComponentRegistry::registerComponent operation sent a ComponentChangeEventType event to the Outgoing Domain Management event channel upon successful registration of a component. The ComponentChangeEventType contains the following: 1. The producerId is the identifier attribute of the DomainManagerComponent. 2. The ComponentChange is ADDED. 3. The domainComponent is the input registering component CF::ComponentType.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA194	The registerComponent operation shall establish any pending connections from the registeringComponent.	OE	3.1.3.3.2.3.3	Ensure all pending port connections to / from the registering component that can be established using previously registered components are established.	1. The DomainManagerComponent has pending connections for a component to be registered. 2. The DomainManagerComponent established all the pending connections that can be satisfied by the registering component.
SCA195	The unregisterComponent operation shall, upon the successful unregistration of a component, write an ADMINISTRATIVE_EVENT log record to a DomainManagerComponent's log.	OE	3.1.3.3.2.3.3	To ensure a successful unregisterComponent operation is logged as an ADMINISTRATIVE_EVENT in the DomainManagerComponent's log.	1. Log Service is supported. 2. DomainManagerComponent's log contains an ADMINISTRATIVE_EVENT for each unregisterComponent operation that was successful.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA196	The unregisterComponent operation shall send a ComponentChangeEventType event to the Outgoing Domain Management event channel, upon successful unregistration of a component.	OE	3.1.3.3.2.3.3	To ensure the DomainManagerComponent sends a ComponentChangeEventType event to the Outgoing Domain Management event channel upon successful unregistration of a component.	The DomainManagerComponent's FullComponentRegistry::unregisterComponent operation sent a ComponentChangeEventType event to the Outgoing Domain Management event channel upon successful unregistration of a component. The ComponentChangeEventType contains the following: 1. The producerId is the identifier attribute of the DomainManagerComponent. 2. The ComponentChange is REMOVED. 3. The domainComponent is the unregistered component CF::ComponentType.
SCA197	The unregisterComponent operation shall, upon unsuccessful unregistration of a component, write a FAILURE_ALARM log record to a DomainManagerComponent's log.	OE	3.1.3.3.2.3.3	To ensure an unsuccessful unregisterComponent operation attempt is logged as a FAILURE_ALARM in the DomainManagerComponent's log.	1. Log Service is supported. 2. DomainManagerComponent's log contains a FAILURE_ALARM for each unregisterComponent operation attempt that was unsuccessful.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA198	The unregisterComponent operation shall disconnect any connections (including those made to the Event Service event channels) to the unregistering component indicated by the input identifier parameter.	OE	3.1.3.3.2.3.3	To ensure all connections for the component being unregistered are disconnected.	1. Identify all component's connection(s) including those made to the Event Service event channels as specified in the domain profile. 2. No connection identified in step 1 exists for the component being unregistered.
SCA199	Connections broken as a result of this unregisterComponent operation shall be considered as pending for future connections when the component to which the component was connected still exists.	OE	3.1.3.3.2.3.3	Ensure ports disconnected as a result of unregistering a component are maintained as pending by the DomainManager component.	Do Not verify - Not testable
SCA201	The registerComponent operation shall establish any connections for the DeviceManagerComponent indicated by the input registeringComponent parameter, which are specified in the connections element of the DeviceManagerComponent's DCD file, that are possible with the current set of registered components.	OE	3.1.3.3.2.3.3	To ensure the DomainManager establishes all possible connections defined in the DCD file of the registering DeviceManagerComponent.	For a registering DeviceManagerComponent, all connections defined in its DCD file that can possibly be connected are established.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA202	For connections established for an Event Service's event channel, the registerComponent operation shall connect a CosEventComm::PushConsumer or CosEventComm::PushSupplier object to the event channel as specified in the DCD's domainfinder element.	OE	3.1.3.3.2.3.3	To specify the Event Service interfaces that the DomainManager must use /provide to establish connections to an event channel defined in the domain profile.	1. Identify all of the Event Channel connections specified in the DeviceManager's DCD's domainfinder element. 2. For the registering component, all specified connections identified in step 1 are established.
SCA203	If the event channel does not exist, the registerComponent operation shall create the event channel.	OE	3.1.3.3.2.3.3	To ensure the domain manager creates the event channel if it does not exist.	For connections to be established in SCA 202 the event channel is created if it did not already exist.
SCA204	The registerComponent operation shall mount the DeviceManagerComponent's FileSystemComponent to the DomainManagerComponent's FileManagerComponent.	OE	3.1.3.3.2.3.3	To ensure the FileSystem of a registering DeviceManager is mounted to the DomainManager's FileManager.	After a DeviceManager has registered with the DomainManager the DeviceManager FileSystem is mounted to the DomainManager's FileManager.
SCA205	The mounted FileSystem name shall have the format, "/DomainName/HostName", where DomainName is the name of the domain and HostName is the identifier of the input registeringManager.	OE	3.1.3.3.2.3.3	To ensure a known location for the FileSystem.	When a DeviceManager successfully registered with the DomainManager, the DomainManager's FileManager must have a path starting with "/DomainName/HostName" for each mounted FileSystem.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA208	The unregisterComponent operation shall disconnect the established connections (including those made to the Event Service event channels) of the unregistering component as well as for its registered components that have not already been disconnected by the unregistering component when the unregisteringComponent's type field is DEVICE_MANAGER_COMPONENT.	OE	3.1.3.3.2.3.3	To ensure the DomainManager disconnects all established connections of the unregistering DeviceManager component and also disconnects the established connections for the components registered with the unregistering device manager.	1. For the unregistering DeviceManager, identify all of its connection(s) and those of its registered component(s) as specified in the domain profile. 2. No connection(s) identified in step 1 exists.
SCA210	The unregisterComponent operation shall unmount all DeviceManagerComponent's file systems from its FileManagerComponent when the unregistering component's type field is DEVICE_MANAGER_COMPONENT.	OE	3.1.3.3.2.3.3	To ensure the DomainManager unmounts the unregistering DeviceManager's file system(s) from the DomainManager's FileManager.	1. Identify all the unregistering DeviceManager file systems. 2. None of the file systems identified in step 1 are mounted by the DomainManager's FileManager.
SCA214	A DomainManagerComponent shall realize the DomainManager interface.	OE	3.1.3.3.2.3.4	To ensure a DomainManagerComponent provides a standard mechanism to configure the domain and manage devices, services and applications.	The DomainManagerComponent inherits the DomainManager interface.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA532	A DomainManagerComponent shall fulfill the BaseComponent requirements.	OE	3.1.3.3.2.3.4	To ensure a DomainManagerComponent satisfies the requirements of a BaseComponent.	Will be satisfied by the verification of BaseComponent requirements.
SCA559	A DomainManagerComponent shall realize the DomainInstallation interface.	OE	3.1.3.3.2.3.4	To ensure a DomainManager provides a standard mechanism to manage the installation of applications.	The DomainManagerComponent inherits the DomainInstallation interface.
SCA560	A DomainManagerComponent shall realize EventChannelRegistry interface.	OE	3.1.3.3.2.3.4	To ensure a DomainManager provides a standard mechanism to manage registration with the event channel.	The DomainManagerComponent inherits the EventChannelRegistry interface.
SCA215	A DeviceManagerComponent shall be accompanied by the appropriate Domain Profile files per section 3.1.3.6.	OE	3.1.3.3.2.4.3	To ensure a DeviceManagerComponent has its required domain profile file(s).	The DeviceManagerComponent has a DCD and all of its referenced domain profile file(s).
SCA216	A DeviceManagerComponent upon start up shall register with a DomainManagerComponent via the ComponentRegistry interface.	OE	3.1.3.3.2.4.3	To ensure the DomainManager has a complete list of registered DeviceManagers.	The DeviceManager's start up sequence calls the CF::ComponentRegistry::registerComponent of the DomainManager's ComponentRegistry interface.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA217	A DeviceManagerComponent shall create FileSystemComponents implementing the FileSystem interface for each OS file system.	OE	3.1.3.3.2.4.3	FileSystemComponent(s) are created for each file System listed in the DeviceManager's DCD.	A FileSystemComponent exists for each OS file system identified within the DeviceManager's DCD.
SCA218	If multiple FileSystemComponents are to be created, the DeviceManagerComponent shall mount created FileSystemComponents to a FileManagerComponent (widened to a FileSystemComponent through the CF::ManagerInfo's FileSys field).	OE	3.1.3.3.2.4.3	If multiple FileSystemComponents are created they are mounted to a single federated file system.	If multiple FileSystemComponents are created the DeviceManagerComponent contains a FileManagerComponent that has mounted all of the file systems identified within its DCD.
SCA224	A DeviceManagerComponent shall use the stacksize and priority elements as specified in the componentinstantiation element's SPD implementation code for the execute operation options parameter.	OE	3.1.3.3.2.4.3	To ensure the DeviceManagerComponent uses the stacksize and priority values defined in the code element of the implementation element in the SPD associated to the component instance when invoking the execute operation.	The stack size and priority values defined within the code element of the implementation element in the SPD associated to the component instance are passed to execute operation options parameter.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA226	The DeviceManagerComponent shall use the stacksize and priority elements as specified in the componentinstantiation element's SPD implementation code as qualifiers parameter for the ComponentFactory::createComponent operation.	OE	3.1.3.3.2.4.3	To ensure the DeviceManagerComponent uses the stacksize and priority values defined in the code element of the implementation element in the SPD associated to the component instance when invoking the ComponentFactory::createComponent operation.	The stack size and priority values defined within the code element of the implementation element in the SPD associated to the component instance are passed to ComponentFactory::createComponent operation qualifiers parameter.
SCA227	The DeviceManagerComponent shall initialize deployed components that are instantiated by the DeviceManagerComponent provided they realize the LifeCycle interface.	OE	3.1.3.3.2.4.3	To ensure the DeviceManagerComponent initializes the components it instantiates provided the component realizes the LifeCycle interface.	1. The instantiated component realizes the LifeCycle interface. 2. The component is initialized.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA228	After component initialization, the DeviceManagerComponent shall configure deployed components that are instantiated by the DeviceManagerComponent, provided they realize the PropertySet interface.	OE	3.1.3.3.2.4.3	To provide a standard sequence of operations to be performed on the components deployed by a DeviceManagerComponent.	Once the components have been initialized, the DeviceManagerComponent calls the configure operation for each component that it has instantiated and realizes the CF::PropertySet interface and has configure properties with readwrite or writeonly properties with initial values.
SCA229	The DeviceManagerComponent shall configure a DCD's componentinstantiation element provided the componentinstantiation element has configure readwrite or writeonly properties with values.	OE	3.1.3.3.2.4.3	To ensure that the components instantiated by a DeviceManagerComponent and realizing the CF::PropertySet interface are properly configured.	The DeviceManagerComponent calls the CF::PropertySet::configure for each component that it has instantiated with all the component's configurable properties. May be verified by SCA228.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA230	The DeviceManagerComponent shall register a registering component with the DomainManagerComponent when the DeviceManagerComponent has previously registered with the DomainManagerComponent.	OE	3.1.3.3.2.4.3	To ensure that all components that register to a DeviceManagerComponent are also registered with the DomainManagerComponent.	1. The DeviceManagerComponent is registered with the DomainManagerComponent. 2. The DeviceManagerComponent registers to the DomainManager the CF::ComponentType of a registering component further tagged with the specializedInfo field containing an id of DEVICE_MANAGER_ID and a value of the DeviceManagerComponent's identifier attribute.
SCA231	The registerComponent operation shall, upon unsuccessful component registration, write a FAILURE_ALARM log record to a domain manager's log.	OE	3.1.3.3.2.4.3	To ensure an unsuccessful registerComponent operation attempt is logged as a FAILURE_ALARM in the DomainManagerComponent's log.	1. Log Service is supported. 2. DomainManagerComponent's log contains a FAILURE_ALARM for each registerComponent operation attempt that was unsuccessful.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA232	The unregisterComponent operation shall, upon unsuccessful unregistration of a component, write a FAILURE_ALARM log record to a DomainManagerComponent's log.	OE	3.1.3.3.2.4.3	To ensure an unsuccessful unregisterComponent operation is logged as a FAILURE_ALARM in the DomainManagerComponent's log.	1. Log Service is supported. 2. DomainManagerComponent's log contains a FAILURE_ALARM for each unregisterComponent operation attempt that was unsuccessful.
