SCA 4.1 Deployment Data

Document WINNF-15-R-0014
Version V1.0.0
23 April 2015
This document has been prepared by the SCA 4.1 Draft Issue Adjudication Task Group 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 It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the SCA 4.1 Draft Issue Adjudication Task Group.

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.
Proposal

This document contains a proposal to change the Draft SCAv4.1 specification for the domain management interfaces to utilize the ComponentType specializedInfo for component deployment information. SpecializedInfo was introduced in SCAv4.1.

Proposal author:
• Hugues Latour, Communication Research Centre Canada (CRC)
• Jerry Bickle, Raytheon

Proposal contributors/reviewers:
• Chris Hagen, Rockwell Collins
• Chuck Linn, Harris
• Francois Levesque, NordiaSoft
• Kevin Richardson, MITRE
• Sarah Miller, Rockwell Collins
Recommendation

SCA v4.1 Application Deployment Data
Topics

Description of the Issue
Summary of the Proposal
Detailed Proposal

Specifications Changes (found in a word document)
• Main Specification Changes
• IDL Specification Changes
• Appendix C, D and F Changes
Description of the Issue

SCA 4 introduce the push approach to avoid callbacks. This is not the case for component deployment data. Have to implement ApplicationDeploymentAttributes interface to get application deployment information.

Have to make multiple calls to get deployment information.

• The component deployment information is broken apart in separate interface readonly attributes

Also SCA 4 lost the capability of providing deployment information for platform components due to DeviceManager interface change.
Rationale

1. Able to get Component Deployment Data without multiple callbacks
2. Able to get Component Deployment Data without implementing ApplicationDeploymentAttributes interface. Push Only Approach
3. Able to get Component Deployment Data for Platform Components
4. Uniformity and Consistent use of ComponentType
5. Rename ApplicationDeploymentAttributes as DeploymentAttributes used for DeviceManagerComponent for late registration.
   1. Name change is due to scope being different
6. Removed DeviceManagerAttributes since now covered by DeploymentAttributes, and filesys and profile attributes are not needed since in ComponentType.
Summary of the Proposal

1. SCA 4.1 added specializedInfo field to ComponentType.
2. Define a Deployment Info IDs and types (as necessary) that captures a component’s deployment information that can be used with a ComponentType’s specializedInfo field.
   • This approach is scalable and easier to add changes instead of DeploymentInfoType that is all encompassing
3. Tie the Deployment Info to AppDeploymentData and DeviceMgrDeploymentData UOFs
4. Modify and rename ApplicationDeploymentAttributes to DeploymentAttributes
5. Remove DeviceManagerAttributes.
6. Expand the component Deployment Data UOF capability to DeviceManagerComponent
7. Use consistent terminology (deploy, create, launch) - Use Deploy terminology, replace launch and create with deploy. Deploy and deployment terminology is used in various places in specification already. This is also needed to better support lw profile where there is no registration.
8. Update Appendix D and F for DeviceMgrDeploymentData UOF
9. Removed Profile Name parameter for Device Component deployment since profile attribute was removed.
3.1.3.1.3.28 ManagerInfo

The ManagerInfo structure defines the supplemental information that is used as the ComponentType’s specializedInfo for a manager component. The fileSys field is the file system used by this manager component. The registereddeployedComponents field is a sequence of components that have registered with been deployed by this manager component. The associated ID for ManagerInfo type is MANAGER_INFO_ID. For each deployed component, its ComponentType ‘s identifier, componentObject, profile, providesPorts, and type fields are as specified in 3.1.3.1.3.17 ComponentType.

Add New Requirement - SCA XXX For each deployed BaseComponent in deployedComponents, the deployed component specializedInfo shall contain Deployment information. The deployment information consists of:

1. A DataType with an id of PROCESS_ID and value of ExecutableInterface::ProcessID_Type.
2. A DataType with an id of IMPLEMENTATION_ID and value of the profile’s implementation id used for deployment.
3. A DataType with an id of TARGET_DEVICE_ID and value of the DeviceComponent’s identifier that component is deployed on.
4. An DataType with an id of USES_DEVICE_ID and a value of CF::UsesDeviceAssignmentSequence type when a component has uses devices as indicated by its profile.