SCA233	The unregisterComponent operation shall unregister the registered component specified by the input identifier parameter from the DomainManagerComponent if it is registered with the DeviceManagerComponent and the DeviceManagerComponent is not shutting down.	OE	3.1.3.3.2.4.3	Unregister the specified component that has registered with the domain manager.	1. The component identified by the input identifier parameter is registered with a device manager component. 2. the device manager component is not shutting down. 3. The component is unregistered from the DomainManager and also the DeviceManager (SCA573).

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA438	When a DeviceComponent is deployed via PlatformComponentFactoryComponent, the DeviceManagerComponent shall supply the following properties as the qualifiers parameter to the referenced ComponentFactory::createComponent operation: 1. Device Identifier - The ID is "DEVICE_ID" and the value is a string that corresponds to the DCD componentinstantiation id attribute; 2. Composite Device IOR - The ID is "Composite_DEVICE_IOR" and the value is a string that is an AggregateDeviceComponent stringified IOR (this parameter is only used when the DCD componentinstantiation element represents the child device of another componentinstantiation element); 3. The componentinstantiation componentfactoryref element properties whose kindtype element is "factoryparam".	OE	3.1.3.3.2.4.3	Specifies the mandatory parameters that a DeviceManagerComponent must provide to the ComponentFactory::createComponent operation when deploying a DeviceComponent using a PlatformComponentFactoryComponent.	For a device component deployed via a PlatformComponentFactoryComponent, the DeviceManagerComponent provides the following properties to the createComponent operation: - A property for which the ID is "DEVICE_ID" and the value is a string that corresponds to the DCD componentinstantiation id attribute; - A property for which the ID is "Composite_DEVICE_IOR" and the value is a string that is an AggregateDeviceComponent stringified IOR (if applicable); - Properties specified by the componentinstantiation componentfactoryref element properties whose kindtype element is "factoryparam" in the domain profile.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA439	When a ServiceComponent is deployed via a PlatformComponentFactoryComponent, the DeviceManagerComponent shall supply the following properties as the qualifiers parameter to the referenced PlatformComponentFactoryComponent's createComponent operation: 1. Service Name when the DCD componentinstantiation usagename element is non-null value - The ID is "SERVICE_NAME" and the value is a string in an "identifier\type" format that corresponds to the DCD componentinstantiation usagename element; 2. The componentinstantiation componentfactoryref element properties whose kindtype element is "factoryparam".	OE	3.1.3.3.2.4.3	Specifies the mandatory parameters that a DeviceManagerComponent must provide to the ComponentFactory::createComponent operation when deploying a ServiceComponent using a PlatformComponentFactoryComponent.	For a service component deployed via a PlatformComponentFactoryComponent, the DeviceManagerComponent provides the following properties to the createComponent operation: - A property for which the ID is "SERVICE_NAME" and the value is a string in an "identifier\type" format that corresponds to the DCD componentinstantiation usagename element; - Properties specified by the componentinstantiation componentfactoryref element properties whose kindtype element is "factoryparam" in the domain profile.

SCA442	When a DeviceComponent is deployed by the DeviceManagerComponent, the DeviceManagerComponent shall supply execute operation parameters for a device consisting of: 1. Component Registry IOR when the DCD componentinstantiation stringifiedobjectref element is null value - The ID is "COMPONENT_REGISTRY_IOR" and the value is a string that is the ComponentRegistry stringified IOR; 2. Device Identifier - The ID is "DEVICE_ID" and the value is a string that corresponds to the DCD componentinstantiation id attribute; 3. Composite Device IOR - The ID is "Composite_DEVICE_IOR" and the value is a string that is an AggregateDeviceComponent stringified IOR (this parameter is only used when the DCD componentinstantiation element represents the child device of another componentinstantiation element); 4. The execute ("execparam") properties as specified in the DCD for a componentinstantiation element (a	OE	3.1.3.3.2.4.3	Specifies the mandatory execute operation parameters that a DeviceManagerComponent must provide when deploying a DeviceComponent.	The DeviceManagerComponent provides the following executable parameters to the DeviceComponent: - A property for which the ID is "COMPONENT_REGISTRY_IOR" and the value is a string that corresponds to the ComponentRegistry stringified IOR (if applicable); - A property for which the ID is "DEVICE_ID" and the value is a string that corresponds to the DCD componentinstantiation id attribute; - A property for which the ID is "Composite_DEVICE_IOR" and the value is a string that is an AggregateDeviceComponent stringified IOR (if applicable); - The properties of the componentInstantiation element whose kindType is execparam specified in the domain profile.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
	DeviceManagerComponent passes execparam parameters' IDs and values as string values).				

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA449	If a PlatformComponentFactoryComponent is deployed by the DeviceManagerComponent, a DeviceManagerComponent shall supply execute operation parameters consisting of: 1. Component Registry IOR - The ID is "COMPONENT_REGISTRY_IOR" and the value is a string that is the ComponentRegistry stringified IOR when the DCD componentinstantiation stringifiedobjectref element is null value; 2. Component Identifier - The ID is "COMPONENT_IDENTIFIER" and the value is a string that corresponds to the DCD componentinstantiation id attribute; 3. The execute ("execparam") properties as specified in the DCD for a componentinstantiation element (a DeviceManagerComponent passes execparam parameters' IDs and values as string values).	OE	3.1.3.3.2.4.3	Specifies the mandatory execute operation parameters that a DeviceManagerComponent must provide when deploying a PlatformComponentFactoryComponent.	The DeviceManagerComponent provides the following executable parameters to the PlatformComponentFactoryComponent: - A property for which the ID is "COMPONENT_REGISTRY_IOR" and the value is a string that corresponds to the ComponentRegistry stringified IOR (if applicable); - A property for which the ID is "DEVICE_ID" and the value is a string that corresponds to the DCD componentinstantiation id attribute; - The properties of the componentInstantiation element whose kindType is execparam specified in the domain profile.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA538	If a ServiceComponent is deployed by the DeviceManagerComponent, a DeviceManagerComponent shall supply execute operation parameters consisting of: 1. Component Registry IOR - The ID is "COMPONENT_REGISTRY_IOR" and the value is a string that is the ComponentRegistry stringified IOR when the DCD componentinstantiation stringifiedobjectref element is null value; 2. Service Name when the DCD componentinstantiation usagename element is non-null value - The ID is "SERVICE_NAME" and the value is a string in an "identifier\type" format that corresponds to the DCD componentinstantiation usagename element; 3. The execute ("execparam") properties as specified in the DCD for a componentinstantiation element (a DeviceManagerComponent passes execparam parameters' IDs and values as string values).	OE	3.1.3.3.2.4.3	Specifies the mandatory execute operation parameters that a DeviceManagerComponent must provide when deploying a ServiceComponent.	The DeviceManagerComponent provides the following executable parameters to the ServiceComponent: - A property for which the ID is "COMPONENT_REGISTRY_IOR" and the value is a string that corresponds to the ComponentRegistry stringified IOR (if applicable); - A property for which the ID is "SERVICE_NAME" and the value is a string in an "identifier\type" format that corresponds to the DCD componentinstantiation usagename element; - The properties of the componentInstantiation element whose kindType is execparam specified in the domain profile.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA572	The DeviceManagerComponent shall assign a CF::ComponentType's specializedInfo allocation properties with an id of ALLOCATION_PROPS_ID and a value of type CF::AllocationProperties when no CF::AllocationProperties are supplied by the deployed BasePlatformComponent.	OE	3.1.3.3.2.4.3	To ensure a deployed BasePlatformComponent's specializedInfo field contains CF::AllocationProperties.	1. The BasePlatformComponent does not include CF::AllocationProperties when registering. 2. Deployed BasePlatformComponent specializedInfo contains an id of ALLOCATION_PROPS_ID and value of CF::AllocationProperties.
SCA573	The DeviceManagerComponent shall unregister the PlatformComponentFactoryComponent's BasePlatformComponents when a PlatformComponentFactoryComponent unregisters with the DeviceManagerComponent.	OE	3.1.3.3.2.4.3	Ensure all BasePlatformComponents instantiated by a PlatformComponentFactoryComponent are unregistered when the PlatformComponentFactoryComponent unregisters.	1. Identify the BasePlatformComponents instantiated by a PlatformComponentFactoryComponent specified within the DeviceManager's DCD. 2. As a result of PlatformComponentFactoryComponent unregistering, the BasePlatformComponents identified in Step 1 are not registered with DeviceManager.
SCA577	The registering DeviceManagerComponent CF::ComponentType's specializedInfo field shall contain a CF::ManagerInfo structure with an id of MANAGER_INFO_ID and a value of type CF::ManagerInfo that holds the BasePlatformComponents that have been deployed by or registered with a DeviceManagerComponent.	OE	3.1.3.3.2.4.3	To provide a standard structure for the DeviceManager to pass it's deployed and / or registered components.	The specialized info fields of the DeviceManagerComponent's CF::ComponentType has a property with an id of MANAGER_INFO_ID and a value of type CF::ManagerInfo structure that contains the components that have been deployed by and / or registered with the DeviceManagerComponent.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA234	A DeviceManagerComponent shall realize the ComponentIdentifier interface.	OE	3.1.3.3.2.4.4	To ensure a DeviceManagerComponent provides an identifier.	The DeviceManagerComponent inherits the ComponentIdentifier interface.
SCA235	A DeviceManagerComponent shall fulfill the BaseComponent requirements.	OE	3.1.3.3.2.4.4	To ensure a DeviceManagerComponent satisfies the requirements of a BaseComponent.	Will be satisfied by the verification of BaseComponent requirements.
SCA236	Each mounted file system name shall be unique within a DeviceManagerComponent.	OE	3.1.3.3.2.4.4	To ensure each file system mount point for a DeviceManagerComponent is uniquely named.	Names of file systems mount points for file systems of a DeviceManagerComponent are unique.
SCA561	A DeviceManagerComponent shall realize the DeploymentAttributes interface.	OE	3.1.3.3.2.4.4	To ensure a DeviceManager provides a standard mechanism to obtain deployment attributes.	The DeviceManagerComponent inherits the DeploymentAttributes interface.
SCA562	A DeviceManagerComponent shall realize the ReleasableManager interface.	OE	3.1.3.3.2.4.4	To ensure a DeviceManager provides a standard mechanism for terminating itself.	The DeviceManagerComponent inherits the ReleasableManager interface.
SCA243	The adminState attribute shall return the device's admin state value.	OE	3.1.3.4.1.1.4.1	To provide a standard mechanism to retrieve a device's admin state.	The adminState attribute returns the device's current administrative state.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA244	The adminState attribute shall only allow the setting of LOCKED and UNLOCKED values, where setting LOCKED is only effective when the adminState attribute value is UNLOCKED, and setting UNLOCKED is only effective when the adminState attribute value is LOCKED or SHUTTING_DOWN.	OE	3.1.3.4.1.1.4.1	Defines the constraints associated with setting the DeviceComponent's adminState attribute.	1. A DeviceComponent's adminState is set to LOCKED upon receiving a call to change its state to LOCKED only when its adminState attribute value is UNLOCKED. 2. A DeviceComponent's adminState is set to UNLOCKED upon receiving a call to change its state to UNLOCKED only when its adminState attribute value is LOCKED or SHUTTING_DOWN.
SCA248	The readonly usageState attribute shall return the device's usage state (IDLE, ACTIVE, or BUSY).	OE	3.1.3.4.1.2.4.1	To provide a standard mechanism to retrieve a device's usage state.	The device's usageState attribute returns the current usageState of the device.
SCA250	The allocateCapacity operation shall reduce the current capacities of the device based upon the input capacities parameter, when usageState attribute is not BUSY.	OE	3.1.3.4.1.2.5.1.3	To ensure that the device has the capacities needed by each component and that the current capacities are adjusted accordingly.	The DeviceComponent adjusts the capacities of the properties identified in the input parameter by their specified amount. Capacity adjustment is Device algorithm specific.
SCA251	The allocateCapacity operation shall set the device's usageState attribute to BUSY, when the device determines that it is not possible to allocate any further capacity.	OE	3.1.3.4.1.2.5.1.3	Provides an indication that a device cannot support additional capacity allocation.	Device usageState is changed to BUSY when a device's capacity has been met or exceeded.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA252	The allocateCapacity operation shall set the usageState attribute to ACTIVE, when capacity is being used and any capacity is still available for allocation.	OE	3.1.3. 4.1.2. 5.1.3	Provides an indication that a device can support additional capacity allocation.	Device usageState is ACTIVE when the DeviceComponent still has capacity after the current allocate capacity operation.
SCA253	The allocateCapacity operation shall only accept properties for the input capacities parameter which are simple properties whose kindtype is "allocation" and whose action element is "external" contained in the component's SPD.	OE	3.1.3. 4.1.2. 5.1.3	Specifies the permissible properties for the input capacities parameter of the allocateCapacity operation.	Identify the simple properties whose kindtype is "allocation" and whose action element is "external" contained in the component's profile. 2. The allocateCapacity operation only accepts the properties identified in step 1.
SCA254	The allocateCapacity operation shall return TRUE, if the capacities have been allocated, or FALSE, if not allocated.	OE	3.1.3. 4.1.2. 5.1.4	Provides indication for successful or unsuccessful capacity allocation.	allocateCapacity operation returns TRUE if all of the capacities identified in the input parameter have been successfully allocated and FALSE otherwise.
SCA255	The allocateCapacity operation shall raise the InvalidCapacity exception, when the input capacities parameter contains invalid properties or when attributes of those CF::Properties contain an unknown id or a value of the wrong data type.	OE	3.1.3. 4.1.2. 5.1.5	To ensure an allocateCapacity operation raises an exception when the input capacities parameter contains invalid properties or when attributes of those CF::Properties contain an unknown id or a value of the wrong data type.	The allocateCapacity operation raised an InvalidCapacity exception when the input capacities parameter contained invalid properties or when attributes of those CF::Properties contained an unknown id or a value of the wrong data type.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA257	The deallocateCapacity operation shall increment the current capacities of the device based upon the input capacities parameter.	OE	3.1.3.4.1.2.5.2.3	To ensure that the device adjusts its capacities that are deallocated.	The DeviceComponent adjusts the capacities of the properties identified in the input parameter by their specified amount.
SCA258	The deallocateCapacity operation shall set the usageState attribute to ACTIVE when, after adjusting capacities, any of the device's capacities are still being used.	OE	3.1.3.4.1.2.5.2.3	Provides an indication a portion of the device capacities have been allocated and the device can support additional capacity allocation.	Device usageState is ACTIVE when the DeviceComponent has capacity after the current deallocate capacity operation.
SCA259	The deallocateCapacity operation shall set the usageState attribute to IDLE when, after adjusting capacities, none of the device's capacities are still being used.	OE	3.1.3.4.1.2.5.2.3	Provides an indication that a device has none of its capacities are allocated.	DeviceComponent usageState is changed to IDLE when after adjusting the capacity of the properties identified by the input parameter by their specified amount and none of the device's capacities are being used.
SCA261	The deallocateCapacity operation shall raise the InvalidCapacity exception, when the capacity ID is invalid or the capacity value is the wrong type.	OE	3.1.3.4.1.2.5.2.5	To ensure the deallocateCapacity operation raises an exception when the input capacities parameter contains properties with an invalid ID or value of the wrong type.	The deallocateCapacity operation raised an InvalidCapacity exception when the input capacities parameter contained properties with an invalid ID or value of the wrong type.
SCA263	The readonly operationalState attribute shall return the device's operational state (ENABLED or DISABLED).	OE	3.1.3.4.1.3.4.1	To provide a standard mechanism to retrieve a device's operational state.	The operational state attribute returns the device's current operationalState.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA266	The readonly compositeDevice attribute shall return the object reference of the AggregateDeviceComponent.	OE	3.1.3.4.1.4.4.1	To provide a standard mechanism to retrieve a device's aggregate (parent) device.	The compositeDevice attribute returns an object reference to the device's aggregate (parent) device or a nil object reference when no aggregate device.
SCA267	The readonly compositeDevice attribute shall return a nil object reference when this DeviceComponent is not a parent.	OE	3.1.3.4.1.4.4.1	To ensure the compositeDevice attribute returns a nil object reference to indicate the DeviceComponent is not a parent.	The compositeDevice attribute returned a nil object reference for a DeviceComponent that is not a parent.
SCA268	The load operation shall load the file identified by the input fileName parameter on the DeviceComponent based upon the input loadKind parameter.	OE	3.1.3.4.1.5.5.1.3	Provides a standard mechanism for loading a file on a specific device.	The file specified by the input fileName is loaded by the DeviceComponent.
SCA269	Multiple loads of the same file as indicated by the input fileName parameter shall not result in an exception.	OE	3.1.3.4.1.5.5.1.3	To ensure multiple attempts to load the same file do not cause an exception.	No exceptions are raised when the same file is loaded 2 or more times.
SCA271	The load operation shall raise the InvalidLoadKind exception when the input loadKind parameter is not supported.	OE	3.1.3.4.1.5.5.1.5	To ensure an error is reported when the load operation is unable to load the type of file designated by the loadKind parameter.	1. The loadKind of the file to be loaded is not defined in the loadable device component's profile. 2. The InvalidLoadKind exception is raised.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA272	The load operation shall raise the CF::InvalidFileName exception when the file designated by the input fileName parameter cannot be found.	OE	3.1.3. 4.1.5. 5.1.5	To ensure the load operation raises an exception when the file specified by the input fileName parameter cannot be found.	The load operation raised a CF::InvalidFileName exception when the file specified by the input fileName parameter could not be found.
SCA273	The load operation shall raise the LoadFail exception when an attempt to load the device is unsuccessful.	OE	3.1.3. 4.1.5. 5.1.5	To ensure the load operation raises an exception when an attempt to load a device is unsuccessful.	The load operation raised a LoadFail exception when an attempt to load the device was unsuccessful.
SCA274	The unload operation shall unload the file identified by the input fileName parameter from the loadable device when the number of unload requests matches the number of load requests for the indicated file.	OE	3.1.3. 4.1.5. 5.2.3	Provides a standard mechanism for unloading a file on a specific device.	1. The number of unload requests equals the number of load requests. 2. The file specified by the input fileName is unloaded from the LoadableDeviceComponent.
SCA276	The unload operation shall raise the CF::InvalidFileName exception when the file designated by the input fileName parameter cannot be found.	OE	3.1.3. 4.1.5. 5.2.5	To ensure the unload operation raises an exception when the file designated by the input fileName parameter cannot be found.	The unload operation raised a CF::InvalidFileName exception when the file designated by the input fileName parameter could not be found.
SCA279	The execute operation shall execute the file identified by the input filename parameter using the input parameters and options parameters.	OE	3.1.3. 4.1.6. 5.1.3	Execute the file with the input parameters.	Verified by SCA280 and SCA282.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA280	The execute operation shall map the input parameters (id/value string pairs) parameter as an argument to the operating system "execute/thread" function.	OE	3.1.3. 4.1.6. 5.1.3	To ensure the input parameters are correctly passed to the target function being executed.	The input parameters are passed (e.g. argc/argv if loadKind is executable) to the target function being executed.
SCA282	The execute operation shall return a unique ExecutionID_Type for the process/thread that it created.	OE	3.1.3. 4.1.6. 5.1.4	Provides the identity of a created process/thread.	The execute operation returns an ExecutionID_Type structure containing the created process/thread's identifier.
SCA284	The execute operation shall raise the InvalidFunction exception when the function indicated by the input entry point options parameter is not executed because it does not exist on the device.	OE	3.1.3. 4.1.6. 5.1.5	To ensure the execute operation raises an exception when a function specified by the input entry point options parameter does not exist on the device.	The execute operation raised an InvalidFunction exception when a function specified by an input entry point options parameter did not exist on the device.
SCA285	The execute operation shall raise the CF::InvalidFileName exception when the file name indicated by the input filename parameter does not exist for the device to be executed.	OE	3.1.3. 4.1.6. 5.1.5	To ensure the execute operation raises an exception when the file identified by the input parameter file name does not exist.	The execute operation raised a CF::InvalidFileName exception when the file identified by input file name parameter did not exist.
SCA286	The execute operation shall raise the InvalidParameters exception when the input parameter ID or value attributes are not valid strings.	OE	3.1.3. 4.1.6. 5.1.5	To ensure the execute operation raises an exception when the input parameter ID or value attributes are not valid strings.	The execute operation raised an InvalidParameters exception when the input parameter ID or value attributes were not valid strings.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA287	The execute operation shall raise the InvalidOptions exception when the input options parameter does not comply with sections 3.1.3.4.1.6.3.6 STACK_SIZE_ID,3.1.3.4.1.6.3.7 PRIORITY_ID, 3.1.3.4.1.6.3.9 PROCESS_COLLOCATION_ID, 3.1.3.4.1.6.3.10, ENTRY_POINT_ID, and 3.1.3.4.1.6.3.11 CORE_AFFINITY_ID.	OE	3.1.3.4.1.6.5.1.5	To ensure the execute operation raises an exception when the input options parameter does not comply with valid options.	The execute operation raised an InvalidOptions exception when the input options parameter did not comply with any of the definitions for STACK_SIZE_ID, PRIORITY_ID, PROCESS_COLLOCATION_ID, ENTRY_POINT_ID, and CORE_AFFINITY_ID.
SCA288	The execute operation shall raise the ExecuteFail exception when the operating system "execute/thread" function is not successful.	OE	3.1.3.4.1.6.5.1.5	To ensure an execute operation raises an exception when the operating system cannot execute the "execute/thread" function successfully.	The execute operation raised an ExecuteFail exception when the operating system "execute/thread" function was not successful.
SCA289	The terminate operation shall terminate the execution of the process/thread designated by the executionId input parameter on the device.	OE	3.1.3.4.1.6.5.2.3	Terminate process / thread designated by the executionId input parameter.	Processes/threads associated with the input executionId no longer exist on the processing unit.
SCA291	The terminate operation shall raise the InvalidProcess exception when the executionId does not exist for the device.	OE	3.1.3.4.1.6.5.2.5	To ensure the terminate operation raises an exception when the executionId does not exist for the device.	The terminate operation raised an InvalidProcess exception when the executionId did not exist for the device.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA292	The readonly devices attribute shall return a list of devices that have been added to this device or a sequence length of zero if the device has no aggregation relationships with other devices.	OE	3.1.3.4.1.7.4.1	To provide a standard mechanism to retrieve a device's aggregated (child) devices.	1. Identify all the child devices of a device in the domain. 2a. The device returns an empty list when there are no child devices identified in step 1. 2b. The device returns the list of the child devices identified in step 1.
SCA293	The addDevice operation shall add the input associatedDevice parameter to the AggregateDevice's devices attribute when the associatedDevice associated with the input identifier parameter does not exist in the devices attribute.	OE	3.1.3.4.1.7.5.1.3	Defines a standard mechanism for adding a device to an AggregateDeviceComponent.	1a. If the input associatedDevice is not a member of the AggregateDevice's devices attribute it is added to the AggregateDevice's devices attribute. 1b. If the input associatedDevice is a member of the AggregateDevice's devices attribute it is not added again.
SCA295	The addDevice operation shall raise the CF::InvalidObjectReference when the input associatedDevice parameter is a nil object reference.	OE	3.1.3.4.1.7.5.1.5	To ensure the addDevice operation raises an exception when the input associatedDevice parameter is a nil object reference.	The addDevice operation raised a CF::InvalidObjectReference exception when the input associatedDevice parameter had a nil object reference.