• Tie New Requirement to Device Mgr DeploymentData UOF
3.1.3.1.3.28 ManagerInfo

struct ManagerInfo
{
    CF::FileSystem fileSys;
    CF::Components registereddeployedComponents;
}

Update Appendix C and SpecializedInfo IDL file.
Update 3.1.3.1.3.29 SpecializedInfo Identifiers

Add PROCESS_ID, IMPLEMENTATION_ID, TARGET_DEVICE_ID, USES_DEVICES_ID, COMPONENTS_ID

This string constant is the identifier for ExecutableInterface::ProcessID_Type value within a ComponentType’s specializedInfo.

```cpp
const string PROCESS_ID = "PROCESS_ID";
```

This string constant is the identifier for SPD implementation id string value within a ComponentType’s specializedInfo, which is the implementation used for the creation of the component.

```cpp
const string IMPLEMENTATION_ID = "IMPLEMENTATION_ID";
```

This string constant is the identifier for the device identifier string value within a ComponentType’ specializedInfo field, which is the device that deployed the component.

```cpp
const string TARGET_DEVICE_ID = "TARGET_DEVICE";
```
Update 3.1.3.1.3.29 SpecializedInfo Identifiers

Add `PROCESS_ID`, `IMPLEMENTATION_ID`, `TARGET_DEVICE_ID`, `USES_DEVICES_ID`, `COMPONENTS_ID`

This string constant is the identifier for the CF::UsesDeviceAssignmentSequence value within a ComponentType’s specializedInfo, which denotes the devices used by component.

```cpp
const string USESDEVICE_ID = "USES_DEVICE";
```

This string constant is the identifier for the CF::Components type value within a ComponentType’s specializedInfo field.

```cpp
const string COMPONENTS_ID = "COMPONENTS";
```

Update Appendix C and SpecializedInfo IDL file.
3.1.3.1.3.30 UsesDeviceAssignmentType

This structure associates a component’s profile uses device identifier with the assigned device identifier of the device being used.

```csharp
struct UsesDeviceAssignmentType
{
    string usesDeviceId;
    string assignedDeviceId;
};
```
3.1.3.1.3.30 UsesDeviceAssignmentSeq

The sequence provides an unbounded sequence of 0..n of UseDeviceAssignmentType. The UsesDeviceAssignmentSeq is a BaseComponent supplemental deployment information for a ComponentType’s specializedInfo. The associated ID for UsesDeviceAssignmentSeq type is USES_DEVICE_ID.

typedef sequence <UsesDeviceAssignmentType> UsesDeviceAssignmentSeq;

Norte: DeviceAssignment type is similar to this but has componentId instead of usesDeviceId.

**Update Appendix C and SpecializedInfo IDL file.**
3.1.3.3.1.2 ApplicationDeploymentAttributes

1. Rename ApplicationDeploymentAttributes interface to DeploymentAttributes.

2. Remove Interface types
   - 3.1.3.3.1.2.3.1 ComponentProcessIdType, 3.1.3.3.1.2.3.2 ComponentProcessIdSequence
   - 3.1.3.3.1.2.3.3 ComponentElementType, 3.1.3.3.1.2.3.4 ComponentElementSequence

3. Remove readonly attributes
   - 3.1.3.3.1.2.4.1 componentProcessIds, 3.1.3.3.1.2.4.2 componentDevices
   - 3.1.3.3.1.2.4.3 componentImplementations,

4. Modify 3.1.3.3.1.2.4.4 registereddeployedComponents

   SCA64 The registereddeployedComponents attribute shall return the list of ApplicationDeployed BaseComponents that have registered been successfully deployed or a sequence length of zero if no ApplicationBaseComponents have been deployed registered. For each deployed component, its ComponentType's identifier, componentObject, profile, providesPorts, and type fields are as specified in 3.1.3.1.3.17 ComponentType.

5. Update Figures and text for name change

6. Update Appendix C and rename CFApplicationDeploymentAttributes IDL file to CFDeploymentAttributes.idl
3.1.3.3.1.2 Application Deployment Attributes

- Components

Add New Requirement - SCA XXX For each deployed BaseComponent in deployedComponents, the deployed component specializedInfo shall contain Deployment information. The deployment information consists of:

1. A DataType with an id of PROCESS_ID and value of ExecutableInterface::ProcessID_Type.
2. A DataType with an id of IMPLEMENTATION_ID and value of the profile’s implementation id used for deployment.
3. A DataType with an id of TARGET_DEVICE_ID and value of the DeviceComponent’s identifier that component is deployed on.
4. An DataType with an id of USES_DEVICE_ID and a value of CF::UsesDeviceAssignmentSequence type when a component has uses devices as indicated by its profile.

- Tie New Requirement to DeviceMgr DeploymentData UOF
3.1.3.3.2 Components

The DeviceManagerComponent is responsible for creation deployment of DeviceComponents and launching ServiceComponents.
3.1.3.2.2.2.3 ManagebaleApplicationComponent Semantics

Change From

SCA82 A ManageableApplicationComponent created via an ExecutableDeviceComponent shall register with its launching ApplicationFactoryComponent via the ComponentRegistry::registerComponent operation.