SCA296	The removeDevice operation shall remove the device that corresponds to the input identifier parameter from the AggregateDevice's devices attribute.	OE	3.1.3.4.1.7.5.2.3	Ensure the child Device identified by the input parameter is removed from the parent Device's list of child Devices.	The device identified by the input identifier parameter is not present in the devices attribute of the parent AggregateDevice.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA297	The removeDevice operation shall raise the CF::InvalidObjectReference when the device that corresponds to the input identifier parameter is a nil object reference or does not exist in the AggregateDevice devices attribute.	OE	3.1.3.4.1.7.5.2.5	To ensure the removeDevice operation raises an exception when the device that corresponds to the input identifier parameter is a nil object reference or does not exist in the AggregateDevice devices attribute.	The removeDevice operation raised a CF::InvalidObjectReference exception when the device that corresponded to the input identifier parameter was a nil object reference or did not exist in the AggregateDevice devices attribute.
SCA299	The values associated with the parameters (COMPOSITE_DEVICE_IOR, and DEVICE_ID) as described in 3.1.3.3.2.4.3 shall be used to set the DeviceComponent's compositeDevice, and identifier attributes, respectively.	OE	3.1.3.4.2.1.3	To ensure that a child device component has the information necessary to identify itself.	The DeviceComponent sets its compositeDevice and identifier attributes to the values contained in the COMPOSITE_DEVICE_IOR and DEVICE_ID parameters passed to the executable.
SCA458	A child DeviceComponent shall add itself to a parent device using the executable Composite Device IOR and DEVICE_ID parameters per 3.1.3.3.2.4.3.	OE	3.1.3.4.2.1.3	A child device adds itself to its parent AggregateDevice.	The device identified by the input identifier parameter is present in the devices attribute of the parent AggregateDevice.
SCA241	The releaseObject operation shall unregister its device from its DeviceManagerComponent.	OE	3.1.3.4.2.1.3.1	To ensure the released device is unregistered from its DeviceManagerComponent.	The released device is no longer registered with the DeviceManager.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA237	The releaseObject operation shall assign the LOCKED state to the adminState attribute, when the adminState attribute is UNLOCKED.	OE	3.1.3.4.2.1.3.1	To ensure the device transitions to LOCKED prior to being released.	The device's adminState transitions from UNLOCKED to LOCKED when releaseObject is called.
SCA238	The releaseObject operation shall call the releaseObject operation on all of the DeviceComponents contained within its referenced AggregateDeviceComponent when the DeviceComponent is a parent device.	OE	3.1.3.4.2.1.3.1	To ensure all of the child DeviceComponents are released when the parent device is released.	All child DeviceComponents of the parent device are released.
SCA239	The releaseObject operation shall cause the removal of a DeviceComponent from the referenced AggregateDeviceComponent of its parent when this DeviceComponent is a child device.	OE	3.1.3.4.2.1.3.1	To ensure a child DeviceComponent is removed from its parent device when it is released.	1. The releaseObject operation is called on a child device. 2. The devices attribute of the aggregate device (parent) does not contain the released device.
SCA240	The releaseObject operation shall cause the device to be unavailable and released from the operating environment when the adminState attribute transitions to LOCKED.	OE	3.1.3.4.2.1.3.1	To ensure the device is made unavailable and released when the device's adminState is transitioned to LOCKED.	The device is released from the operating environment.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA245	The adminState attribute, upon being commanded to be LOCKED, shall set the adminState to LOCKED for its entire aggregation of DeviceComponents (if it has any).	OE	3.1.3.4.2.1.3.1	To ensure the adminState is set to LOCKED for all child DeviceComponent(s) when an AggregateDeviceComponent adminState is LOCKED.	1. AggregateDeviceComponent adminState is LOCKED 2. All DeviceComponents aggregated by the AggregateDeviceComponent have an adminState of LOCKED.
SCA247	The DeviceComponent shall send a StateChangeEventType event to the Incoming Domain Management event channel, whenever the adminState attribute changes.	OE	3.1.3.4.2.1.3.1	To ensure the DeviceComponent sends a StateChangeEventType event to the Incoming Domain Management event channel whenever the DeviceComponent adminState attribute changes state.	The DeviceComponent sent a StateChangeEventType event to the Incoming Domain Management event channel whenever the DeviceComponent adminState attribute changed state.
SCA249	The DeviceComponent shall send a StateChangeEventType event to the Incoming Domain Management event channel, whenever the usageState attribute changes.	OE	3.1.3.4.2.1.3.1	To ensure the DeviceComponent sends a StateChangeEventType event to the Incoming Domain Management event channel whenever the DeviceComponent usageState attribute changes state.	The DeviceComponent sent a StateChangeEventType event to the Incoming Domain Management event channel whenever the DeviceComponent usageState attribute changed state.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA256	The allocateCapacity operation shall raise the CF::InvalidState exception when the DeviceComponent's adminState is not UNLOCKED.	OE	3.1.3.4.2.1.3.1	To ensure the allocateCapacity operation raises an exception when the DeviceComponent's adminState is not UNLOCKED.	The allocateCapacity operation raised the CF::InvalidState exception when the DeviceComponent's adminState was not UNLOCKED.
SCA260	The deallocateCapacity operation shall set the adminState attribute to LOCKED as specified in this section.	OE	3.1.3.4.2.1.3.1	Provides an indication that the device is not available for use during releaseObject.	Do not verify. The Device does not exist after releaseObject completes.
SCA262	The deallocateCapacity operation shall raise the CF::InvalidState exception, when the DeviceComponent's adminState is LOCKED.	OE	3.1.3.4.2.1.3.1	To ensure the deallocateCapacity operation raises an exception when the DeviceComponent's adminState is LOCKED.	The deallocateCapacity operation raised the CF::InvalidState exception, when the DeviceComponent's adminState was LOCKED.
SCA264	The DeviceComponent shall send a StateChangeEventType event to the Incoming Domain Management event channel, whenever the operationalState attribute changes.	OE	3.1.3.4.2.1.3.1	To ensure the DeviceComponent sends a StateChangeEventType event to the Incoming Domain Management event channel whenever the DeviceComponent operationalState attribute changes state.	The DeviceComponent sent a StateChangeEventType event to the Incoming Domain Management event channel whenever the DeviceComponent operationalState attribute changed state.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA511	The allocateCapacity operation shall raise the CF::InvalidState exception when the DeviceComponent's operationalState is DISABLED.	OE	3.1.3.4.2.1.3.1	To ensure the allocateCapacity operation raises an exception when the DeviceComponent's operationalState is DISABLED.	The allocateCapacity operation raised a CF::InvalidState exception when the DeviceComponent's operationalState was DISABLED.
SCA516	The deallocateCapacity operation shall raise the CF::InvalidState exception, when the DeviceComponent's operationalState is DISABLED.	OE	3.1.3.4.2.1.3.1	To ensure the deallocateCapacity operation raises an exception when the DeviceComponent's operationalState is DISABLED.	The deallocateCapacity operation raised a CF::InvalidState exception when the DeviceComponent's operationalState was DISABLED.
SCA526	A DeviceComponent shall fulfill the BasePlatformComponent requirements.	OE	3.1.3.4.2.1.4	To ensure a DeviceComponent satisfies the requirements of a BasePlatformComponent.	Will be satisfied by the verification of BasePlatformComponent requirements.
SCA534	A DeviceComponent shall realize the DeviceAttributes interface.	OE	3.1.3.4.2.1.4	To ensure a DeviceComponent provides a standard mechanism to obtain its attributes.	The DeviceComponent inherits the DeviceAttributes interface.
SCA535	A DeviceComponent shall realize the AdministratableInterface interface.	OE	3.1.3.4.2.1.4	To ensure a DeviceComponent provides a standard mechanism to manage its administrative state.	The DeviceComponent inherits the AdministratableInterface interface.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA536	A DeviceComponent shall realize the CapacityManagement interface.	OE	3.1.3.4.2.1.4	To ensure a DeviceComponent provides a standard mechanism to manage its capacities.	The DeviceComponent inherits the CapacityManagement interface.
SCA539	A DeviceComponent shall realize the AggregateDeviceAttributes interface.	OE	3.1.3.4.2.1.4	To ensure a DeviceComponent provides a standard mechanism to obtain its aggregate Device (parent).	The DeviceComponent inherits the AggregateDeviceAttributes interface.
SCA563	A DeviceComponent shall realize the LifeCycle interface.	OE	3.1.3.4.2.1.4	To ensure a DeviceComponent provides a standard mechanism to manage its life cycle.	The DeviceComponent inherits the LifeCycle interface.
SCA306	The load operation shall support the load types as stated in the LoadableDeviceComponent's profile supported_load_types allocation property.	OE	3.1.3.4.2.2.3	Ensures the load types specified in the LoadableDeviceComponent's profile are supported in the load operation.	1. Identify the file types (i.e. Executable, KernelModule, SharedLibrary, and Driver) specified in the Domain profile for the Device. See SCA 4.1 Appendix D Attachment 1. 2. The LoadableDeviceComponent's CF::LoadableInterface::load operation does not raise an InvalidLoadKind exception because of the file types listed in 1 above.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA307	When a LoadType is not defined for the LoadableDeviceComponent, the load operation shall support all SPD code element types.	OE	3.1.3.4.2.2.3	To ensure all types of files can be loaded on a LoadableDeviceComponent when no LoadType is defined in its profile.	1. The LoadableDeviceComponent does not have supported_load_types property defined in its profile. 2. The load operation does not raise the InvalidLoadKind exception.
SCA270	The load operation shall raise the CF::InvalidState exception if upon entry the LoadableDeviceComponent's adminState attribute is either LOCKED or SHUTTING_DOWN.	OE	3.1.3.4.2.2.3.1	To ensure the load operation raises an exception when the LoadableDeviceComponent's adminState attribute is either LOCKED or SHUTTING_DOWN.	The load operation raised the CF::InvalidState exception when the LoadableDeviceComponent's adminState attribute was either LOCKED or SHUTTING_DOWN.
SCA275	The unload operation shall raise the CF::InvalidState exception if upon entry the LoadableDeviceComponent's adminState attribute is LOCKED.	OE	3.1.3.4.2.2.3.1	To ensure the unload operation raises an exception when the LoadableDeviceComponent's adminState attribute is LOCKED.	The unload operation raised an InvalidState exception when the LoadableDeviceComponent's adminState attribute was LOCKED.
SCA512	The load operation shall raise the CF::InvalidState exception if upon entry the LoadableDeviceComponent's operationalState attribute is DISABLED.	OE	3.1.3.4.2.2.3.1	To ensure the load operation raises an exception when the LoadableDeviceComponent's operationalState attribute is DISABLED.	The load operation raised the CF::InvalidState exception when the LoadableDeviceComponent's operationalState attribute was DISABLED.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA513	The unload operation shall raise the CF::InvalidState exception if upon entry the LoadableDeviceComponent's operationalState attribute is DISABLED.	OE	3.1.3.4.2.2.3.1	To ensure the unload operation raises an exception when the LoadableDeviceComponent's operationalState attribute is DISABLED.	The unload operation raised an InvalidState exception when the LoadableDeviceComponent's operationalState attribute was DISABLED.
SCA308	A LoadableDeviceComponent shall realize the LoadableInterface interface.	OE	3.1.3.4.2.2.4	To ensure a LoadableDeviceComponent provides a standard mechanism to load and unload software / firmware.	The LoadableDeviceComponent inherits the LoadableInterface interface.
SCA309	A LoadableDeviceComponent shall fulfill the DeviceComponent requirements.	OE	3.1.3.4.2.2.4	To ensure a LoadableDeviceComponent satisfies the requirements of a DeviceComponent.	Will be satisfied by the verification of DeviceComponent requirements.
SCA310	An ExecutableDeviceComponent shall accept the executable parameters as specified in section 3.1.3.4.1.6.5.1.3 (ExecutableInterface::execute).	OE	3.1.3.4.2.3.3	Ensure the execute operation accepts the specified execute parameters.	Verified by SCA282
SCA283	The execute operation shall raise the CF::InvalidState exception if upon entry the ExecutableDeviceComponent's adminState attribute is either LOCKED or SHUTTING_DOWN.	OE	3.1.3.4.2.3.3.1	To ensure an exception is raised when the ExecutableDeviceComponent's adminState attribute is either LOCKED or SHUTTING_DOWN.	The execute operation raised the CF::InvalidState exception when the ExecutableDeviceComponent's adminState attribute was either LOCKED or SHUTTING_DOWN.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA290	The terminate operation shall raise the CF::InvalidState exception if upon entry the ExecutableDeviceComponent's adminState attribute is LOCKED.	OE	3.1.3.4.2.3.3.1	To ensure the terminate operation raises an exception when the ExecutableDeviceComponent's adminState attribute is LOCKED.	The terminate operation raised a CF::InvalidState exception when the ExecutableDeviceComponent's adminState attribute was LOCKED.
SCA514	The execute operation shall raise the CF::InvalidState exception if upon entry the ExecutableDeviceComponent's operationalState attribute is DISABLED.	OE	3.1.3.4.2.3.3.1	To ensure the execute operation raises an exception when the ExecutableDeviceComponent's adminState attribute is DISABLED.	The execute operation raised the CF::InvalidState exception when the ExecutableDeviceComponent's adminState attribute was DISABLED.
SCA515	The terminate operation shall raise the CF::InvalidState exception if upon entry the ExecutableDeviceComponent's operationalState attribute is DISABLED.	OE	3.1.3.4.2.3.3.1	To ensure the terminate operation raises an exception when the ExecutableDeviceComponent's operationalState attribute is DISABLED.	The terminate operation raised an CF::InvalidState exception when the ExecutableDeviceComponent's operationalState attribute was DISABLED.
SCA311	An ExecutableDeviceComponent shall realize the ExecutableInterface interface.	OE	3.1.3.4.2.3.4	To ensure an ExecutableDeviceComponent provides a standard mechanism to execute and terminate software / firmware.	The ExecutableDeviceComponent inherits the ExecutableInterface interface.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA312	An ExecutableDeviceComponent shall fulfill the DeviceComponent requirements.	OE	3.1.3.4.2.3.4	To ensure an ExecutableDeviceComponent satisfies the requirements of a DeviceComponent.	Will be satisfied by the verification of DeviceComponent requirements.
SCA564	An ExecutableDeviceComponent shall realize the LoadableInterface interface.	OE	3.1.3.4.2.3.4	To ensure an ExecutableDeviceComponent is able to provide a standard mechanism to load and unload software / firmware.	The ExecutableDeviceComponent inherits the LoadableInterface interface.
SCA313	An AggregateDeviceComponent shall realize the AggregateDevice interface.	OE	3.1.3.4.2.4.4	To ensure an AggregateDeviceComponent provides a standard mechanism to manage its aggregated (child) Devices.	The AggregateDeviceComponent inherits the AggregateDevice interface.
SCA320	The readonly fileName attribute shall return the pathname used as the input fileName parameter of the FileSystem::create operation when the file was created.	OE	3.1.3.5.1.1.4.1	To provide a standard mechanism to retrieve the pathname of the file.	The fileName attribute includes the pathname of the created file which is the same as the input filename provided in the FileSystem::create operation.
SCA321	The readonly filePointer attribute shall return the current file position.	OE	3.1.3.5.1.1.4.2	To provide a standard mechanism to retrieve the current file position.	The filePointer attribute returns the current file position.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA322	The read operation shall read, from the referenced file, the number of octets specified by the input length parameter and advance the value of the filePointer attribute by the number of octets actually read.	OE	3.1.3.5.1.1.5.1.3	To ensure standard behavior of read operation when not end-of-file.	1. The requested number of octets are returned with correct content 2. The filePointer is advanced by the number of octets read.
SCA323	The read operation shall read less than the number of octets specified in the input length parameter, when an end-of-file is encountered.	OE	3.1.3.5.1.1.5.1.3	To ensure standard behavior when the read operation encounters end-of-file.	1. The read operation returns the number of octets with the correct content between the file pointer and the end-of-file when end-of-file is encountered. 2. Verify with SCA 322
SCA324	The read operation shall return a CF::OctetSequence that equals the number of octets actually read from the file via the out data parameter.	OE	3.1.3.5.1.1.5.1.4	To ensure standard content of the read operation's return.	The read operation returns CF::OctetSequence containing the content of the file octets that were actually read.
SCA325	If the filePointer attribute value reflects the end of the file, the read operation shall return a zero-length CF::OctetSequence.	OE	3.1.3.5.1.1.5.1.4	Specifies the read operation return value when the file pointer is at the end of the file when the read operation is invoked.	1. The filePointer attribute value is at end of the file. 2. The read operation returns a zero-length CF::OctetSequence.
SCA326	The read operation shall raise the IOException when a read error occurs.	OE	3.1.3.5.1.1.5.1.5	To ensure the read operation raises an exception when a read error occurs.	The read operation raised a CF::File::IOException exception when a read error occurred.
SCA327	The write operation shall write data to the file referenced.	OE	3.1.3.5.1.1.5.2.3	To ensure standard behavior of the write operation.	Referenced file contains the content from the write operation data parameter.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA328	The write operation shall increment the filePointer attribute to reflect the number of octets written, when the operation is successful.	OE	3.1.3. 5.1.1. 5.2.3	To ensure standard behavior of the filePointer attribute when writing to a file is successful.	The filePointer is advanced by the number of octets written to the file when the write operation is successful.
SCA329	If the write operation is unsuccessful, the value of the filePointer attribute shall maintain or be restored to its value prior to the write operation call.	OE	3.1.3. 5.1.1. 5.2.3	Ensure the filePointer remains unchanged if write operation fails.	The filePointer is unchanged when the write is unsuccessful.
SCA330	The write operation shall raise the IOException when a write error occurs.	OE	3.1.3. 5.1.1. 5.2.5	To ensure the write operation raises an exception when a write error occurs.	The write operation raised a CF::File::IOException exception when a write error occurred.
SCA331	The sizeOf operation shall return the number of octets stored in the file.	OE	3.1.3. 5.1.1. 5.3.4	To ensure standard behavior of the sizeOf operation.	The sizeOf operation returned the number of octets stored in the file.
SCA443	The sizeOf operation shall raise the CF::FileException when a file-related error occurs (e.g., file does not exist anymore).	OE	3.1.3. 5.1.1. 5.3.5	To define a standard mechanism for reporting errors when the sizeOf operation fails.	1. A file related error occurs when calling the sizeOf operation. 2. The CF::FileException is raised.
SCA332	The close operation shall release any OE file resources associated with the component.	OE	3.1.3. 5.1.1. 5.4.3	Ensure all OE file resources associated with the CF::File are released when the file is closed.	1. The CF::File is closed. 2. All OE file resources associated with the closed CF::File are released.
SCA333	The close operation shall make the file unavailable to the component.	OE	3.1.3. 5.1.1. 5.4.3	Ensure a client cannot use a closed CF::File.	CF:: File is not accessible by the client.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA334	The close operation shall raise the CF::FileException when it cannot successfully close the file.	OE	3.1.3.5.1.1.5.4.5	To ensure the close operation raises an exception when the file cannot be successfully closed.	The close operation raised the CF::FileException exception when it could not successfully close the file.
SCA335	The setFilePointer operation shall set the filePointer attribute value to the input filePointer.	OE	3.1.3.5.1.1.5.5.3	To ensure standard behavior of the setFilePointer operation.	1. No exception occurs 2.The filePointer attribute value is equal to the input filePointer.
SCA336	The setFilePointer operation shall raise the CF::FileException when the file pointer for the referenced file cannot be set to the value of the input filePointer parameter.	OE	3.1.3.5.1.1.5.5.5	To ensure the setFilePointer operation raises an exception when the file pointer for the referenced file cannot be set to the value of the input filePointer parameter.	The setFilePointer operation raised a CF::FileException exception when the file pointer could not be set to the value of the input filePointer parameter.
SCA337	The setFilePointer operation shall raise the InvalidFilePointer exception when the value of the filePointer parameter exceeds the file size.	OE	3.1.3.5.1.1.5.5.5	To ensure the setFilePointer operation raises an exception when the value of filepointer parameter exceeds the file size.	The setFilePointer operation raised the InvalidFilePointer exception when the filepointer parameter value exceeded the file size.
SCA338	At a minimum, the FileSystem interface implementation shall support name, kind, and size information for a file.	OE	3.1.3.5.1.2.3.3	To ensure a minimum set of file attributes is supported by all FileSystems.	The FileInformationType structure contains at minimum the values for the name, kind and size attributes.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA445	For this property, the identifier is CREATED_TIME_ID and the value shall be an unsigned long long data type containing the number of seconds since 00:00:00 UTC, Jan 1, 1970.	OE	3.1.3.5.1.2.3.6	To specify a standard property name, data type, and value for the file creation time.	1. CREATED_TIME_ID property is supported. 2. The value of the CREATED_TIME_ID property equals the time the file has been created in the format specified by the requirement.
SCA446	For this property, the identifier is MODIFIED_TIME_ID and the value shall be an unsigned long long data type containing the number of seconds since 00:00:00 UTC, Jan 1, 1970.	OE	3.1.3.5.1.2.3.7	To specify a standard property name, data type, and value for the file modification time.	1. MODIFIED_TIME_ID property is supported. 2. The value of the MODIFIED_TIME_ID property equals the time the file was last modified in the format specified by the requirement.
SCA447	For this property, the identifier is LAST_ACCESS_TIME_ID and the value shall be an unsigned long long data type containing the number of seconds since 00:00:00 UTC, Jan 1, 1970.	OE	3.1.3.5.1.2.3.8	To specify a standard property name, data type, and value for the file last access time.	1. LAST_ACCESS_TIME_ID property is supported. 2. The value of the LAST_ACCESS_TIME_ID property equals the time the file was last accessed in the format specified by the requirement.
SCA339	The remove operation shall remove the plain file which corresponds to the input fileName parameter.	OE	3.1.3.5.1.2.5.1.3	To ensure the standard behavior of the remove operation, which affects only files that are not directories.	The plain file specified in the input fileName parameter no longer exists.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA340	The remove operation shall raise the CF::InvalidFileName exception when the input fileName parameter is not a valid absolute pathname.	OE	3.1.3.5.1.2.5.1.5	To ensure the remove operation raises an exception when the input fileName parameter is not a valid absolute pathname.	The remove operation raised the CF::InvalidFileName exception when the input fileName parameter was not a valid absolute pathname.
SCA341	The remove operation shall raise the CF::FileException when a file-related error occurs.	OE	3.1.3.5.1.2.5.1.5	To ensure the remove operation raises an exception when a file-related error occurs.	The remove operation raised the CF::FileException when a file-related error occurred.
SCA342	The copy operation shall copy the source file identified by the input sourceFileName parameter to the destination file identified by the input destinationFileName parameter.	OE	3.1.3.5.1.2.5.2.3	To ensure the standard behavior of the copy operation, which affects only files that are not directories.	1. The sourceFileName does not equal the destinationFileName. 2. The destination file does not exist when the copy operation is invoked. 3. After the copy operation completes a file exists with the name of the destinationFileName parameter that has identical content to the sourceFileName file.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA343	The copy operation shall overwrite the destination file, when the destination file already exists and is not identical to the source file.	OE	3.1.3. 5.1.2. 5.2.3	To ensure the standard behavior of the file copy operation when a file with destinationFileName exists and destinationFilleName is not equal to the sourceFileName.	1.The sourceFileName does not equal the destinationFileName. 2. The destination file does exist when the copy operation is invoked. 3. After the copy operation completes a file exists with the name of the destinationFileName parameter that has identical content to the sourceFileName file.
SCA344	The copy operation shall raise the CF::FileException exception when a file-related error occurs.	OE	3.1.3. 5.1.2. 5.2.5	To ensure the copy operation raises an exception when a file-related error occurs.	The copy operation raised the CF::FileException exception when a file-related error occurred.
SCA345	The copy operation shall raise the CF::InvalidFileName exception when the destination pathname is identical to the source pathname.	OE	3.1.3. 5.1.2. 5.2.5	To ensure the copy operation raises an exception when the destination pathname is identical to the source pathname.	The copy operation raised the CF::InvalidFileName exception when the destination pathname was identical to the source pathname.