To

SCA82 When COMPONENT_REGISTRY_IOR parameter is supplied, a ManageableApplicationComponent shall register via the ComponentRegistry::registerComponent operation.
3.1.3.3.1.3.5.1.3 ApplicationFactory Behavior

SCA74 The create operation shall **execute** deploy the application software modules **ApplicationComponents** as specified in the SAD.

3.1.3.3.2.2.3 ApplicationFactoryComponent Semantics

SCA76 When the create operation creates 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.

SCA542 When the create operation creates 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.

SCA77 When the create operation creates deploys an ApplicationComponent via an ApplicationComponentFactoryComponent, it shall provide the Component Identifier parameter as defined in this section.

SCA81 When an ApplicationComponent is created--deployed via an ExecutableDeviceComponent, 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.
Detailed Proposal

Remove 3.1.3.3.1.6 DeviceManagerAttributes
Update Figures and text where ever DeviceManagerAttributes is used.
Update Appendix C and remove CFDeviceManagerAttributes.idl file.
3.1.3.3.2.1.4 ApplicationManagerComponent Constraints

Remove SCA553 Requirement, no longer needed

SCA553 An ApplicationManagerComponent shall realize the ApplicationDeploymentAttributes interface.
3.1.3.3.2.2.3 ApplicationFactoryComponent Semantics

Remove SCA524 requirement since new requirement is now at return ComponentType instead of on DeploymentAttributes interface.

SCA524 The create operation shall add the ManageableApplicationComponent(s) launched by an ApplicationComponentFactoryComponent to the ApplicationManager registeredComponents attribute of the ApplicationFactoryComponent.
3.1.3.3.1.3.5.1.4 Application Factory return

- **Add Requirements** –
- SCA XXX The returned ComponentType’s specializedInfo shall contain application deployed components as identified by COMPONENTS_ID and CF::Components type value. For each deployed component, its ComponentType’s identifier, componentObject, profile, providesPorts, and type fields are as specified in 3.1.3.1.3.17 ComponentType.

- Tie req to Interrogable UOF
3.1.3.3.1.3.5.1.4 Application Factory return

- **Add Requirements** –
- SCA XXX Each application deployed component’s specializedInfo shall contain Deployment Information. (Tie req to App Deployment Data UOF)
- The deployment information consists of:
  1. A DataType with an id of PROCESS_ID and value of ExecutableInterface::ProcessID_Type.
  2. A DataType with an id of IMPLEMENTATION_ID and value of the profile’s implementation id used for deployment.
  3. A DataType with an id of TARGETDEVICE_ID and value of the DeviceComponent’s identifier that the component is deployed on.
  4. An DataType with an id of USESDEVICE_ID and a value of CF::UsesDeviceAssignmentSequence type when a component has uses devices as indicated by its profile.
3.1.3.3.2.4.3 DeviceManagerComponent Semantics

**Modify Statements** - The ComponentType’s type field of the registering DeviceManagerComponent is a DEVICE_MANAGER_COMPONENT. The registering DeviceManagerComponent ComponentType’s specializedInfo field contains an ID of MANAGER_INFO_ID and a value of ManagerInfo structure data type that holds the BasePlatformComponents that have registered with the DeviceManagerComponent.

**Remove statement** - The specializedInfo field may contain a sequence of AllocationPropertyType structures that holds the BasePlatformComponent’s allocation properties.

The DeviceManagerComponent can launch deploy DeviceComponents, PlatformComponentFactoryComponents and ServiceComponents directly (e.g. thread, posix_spawn), or by using an ExecutableDeviceComponent, or by PlatformComponentFactoryComponent. These components register with the launching deployed by DeviceManagerComponent may register via the ComponentRegistry::registerComponent operation.
3.1.3.3.2.4.3 DeviceManagerComponent Semantics

Make SCA219 and SCA221 are not a requirement since covered by SCA64 and SCA131. To be consistent with DomainManagerComponent.

SCA131 The `registerComponent` operation shall register the component indicated by the input `registeringComponent` parameter, if it does not already exist.

SCA219 Upon successful `registration` `BasePlatformComponent` deployment via the `ComponentRegistry` interface, the `DeviceManagerComponent` shall add the deployed components to its `registeredDeployedComponents` attribute.

Remove SCA221

SCA221 The `DeviceManagerComponent` shall add the `DeviceComponent` and `ServiceComponent` components launched by a `PlatformComponentFactoryComponent` to the `registeredComponents` attribute of the `DeviceManagerComponent`.