SCA346	The copy operation shall raise the CF::InvalidFileName exception when the sourceFileName or destinationFileName input parameter is not a valid absolute pathname.	OE	3.1.3. 5.1.2. 5.2.5	To ensure the copy operation raises an exception when the sourceFileName or destinationFileName input parameter is not a valid absolute pathname.	The copy operation raised the CF::InvalidFileName exception when the sourceFileName or destinationFileName input parameter was not a valid absolute pathname.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA347	The exists operation shall check to see if a file exists based on the fileName parameter.	OE	3.1.3. 5.1.2. 5.3.3	To determine whether a specific file is present in the file system.	Verified by conformance to SCA348.
SCA348	The exists operation shall return TRUE if the file exists, or FALSE if it does not.	OE	3.1.3. 5.1.2. 5.3.4	To ensure the standard behavior of the exists operation.	1a. Returns TRUE when a file with fileName exists in the file system. 1b. Returns FALSE when a file with fileName does not exist in the file system.
SCA349	The exists operation shall raise the CF::InvalidFileName exception when input fileName parameter is not a valid absolute pathname.	OE	3.1.3. 5.1.2. 5.3.5	To ensure the exists operation raises an exception when the input fileName parameter does not contain a valid absolute pathname.	The exists operation raised the CF::InvalidFileName exception when the input fileName parameter did not contain a valid absolute pathname.
SCA350	These wildcards shall only be applied following the right-most forward-slash character ("/") in the pathname contained in the input pattern parameter.	OE	3.1.3. 5.1.2. 5.4.3	To ensure standard placement of wildcards in a path name.	Verified by conformance to SCA448.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA448	The list operation shall support the "*" and "?" wildcard characters (used to match any sequence of characters, including null, and any single character, respectively).	OE	3.1.3. 5.1.2. 5.4.3	Specifies standard wildcard characters "*" and "?" to be used by the list operation to retrieve file information.	1. Verify that all wildcard characters follow the right-most slash "/". 2. Verify that file information is returned for files that match the name provided in the input pattern parameter where "*" matches any number of characters. 3. Verify that file information is returned for files that match the name provided in the input pattern parameter where "?" matches any single character.
SCA351	The list operation shall return a FileInformationSequence for files that match the search pattern specified in the input pattern parameter.	OE	3.1.3. 5.1.2. 5.4.4	To retrieve file information for files specified by the input search pattern.	The FileInformationType for each file matching the input search pattern is returned.
SCA352	The list operation shall return a zero length sequence when no file is found which matches the search pattern.	OE	3.1.3. 5.1.2. 5.4.4	To ensure a standard behavior for the list operation when no file matches the input search pattern.	A zero length sequence is returned when no file matches the input search pattern.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA353	The list operation shall raise the CF::InvalidFileName exception when the input pattern parameter is not an absolute pathname or cannot be interpreted due to unexpected characters.	OE	3.1.3. 5.1.2. 5.4.5	To ensure the list operation raises an exception when the input pattern parameter is not an absolute pathname or cannot be interpreted due to invalid characters.	To ensure the list operation raised a CF::InvalidFileName exception when the input pattern parameter was not an absolute pathname or could not be interpreted due to invalid characters.
SCA354	The list operation shall raise the CF::FileException when a file-related error occurs.	OE	3.1.3. 5.1.2. 5.4.5	To ensure the list operation raises an exception when a file-related error occurs.	The list operation raised the CF::FileException exception when a file-related error occurred.
SCA355	The create operation shall create a new file based upon the input fileName parameter.	OE	3.1.3. 5.1.2. 5.5.3	To provide a standard mechanism to create a new file.	1. The file with the input fileName does not exist. 2. After invoking the create operation the file specified by the input fileName parameter exists.
SCA356	The create operation shall return a file object reference to the created file.	OE	3.1.3. 5.1.2. 5.5.4	To ensure the file reference is provided for the created file.	The create operation returns a reference to the created file.
SCA357	The create operation shall raise the CF::FileException if the file already exists or another file error occurred.	OE	3.1.3. 5.1.2. 5.5.5	To ensure the create operation raises an exception when the file associated with the input fileName parameter already exists or another file error occurred.	The create operation raised the CF::FileException when the file associated with the input fileName parameter already exists or another file error occurred.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA358	The create operation shall raise the CF::InvalidFileName exception when the input fileName parameter is not a valid absolute pathname.	OE	3.1.3. 5.1.2. 5.5.5	To ensure the create operation raises an exception when the input fileName parameter is not a valid absolute pathname.	The create operation raised the CF::InvalidFileName exception when the input fileName parameter was not a valid absolute pathname.
SCA359	The open operation shall open the file referenced by the input fileName parameter.	OE	3.1.3. 5.1.2. 5.6.3	To provide a standard mechanism to open an existing file.	1. The file with the referenced input fileName exists. 2. The file with referenced name in the input fileName parameter is opened.
SCA360	The open operation shall open the file with read-only access when the input read_Only parameter is TRUE.	OE	3.1.3. 5.1.2. 5.6.3	To provide a standard mechanism to open an existing file with read-only access.	Verified by conformance to SCA322 and SCA329.
SCA361	The open operation shall open the file for write access when the input read_Only parameter is FALSE.	OE	3.1.3. 5.1.2. 5.6.3	To provide a standard mechanism to open an existing file with read and write access.	Verified by conformance to SCA327.
SCA362	The open operation shall return a FileComponent reference for the opened file.	OE	3.1.3. 5.1.2. 5.6.4	To ensure a file reference is provided for the opened file.	The open operation returns a reference to the opened file.
SCA363	The open operation shall set the filePointer attribute of the returned file instance to the beginning of the file.	OE	3.1.3. 5.1.2. 5.6.4	To ensure the filePointer position is set to the beginning when a file is opened.	The file is open and its filePointer points to the beginning of the file.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA364	The open operation shall raise the CF::FileException if the file does not exist or another file error occurred.	OE	3.1.3. 5.1.2. 5.6.5	To ensure the open operation raises an exception when the file does not exist or another file error occurred.	The open operation raised the CF::FileException when the file did not exist or another file error occurred.
SCA365	The open operation shall raise the CF::InvalidFileName exception when the input fileName parameter is not a valid absolute pathname.	OE	3.1.3. 5.1.2. 5.6.5	To ensure the open operation raises an exception when the input fileName parameter is not a valid absolute pathname.	The open operation raised the CF::InvalidFileName when the input fileName parameter was not a valid absolute pathname.
SCA366	The mkdir operation shall create a file system directory based on the directoryName given.	OE	3.1.3. 5.1.2. 5.7.3	To provide a standard mechanism to create a directory.	1. A directory with the input directoryName parameter does not exist. 2. After the mkdir operation is invoked a directory named the input directoryName parameter exists.
SCA367	The mkdir operation shall create all parent directories required to create the directoryName path given.	OE	3.1.3. 5.1.2. 5.7.3	To provide a standard mechanism for directory creation when the name provided includes parent directories that do not exist.	1. One or more of the parent directories of the directory specified by the input directoryName parameter do not exist. 2. The parent directories of the directory specified by the input directoryName parameter exist after invoking the mkdir operation.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA368	The mkdir operation shall raise the CF::FileException if the directory indicated by the input directoryName parameter already exists or if a file-related error occurred during the operation.	OE	3.1.3. 5.1.2. 5.7.5	To ensure the mkdir operation raises an exception if the directory indicated by the input directoryName parameter already exists or if a file-related error occurred during the operation.	The mkdir operation raised the CF::FileException exception if the directory indicated by the input directoryName parameter already existed or if a file-related error occurred during the operation.
SCA369	The mkdir operation shall raise the CF::InvalidFileName exception when the directoryName is not a valid directory name.	OE	3.1.3. 5.1.2. 5.7.5	To ensure the mkdir operation raises an exception when the input directoryName parameter is not a valid directory name.	The mkdir operation raised the CF::FileException exception when the input directoryName parameter was not a valid directory name.
SCA370	The rmdir operation shall remove the directory identified by the input directoryName parameter.	OE	3.1.3. 5.1.2. 5.8.3	To provide a standard mechanism for removing empty directories.	1. The directory with the input directoryName exists and is empty. 2. The empty directory is removed after the rmdir operation has been invoked.
SCA371	The rmdir operation shall not remove the directory identified by the input directoryName parameter when the directory contains files.	OE	3.1.3. 5.1.2. 5.8.3	To ensure a non-empty directory is not removed.	1. The directory with the input directoryName exists and is not empty. 2. The Directory and its contents exist after invoking rmdir operation.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA372	The rmdir operation shall raise the CF::FileException when the directory identified by the input directoryName parameter does not exist, the directory contains files, or an error occurs which prohibits the directory from being deleted.	OE	3.1.3.5.1.2.5.8.5	To ensure the rmdir operation raises an exception when the directory identified by the input directoryName parameter does not exist, the directory to be removed contains files, or an error occurs which prohibits the directory from being deleted.	The rmdir operation raised the CF::FileException when the directory identified by the input directoryName parameter did not exist, the directory to be removed contained files, or an error occurred which prohibits the directory from being deleted.
SCA373	The rmdir operation shall raise the CF::InvalidFileName exception when the input directoryName parameter is not a valid path prefix.	OE	3.1.3.5.1.2.5.8.5	To ensure the rmdir operation raises an exception when the input directoryName parameter is not a valid path prefix.	The rmdir operation raised the CF::InvalidFileName exception when the input directoryName parameter was not a valid path prefix.
SCA374	The query operation shall return file system information to the calling client based upon the given fileSystemProperties' ID.	OE	3.1.3.5.1.2.5.9.3	To provide a standard mechanism to obtain file system property information.	Verified by conformance to SCA440.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA440	The FileSystem::query operation shall recognize and provide the designated return values for the following fileSystemProperties (section 3.1.3.5.1.2.3.2): 1. SIZE - an ID value of "SIZE" causes the query operation to return an unsigned long long containing the file system size (in octets); 2. AVAILABLE_SPACE - an ID value of "AVAILABLE_SPACE" causes the query operation to return an unsigned long long containing the available space on the file system (in octets).	OE	3.1.3.5.1.2.5.9.3	To ensure that the FileSystem query operation returns the value for each of the recognized properties that are requested.	1. When "SIZE" is requested, the FileSystem component returns the number of octets allocated to the FileSystem as an unsigned long long type. 2. When "AVAILABLE_SPACE" is requested, the file system returns the number of unused octets in the FileSystem as an unsigned long long type.
SCA375	The query operation shall raise the UnknownFileSystemProperties exception when the given file system property is not recognized.	OE	3.1.3.5.1.2.5.9.5	To define a standard mechanism for reporting errors when the input query property is unknown.	1. One or more properties specified in the input query parameter is not defined by the FileSystem component. 2. The UnknownFileSystemProperties exception containing the unknown properties is raised.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA376	The mount operation shall associate the specified file system with the mount point referenced by the input mountPoint parameter.	OE	3.1.3. 5.1.3. 5.1.3	To provide a standard mechanism to make the file system accessible at the directory path specified by the mount point.	Verified by conformance to SCA382.
SCA377	A mount point name shall begin with a "/" (forward slash character).	OE	3.1.3. 5.1.3. 5.1.3	Establishes a mount point naming convention.	The mountPoint parameter begins with "/" (forward slash character)
SCA378	The mount operation shall raise the MountPointAlreadyExists exception when the mount point already exists in the file manager.	OE	3.1.3. 5.1.3. 5.1.5	To define a standard mechanism for reporting errors when the directory path in the mountPoint parameter already exists.	1. A mount point exists with a name identical to that of the mountPoint parameter. 2. The MountPointAlreadyExists exception is raised.
SCA379	The mount operation shall raise the InvalidFileSystem exception when the input FileSystem is a null object reference.	OE	3.1.3. 5.1.3. 5.1.5	To ensure the mount operation raises an exception when the input FileSystem is a null object reference.	The mount operation raised the InvalidFileSystem exception when the input FileSystem was a null object reference.
SCA461	The mount operation shall raise the CF::InvalidFileName exception when the input mount point does not conform to the file name syntax in section 3.1.3.5.2.2.3.	OE	3.1.3. 5.1.3. 5.1.5	To ensure the mount operation raises an exception when the input mount point does not conform to the standard file name syntax.	The mount operation raised the CF::FileException exception when the input mount point did not conform to the file name syntax in section 3.1.3.5.2.2.3.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA380	The unmount operation shall remove a mounted file system from the file manager whose mounted name matches the input mountPoint name.	OE	3.1.3. 5.1.3. 5.2.3	To provide a standard mechanism for dismounting a file system.	1. FileManagerComponent contains file matching the name of the input mountPoint name. 2. After invoking unmount the FileManagerComponent does not contain a file system matching the name of the input mountPoint name.
SCA381	The unmount operation shall raise the NonExistentMount exception when the mount point does not exist.	OE	3.1.3. 5.1.3. 5.2.5	To ensure the unmount operation raises an exception when the mount point does not exist.	The unmount operation raised a NonExistentMount exception when the mount point did not exist.
SCA382	The getMounts operation shall return a MountSequence that contains the file systems mounted within the file manager.	OE	3.1.3. 5.1.3. 5.3.4	To provide a standard mechanism for retrieving a list of mounted file systems.	The list of file systems mounted by the FileManager is returned.
SCA383	The query operation shall return the combined mounted file systems information to the calling client based upon the given input fileSystemProperties' ID elements.	OE	3.1.3. 5.1.3. 5.4.3	To provide a standard mechanism for obtaining federated file system properties	Verified by conformance to SCA441.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA441	As a minimum, the query operation shall support the following input fileSystemProperties ID elements: 1. SIZE - a property item ID value of "SIZE" causes the query operation to return the combined total size of all the mounted file system as an unsigned long long property value; 2. AVAILABLE_SPACE - a property item ID value of "AVAILABLE_SPACE" causes the query operation to return the combined total available space (in octets) of all the mounted file system as unsigned long long property value.	OE	3.1.3. 5.1.3. 5.4.3	To ensure that the FileManager query operation returns the value for each of the recognized properties that are requested.	1. When "SIZE" is requested, the FileManager component returns the combined total number of octets allocated to all mounted file systems as an unsigned long long type. 2. When "AVAILABLE_SPACE" is requested, the FileManager component returns the combined total number of unused octets for all mounted file systems as an unsigned long long type.
SCA384	The query operation shall raise the UnknownFileSystemProperties exception when the input fileSystemProperties parameter contains an invalid property ID element.	OE	3.1.3. 5.1.3. 5.4.5	To ensure the query operation raises an exception when the input fileSystemProperties parameter contains an invalid property ID element.	The query operation raised the UnknownFileSystemProperties exception when the input fileSystemProperties parameter contained an invalid property ID element.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA397	A FileComponent's filePointer attribute shall be set to the beginning of the file when a FileComponent is opened for read only or created for the first time.	OE	3.1.3.5.2.1.3	To ensure the filePointer position is at the beginning of the file when a new file is created or an existing file is opened as read only.	1. An existing file is open for readonly or a new file is created. 2. The filePointer points to the beginning of the file.
SCA398	A FileComponent's filePointer attribute shall be set at the end of the file when a FileComponent already exists and is opened for write.	OE	3.1.3.5.2.1.3	To ensure the filePointer position is at the end of the file when an existing file is opened as read-write.	1. The file is opened as read-write. 2. The filePointer points to the end of the file.
SCA399	A FileComponent shall realize the File interface.	OE	3.1.3.5.2.1.4	To ensure a FileComponent provides a standard mechanism to read and write its file contents.	The FileComponent inherits the File interface.
SCA400	Valid characters for a FileSystemComponent file name and file absolute pathname shall adhere to POSIX compliant file naming conventions.	OE	3.1.3.5.2.2.3	To provide a standard naming convention for file name and absolute pathname.	FileSystemComponent accepts filenames and file absolute pathname that adhere to POSIX naming conventions.
SCA401	A FileSystemComponent shall realize the FileSystem interface.	OE	3.1.3.5.2.2.4	To ensure a FileSystemComponent provides a standard mechanism to manage access to a file system.	The FileSystemComponent inherits the FileSystem interface.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA402	Valid individual filenames and directory names for a FileSystemComponent shall be 40 characters or less.	OE	3.1.3.5.2.2.4	Establishes upper limit for filename and directory name length to be used by a FileSystemComponent.	All FileSystemComponent's operations accept both filenames and directory names each with lengths up to and including 40 characters.
SCA403	A valid pathname for a FileSystemComponent shall not exceed 1024 characters.	OE	3.1.3.5.2.2.4	Establishes upper limit for pathname (sequence of directory names and filename) length to be used by a FileSystemComponent.	All FileSystemComponent's operations accept pathnames with length up to and including 1024 characters.
SCA404	The FileSystem operations realized by a FileManagerComponent shall remove the name of the mounted file system from input pathnames before passing the pathnames to any operation on a mounted file system.	OE	3.1.3.5.2.3.3	To remove the portion of the directory path representing the mount point implemented by the FileManager.	Verified by file system operations.
SCA405	A FileManagerComponent shall propagate exceptions raised by a mounted file system.	OE	3.1.3.5.2.3.3	To ensure propagation of exceptions from a mounted file system to the invoking component.	Exceptions received from mounted file systems are propagated to the invoking component.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA406	A FileManagerComponent shall use the FileSystem operations of the FileSystemComponent whose associated mount point exactly matches the input fileName parameter to the lowest matching subdirectory.	OE	3.1.3.5.2.3.3	To ensure a FileManagerComponent uses the proper FileSystemComponent associated with a mount point.	The FileManagerComponent uses the FileSystemComponent that matches the most appropriate (longest) mount point portion of the input directory path.
SCA408	A FileManagerComponent shall realize the FileManager interface.	OE	3.1.3.5.2.3.4	To ensure a FileManagerComponent provides a standard mechanism to manage file systems.	The FileManagerComponent inherits the FileManager interface.
SCA409	A FileManagerComponent instantiation shall fulfill the FileSystemComponent component requirements.	OE	3.1.3.5.2.3.4	To ensure a FileManagerComponent satisfies the requirements of a FileSystemComponent.	Will be satisfied by the verification of FileSystemComponent requirements.
SCA298	A BasePlatformComponent shall register with its DeviceManagerComponent via the ComponentRegistry::registerComponent operation.	OE	3.1.3.5.2.4.3	Ensures a platform component registers with its DeviceManager.	BasePlatformComponent calls the provided ComponentRegistry::registerComponent operation.
SCA565	A BasePlatformComponent shall fulfill the BaseComponent requirements.	OE	3.1.3.5.2.4.4	To ensure a BasePlatformComponent satisfies the requirements of a BaseComponent.	Will be satisfied by the verification of BaseComponent requirements.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA416	The PlatformComponentFactoryComponent shall only deploy DeviceComponents or ServiceComponents.	OE	3.1.3.5.2.5.4	To restrict the deployment capabilities for the PlatformComponentFactoryComponent.	A component created by a PlatformComponentFactoryComponent is a DeviceComponent or ServiceComponent.
SCA527	A PlatformComponentFactoryComponent instantiation shall fulfill the BaseFactoryComponent requirements.	OE	3.1.3.5.2.5.4	To ensure a PlatformComponentFactoryComponent satisfies the requirements of a BaseFactoryComponent.	Will be satisfied by the verification of BaseFactoryComponent requirements.
SCA567	A PlatformComponentFactoryComponent instantiation shall fulfill the BasePlatformComponent requirements.	OE	3.1.3.5.2.5.4	To ensure a PlatformComponentFactoryComponent satisfies the requirements of a BasePlatformComponent.	Will be satisfied by the verification of BasePlatformComponent requirements.
SCA314	All ServiceComponents started up by a DeviceManagerComponent shall have a handler registered for the POSIX SIGQUIT signal.	OE	3.1.3.5.2.6.3	Provides a standard mechanism for handling termination of the ServiceComponents not initiated by the releaseObject operation.	ServiceComponents started by DeviceManagerComponents have registered a POSIX SIGQUIT signal handler.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA317	The values associated with the parameters (SERVICE_NAME) as described in 3.1.3.3.2.4.3 shall be used to set the platform service's ComponentIdentifier interface identifier attribute.	OE	3.1.3.5.2.6.3	Sets the ServiceComponent's identifier to the specified value of the SERVICE_NAME execute parameter.	The ServiceComponent's identifier is equal to the value associated with the SERVICE_NAME parameter.
SCA460	Each ServiceComponent shall have an SPD as described in section 3.1.3.6.4.	OE	3.1.3.5.2.6.4	To ensure a ServiceComponent has its required DomainProfile files.	The ServiceComponent has an associated SPD and all of its referenced DomainProfile files.
SCA568	A ServiceComponent shall fulfill the BasePlatformComponent requirements.	OE	3.1.3.5.2.6.4	To ensure a ServiceComponent satisfies the requirements of a BasePlatformComponent.	Will be satisfied by the verification of BasePlatformComponent requirements.
SCA530	A ManageableServiceComponent shall fulfill the ServiceComponent requirements.	OE	3.1.3.5.2.7.4	To ensure a ManageableServiceComponent satisfies the requirements of a ServiceComponent.	Will be satisfied by the verification of ServiceComponent requirements.
SCA569	A ManageableServiceComponent shall realize the LifeCycle interface.	OE	3.1.3.5.2.7.4	To ensure a ManageableServiceComponent provides a standard mechanism to manage its life cycle.	The ManageableServiceComponent inherits the LifeCycle interface.
SCA463	Domain Profile files shall be compliant to the descriptor files provided in Appendix D.	Both	3.1.3.6	Ensures Domain Profile files are compliant with Appendix D.	Each domain profile file is conformant to the appropriate SCA Appendix D DTD.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA471	Any Ada application shall be restricted to using the equivalent Ada functionality, as defined in POSIX Ada language binding [2], designated as mandatory by the target profile or may use the C interface defined in [3].	AP	B.6	To ensure an Ada application does not require functionality beyond the target profile.	An Ada application does not use any functionality not included in the target AEP.
SCA473	The options, limits, and any other constraints on POSIX.1 [1] shall be provided as described in Table 1.	OE	B.6.1	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA537	The functions listed in Table 2 shall behave as described in the applicable clauses of the referenced POSIX [4].	OE	B.6.1.1	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA484	The functions listed in Table 11 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.10	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA485	The functions listed in Table 12 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.11	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA486	The functions listed in Table 13 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.12	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA487	The functions listed in Table 14 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.13	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA488	The function listed in Table 15 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.14	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA489	The function listed in Table 16 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.15	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA465	The functions listed in Table 17 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.16	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA466	The functions listed in Table 18 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.17	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA467	The functions listed in Table 19 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.18	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA468	The function listed in Table 20 shall behave as described in the referenced clause.	OE	B.6.1.19	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA475	The functions in Table 3 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.2	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA469	The function listed in Table 21 shall behave as described in the referenced clause.	OE	B.6.1.20	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA476	The functions listed in Table 4 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.3	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA477	The functions listed in Table 5 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.4	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA478	The functions listed in Table 6 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.5	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA480	The functions listed in Table 7 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.6	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA481	The functions listed in Table 8 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.7	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA482	The functions listed in Table 9 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.8	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA483	The functions listed in Table 10 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.1.9	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA490	The functions listed in Table 22 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.2.1	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA491	The functions listed in Table 23 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.2.2	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA464	The functions listed in Table 24 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.2.3	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA492	The Standard C [3] Library header files listed in Table 25 shall be included within the AEP as described in the referenced clause.	OE	B.6.3	To specify a standard set of C header files available in each AEP.	The files listed in Table 25 (as a minimum) are present as required for the appropriate AEP.
SCA493	The functions listed in Table 26 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.4.1	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.