Add Requirement: SCAXXX When a `PlatformComponentFactoryComponent` unregisters with the `DeviceManagerComponent`, the `DeviceManagerComponent` shall unregister the `PlatformComponentFactoryComponent’s` `BasePlatformComponents`.
3.1.3.3.2.4.3 DeviceManagerComponent Semantics

SCA218 If multiple FileSystemComponents are to be created, the DeviceManagerComponent shall mount created FileSystemComponents to a FileManagerComponent (widened to a FileSystemComponent through the ManagerInfo’s FileSys field attribute).

SCA442 When a DeviceComponent is launched directly (e.g. thread, posix_spawn) or by using an ExecutableDeviceComponent, 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. Profile Name - The ID is "PROFILE_NAME" and the value is a string that is the full mounted file system file path name;

SCA438 When a DeviceComponent is created deployed via PlatformComponentFactoryComponent, the DeviceManagerComponent shall supply the following properties as the qualifiers parameter to the referenced ComponentFactory::createComponent operation:

-1. Profile Name - The ID is "PROFILE_NAME" and the value is a string that is the full mounted file system file path name;
3.1.3.3.2.4.3 DeviceManagerComponent Semantics

SCA439 When a ServiceComponent is created deployed via a PlatformComponentFactoryComponent, the DeviceManagerComponent shall supply the following properties as the qualifiers parameter to the referenced PlatformComponentFactoryComponent's createComponent operation:

SCA227 The DeviceManagerComponent shall initialize registered deployed components that are instantiated by the DeviceManagerComponent provided they realize the LifeCycle interface. Registered components may also be obtained from a PlatformComponentFactoryComponent.

SCA228 After component initialization, the DeviceManagerComponent shall configure registered deployed components that are instantiated by the DeviceManagerComponent, provided they realize the PropertySet interface.
3.1.3.3.2.4.3 DeviceManagerComponent Semantics

Modify Statement: SCA230 The `registerComponent` operation of `DeviceManagerComponent` shall register a deployed component specified by the `registeringComponent` with the `DomainManagerComponent` domain manager when the `DeviceManagerComponent` device manager has previously already registered with `DomainManagerComponent` and the `registeringComponent` has been successfully added to the `DeviceManagerComponent`'s `registeredComponents` attribute.

Add Statements

For deployed `BasePlatformComponents` that register to the `DeviceManagerComponent`, their registration to `DomainManagerComponent` is done after they registered in to the `DeviceManagerComponent`.

The `ComponentType`'s `type` field of the registering component is a `BasePlatformComponent` type (i.e., `DEVICE_COMPONENT`, `LOADABLE_DEVICE_COMPONENT`, `EXECUTABLE_DEVICE_COMPONENT`, `SERVICE_COMPONENT`, and `MANAGEABLE_SERVICE_COMPONENT`, and `PLATFORM_COMPONENT_FACTORY_COMPONENT`). The `registeringComponent`'s `ComponentType` `specializedInfo` contains an id of `DEVICE_MANAGER_ID` and a value of `DeviceManagerComponent`'s identifier attribute.

Modify Statement: SCA233 The `unregisterComponent` operation shall unregister the registered component specified by the input identifier parameter from the `DomainManagerComponent` if it is registered with the device manager and the device manager is not shutting down. An unregistering component is removed from `DeviceManagerComponent`'s `deployedComponents` attribute.
3.1.3.3.2.4.4 DeviceManagerComponent Constraints

SCA561 A DeviceManagerComponent shall realize the DeviceManagerDeploymentAttributes interface.
Appendix D - D-1.11.1 deviceconfiguration

The corba_provider, log_capable, log_producer, and platformcomponentfactorydeployment, and devicemgr_deployment_data attributes represent the optional Units of Functionality (UOF) supported by the DeviceManagerComponent.

Update Figure 36: deviceconfiguration Element Relationships

```xml
<!ELEMENT deviceconfiguration
......
<!ATTLIST deviceconfiguration
id ID #REQUIRED
name CDATA #IMPLIED
corba_provider (false | true) "true"
log_capable (false | true) "false"
log_producer (false | true) "false"
 oe_profile (lightweight| medium|full) "medium"
 platformcomponentfactorydeployment (false | true) "true"
 devicemgr_deployment_data (false | true) "false"
>
```
Appendix F.6.1 Target Operating Environment Units of Functionality

AppDeploymentData - provides the application manager component with returning deployment information capability for deployed application components.

DeviceMgrDeploymentData - provides the device manager component with returning deployment information capability for deployed platform components.

Add DeviceMgrDeploymentData

Update Figure 4.