SCA470	The functions listed in Table 27 shall behave as described in the applicable clauses of POSIX [4].	OE	B.6.4.2	Provide a standard set of operating environment functions.	The mandatory POSIX functionality for the selected profile is provided by the OE.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA501	DTD files are installed in the domain and shall have ".dtd" as their filename extension.	Both	D-1.1	To ensure DTD files have a standard filename extension and are accessible by the Core Framework.	1. The SCA DTD files are accessible to the CF file system. 2. The file's extension is ".dtd".
SCA502	All XML files shall have as the first two lines as an XML declaration (?xml) and a document type declaration (!DOCTYPE).	Both	D-1.1	To ensure a standard location of XML declaration and document type declaration in a XML file.	1. An XML declaration, ?xml, is on 1st line of an XML file. 2. A document type declaration, !DOCTYPE, is on 2nd line of an XML file.
SCA496	A Software Assembly Descriptor file shall have a ".sad.xml" extension.	AP	D-1.10	To ensure a standard file extension for a Software Assembly Descriptor file.	1. The file content implements a Software Assembly Descriptor. 2. The file's extension is ".sad.xml".
SCA497	A Device Configuration Descriptor file shall have a ".dcd.xml" extension.	OE	D-1.11	To ensure a standard file extension for a Device Configuration Descriptor file.	1. The file content implements the Device Configuration Descriptor syntax. 2. The file's extension is ".dcd.xml".
SCA498	A DomainManager Configuration Descriptor file shall have a ".dmd.xml" extension.	OE	D-1.12	To ensure a standard file extension for a DomainManager Configuration Descriptor file.	1. The file content implements the DomainManager Configuration Descriptor syntax. 2. The file's extension is ".dmd.xml".
SCA499	A Platform Deployment Descriptor file shall have a ".pdd.xml" extension.	OE	D-1.13	To ensure a standard file extension for a Platform Deployment Descriptor file.	1. The file content implements the Platform Deployment Descriptor syntax. 2. The file's extension is ".pdd.xml".

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA500	An Application Deployment Descriptor file shall have an ".add.xml" extension.	AP	D-1.14	To ensure a standard file extension for an Application Deployment Descriptor file.	1. The file content implements an Application Deployment Descriptor. 2. The file's extension is ".add.xml".
SCA503	A Software Package Descriptor file shall have a ".spd.xml" extension.	Both	D-1.6	To ensure a standard file extension for a Properties Descriptor file.	1. The file content implements a Software Package Descriptor. 2. The file's extension is ".spd.xml".
SCA504	A Device Package Descriptor File shall have a ".dpd.xml" extension.	OE	D-1.7	To ensure a standard file extension for a Device Package Descriptor file.	1. The file content implements the Device Package Descriptor syntax. 2. The file's extension is ".dpd.xml".
SCA494	A Properties Descriptor shall have a ".prf.xml" extension.	Both	D-1.8	To ensure a standard file extension for a Properties Descriptor file.	1. The file content implements a Properties Descriptor. 2. The file's extension is ".prf.xml".
SCA495	A Software Component Descriptor file shall have a ".scd.xml" extension.	Both	D-1.9	To ensure a standard file extension for a Software Component Descriptor file.	1. The file content implements a Software Component Descriptor. 2. The file's extension is ".scd.xml".
SCA505	The OE shall provide the features designated as mandatory, as specified in E-2.7, for the implemented SCA CORBA profile.	OE	E-2.6.1	Ensures the operating environment provides the standard set of features specified by the CORBA profile.	The mandatory CORBA functionality for the selected profile is provided by the OE.
SCA506	Applications shall be limited to using the features designated as mandatory, as specified in E-2.7, for the implemented SCA CORBA profile.	AP	E-2.6.2	Ensures the Application uses only the standard set of features specified by the CORBA profile.	The Application does not use more than the mandatory features of the selected SCA CORBA profile.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA507	The features included in the Full, LW and ULW Profiles listed in Attachment 1 to this appendix shall behave as described in the applicable clauses of CORBA/e [1].	OE	E-2.7.1	To ensure the behaviors of the features in the standard set of CORBA/e profiles.	The features within the Full, LW, and ULW behave as described in the applicable part of CORBA/e.
SCA508	The features included in the Full, LW and ULW Profiles listed in Attachment 2 to this appendix shall behave as described in the applicable clauses of RT CORBA [2].	OE	E-2.7.2	To ensure the behaviors of the features in the standard set of RT CORBA profiles.	The features within the Full, LW, and ULW behave as described in the applicable part of RT CORBA.
SCA509	The Full and LW Profiles shall support the additional standardized parameters identified in Table 1 to the ORB_init call to allow the root POA to be created with non-default policies.	OE	E-2.7.2.1	To ensure a standard mechanism for Full and LW Profiles to create a root POA with non-default policies.	1. Full or LW Profile is used. 2. root POA created as specified with the non-default policies.
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.	OE	F.7.1.1	Ensures the operating environment provides the SCA specified sub-set of Naming Service features as defined in the OMG Naming Service Specification.	The operating environment provides the SCA specified sub-set of Naming Service features as defined in the OMG Naming Service Specification.
SCA69*	N/A.	OE	F.7.1.3	N/A	N/A
SCA70*	N/A.	OE	F.7.1.3	N/A	N/A

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
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.	OE	F.7.1.3	Provide a standard sequence of operations for initializing, connecting and configuring application components.	1. The create operation calls the initialize operation on each of an application's components. 2. The create operation establishes the connections for each application component. 3. The create operation calls the configure operation on the assembly controller.
SCA552	The installApplication operation shall raise the ApplicationInstallationError exception when SCA V2.2.2 application installation is not supported.	OE	F.7.1.4	To ensure an exception is raised when SCA V2.2.2 application installation is not supported by the platform.	The installApplication operation raised the ApplicationInstallationError exception when an attempt was made to install an SCA V2.2.2 application which is not supported by the platform.
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.	OE	F.7.1.5	Ensure the CF ApplicationManager removes each application component from the Naming Service.	1. Any Application component that is being released which has been registered with the naming service has been removed from the naming service. 2. The naming context bound to the Application component is deleted.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
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.	OE	F.7.1.5	To ensure the ApplicationManager forwards the runTest, start, stop, configure, and query operations to the assemblycontroller. Note: this is a backwards compatibility requirement only.	1. An external entity (e.g. HMI, nonCore Framework component) calls the ApplicationManagerComponent runTest, start, stop, configure, and query operations. 2. ApplicationManagerComponent delegates the corresponding runTest, start, stop, configure, and query operations to the assembly controller.
SCA159 *	The ApplicationManagerComponent shall propagate exceptions raised by the AssemblyComponent's assembly controller.	OE	F.7.1.5	To ensure the ApplicationManagerComponent propagates exceptions raised by an assembly controller.	1. An external entity (e.g. HMI, non Core Framework component) calls the ApplicationManagerComponent runTest, start, stop, configure, and query operations. 2. The ApplicationManagerComponent propagates exceptions raised by runTest, start, stop, configure, and query operations of the application's assembly controller.
SCA160 *	N/A.	OE	F.7.1.5	N/A	N/A
SCA161 *	N/A.	OE	F.7.1.5	N/A	N/A
SCA162 *	N/A.	OE	F.7.1.5	N/A	N/A

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA555	The create operation shall instantiate a SCA V2.2.2 compliant application if the SAD does not have a sca_version element.	OE	F.7.1.6	To create and instantiate SCA V2.2.2 applications to maintain backward compatibility.	1. The SAD does not contain a sca_version element. 2. The create operation instantiates an SCA v2.2.2 application.
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.	OE	F.7.1.7	To ensure the create operation creates required naming contexts for the application. Note: this is a backwards compatibility requirement only.	Verified by SCA76*
SCA557	Upon execution of a software module by the create operation, a Resource or a ResourceFactory component shall register with the Naming Service.	OE	F.7.1.7	Ensures a created Resource or ResourceFactory registers with the Naming Service.	The created component is registered with the Naming Service.
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.	OE	F.7.1.7	Constrains the ApplicationFactory to a candidate set of devices which satisfy the internal allocation property requirements. Note: this is a backwards compatibility requirement only.	1. The application component has properties with a kindtype of "allocation" and an action element with the value NOT "external". 2. The ApplicationFactory's candidate set of devices satisfies step 1 and the requested allocation value.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
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.	OE	F.7.1.7	To ensure the create operation allocates capacity from the candidate set of devices (if applicable). Note: this is a backwards compatibility requirement only.	1. The application component has properties with a kindtype of "allocation" and an action element with the value "external". 2. Resources are allocated on the devices that support the allocation property.
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.	OE	F.7.1.7	To ensure the ApplicationFactory releases allocated capacities when an application is not created successfully. Note: this is a backwards compatibility requirement only.	1. The application is not created successfully. 2. Device capacity is deallocated for those Devices where allocation was successful for the application being created.
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.	OE	F.7.1.7	To ensure a standard mechanism for determining when application modules should be loaded onto Loadable or Executable devices. Note: this is a backwards compatibility requirement only.	1. The Loadable or Executable Device satisfies the application module's allocation requirements. 2. The application modules are loaded.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
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.	OE	F.7.1.7	Ensures the create operation provides the mandatory execute parameters to the component being executed.	1. The component SAD contains a findcomponent element with a namingservice sub-element. 2. The create operation used the mandatory execute parameters (Naming Context IOR, Name Binding and Component Identifier) when calling execute.
SCA77*	N/A.	OE	F.7.1.7	N/A	N/A
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.	OE	F.7.1.7	Provide a standard mechanism for supplying parameters to resources when using a ResourceFactory.	1. Identify the componentinstantiation componentresourcefactoryref element properties whose kindtype element is factoryparam as specified in the domain profile. 2. The create operation used the "factoryparam" parameters when calling the CF::ResourceFactory::createResource operation.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
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.	OE	F.7.1.7	Provide a standard mechanism for supplying execute parameters to an application component.	1. Identify "execparam" parameters as specified in its domain profile. 2. The CF::ApplicationFactory::create used the "execparam" parameters when calling CF::ExecutableDevice::execute.
SCA85*	The create operation shall establish connections for an application which are specified in the SAD domainfinder element.	OE	F.7.1.7	To ensure that the ApplicationFactory establishes application connections specified in the SAD domainfinder element.	The ApplicationFactory established the application connections specified in the SAD domainfinder element.
SCA86*	N/A.	OE	F.7.1.7	N/A	N/A
SCA87*	N/A.	OE	F.7.1.7	N/A	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".	OE	F.7.1.7	To ensure each application's assemblycontroller is configured with properties that are kindtype of "configure" and a mode of "readwrite" or "writeonly" specified in the Domain Profile.	1. Identify "configure" parameters as specified in its domain profile. 2. The CF::ApplicationFactory::create used the configure parameters when calling configure on the assemblycontroller.

SCA 4.1 Req #	SCA 4.1 Requirement Text	SCA 4.1 Requirement Allocation	Doc Sec	Requirement Objective	Requirement Verification Criteria
SCA98*	N/A.	OE	F.7.1.7	N/A	N/A
SCA542*	N/A.	OE	F.7.1.7	N/A	N/A
SCA524*	N/A.	OE	F.7.1.7	N/A	N/A
SCA558	The installApplication operation shall install a SCA V2.2.2 [4] compliant application.	OE	F.7.1.8	Support installation of SCA v2.2.2 applications.	1. SAD does not contain a sca_version element. 2. The CF installed the v2.2.2 application.

5 References

1. Software Communications Architecture Specification, Version 4.1, 20 August 2015

2. Software Communications Architecture Specification Appendix B: SCA Application Environment Profiles, Version 4.1, 20 August 2015

3. Software Communications Architecture Specification Appendix C: Core Framework Interface Definition Language, Version 4.1, 20 August 2015

4. Software Communications Architecture Specification Appendix D: Platform Specific Model – Domain Profile Descriptor Files, Version 4.1, 20 August 2015

5. Software Communications Architecture Specification Appendix E: Platform Specific Model (PSM) – Transfer Mechanisms and Enabling Technologies, Version 4.1, 20 August 2015

6. Software Communications Architecture Specification Appendix F: Units of Functionality and Profiles, Version 4.1, 20 August 2015