SOA™

software

SOLA
&
LIFECYCLE MANAGER
INTEGRATION GUIDE

Trademarks

Copyright

Table of Contents

OVERVIEW

The Smart Controls framework allows the Lifecycle Manager (LM) product to be tightly integrated with the SOA SOLA product allowing for end-to-end-governance of the service lifecycle. SOLA integration is facilitated primarily with listeners and related elements defined in the Library Configuration document. While this document does not describe the actual service flows and associated use cases, it does describe the integration points between the products and the associated library configuration elements.

CONFIGURING SOLA AS A FEDERATED SYSTEM

Connections to external registries such as SOLA are facilitated by specifying a specific <federated-system> element in the <federated-systems> section of the Library Configuration document. A <federated-system> element represents not only the connection to an external system but the identity of the system. Any data stored in Lifecycle Manager for that system will be scoped by the name of the federated-system. For this reason federated-system names must be unique and cannot be changed. The federated-system class for SOLA is SOLAFederatedSystem.

SOLA Federated System details

- **Purpose:**
  Represents an SOA SOLA installation.
- **Class:** com.logiclibrary.integrations.sola.SOLAFederatedSystem
- **Properties:**
  - *endpoint*
    The root URL to the SOLA installation.
    Example: "http://example.com:1454"
    (Required)
  - *user*[1]
    User for authentication with SOLA
    (Required)
  - *password*
    Password for authentication with SOLA
    (Required)
  - *validate-connection*
    Valid values are "true" or "false". If this value is true, then a test connection to SOLA will be made and any communication errors will

[1] The user specified should have privileges in SOLA to create and modify services.

prevent the LPC from uploading. If false, no such connection will be attempted. Default: “true”.
(Optional)

Example Configuration

```
<federated-systems>
  <federated-system name="SOLA"
class="com.logiclibrary.integrations.sola.SOLAFederatedSystem ">
    <properties>
       <property name="endpoint" value="http://example.com:1454 "/>
       <property name="user" value="user"/>
       <property name="password" value="password" encrypt="true"/>
     </properties>
  </federated-system>
</federated-systems>
```

SYNCHRONIZING GROUPS AND ORGANIZATIONS

SOLA integration will synchronize groups and organizations as needed to when publishing services. Therefore, there is not required step needed to maintain the organizations.

PUBLISHING SERVICES

The integration supports the publishing of Lifecycle Manager Assets representing web services[2] into SOLA. There are several different scenarios that SOLA supports (bottom-up, top-down, meet-in-the-middle), currently only bottom-up is supported in the integration, which will be expanded in the future.

Bottom-Up

In the bottom-up scenario, the service is defined in Lifecycle Manager with operations as classifiers. When the service is published to SOLA, a service will be created containing these operations. Lifecycle manager does not manage a WSDL in this scenario. Instead SOLA builds up the WSDL as the set of operations that are passed to SOLA as well as the analysis mapping stage in SOLA.

Project Creation

If a project does not exist in SOLA representing the owning group/project of the service asset in Lifecycle Manager, the project will be automatically created.

---

[2] Specifically, assets representing complete web services (single-asset representation) and those assets representing a service implementation (multi-asset representation). Service interface assets and associated schema assets are currently not explicitly published to Policy Manager.

SOLAPublisher Listener Details

- **Behavior:**
  Publishes and updates service assets into a specified SOLA installation. This listener will create the service in SOLA and populate information within the service based on the information in the asset.
- **Usage Context:**
  Generally configured to be triggered at Asset publish/republish time by the *ASSET_AUTO_PUBLISH*, *ASSET_MANUAL_PUBLISH*, *ASSET_AUTO_REPUBLISH*, and *ASSET_MANUAL_REPUBLISH* Events. Should be restricted with a Filter allowing only Assets that represent web services.
- **Class:** com.logiclibrary.integrations.sola.SOLAPublisher
- **Properties:**
  - The WSDL / schema resolution properties are supported for this importer. See Appendix Y of the Library Process Configuration guide for more details.
  - *federated-system-name*
    This is the name of the SOLA federated-system to publish to (Required)
  - *workflow-type*
    The workflow-type of the asset. Possible values are "bottom-up", "top-down", or "meet-in-the-middle". Currently only "bottom-up" is supported.
  - *operation-classifier*
    The classifier name containing the list of operations to create in the service. If not specified the "operation" classifier will be used.
  - *program-type-classifier*
    The classifier name containing the SOLA program type. If not specified, the "program-type" classifier is used. If program type classifier is found the program type used will be "LM".
  - *program-name-classifier*
    The name of the SOLA program-name. If not specified, the "program-name" classifier is used. If no program name is specified, one will be created and assigned to the service when it is created in SOLA.
- **Prerequisites:**
  It is highly recommended to use the SOLAValidator to validate service assets conform to the requirements needed by SOLA prior to publishing the service to SOLA.
- **Return Codes:**
  - *0 – success*

- *1 – Duplicate service error*

LIFECYCLE

Lifecycle Manager can be notified of state/lifecycle transitions of services within SOLA during the analysis stage. To enable this functionality a webservice API call is made from SOLA to Lifecycle Manager during appropriate points. The LPC (library process configuration) for the library can be made to take advantage of these notifications.

The SOLA_EVENT LPC event will be created for the SOLA. To take advantage of this requires some manual configuration of the LPC. In the global / top-level section of the LPC a new custom-event, filter, action, and listener need to be defined

This event will have a custom property associated with it that indicates the status of the service in SOLA. The property name is "state". Valid values will be "INITIAL", "UPDATED", and "ISREADY".

The custom-event notifies the LPC that the action needs to be take on the SOLA_EVENT

```
<custom-events>
  <custom-event>SOLA_EVENT</custom-event>
  …
</custom-events>
```

The filter lets the LPC take certain actions whenever an event is processed

```
<filters>
  <filter name="SOLAEventFilter">
     <event>SOLA_EVENT</event>
  </filter>
  …
</filters>
```

The action triggers a listener whenever the filter is fired (in this case the AdvanceStatus listener.

```
<actions>
  <action name="Handle SOLA Transition">
    <trigger-event>
      <event-filter>SOLAEventFilter</event-filter>
    </trigger-event>
    <listener>AdvanceStatus</listener>
  </action>
  …
</actions>
```

The listener can be anything, but most likely one that sets a status classifier or other lifecycle related behavior.

```
<listeners>
  <listener name="AdvanceStatus" class="SetAssetClassifier">
    <properties>
      <property name="classifier-name" value="sola-status" />
      <property name="classifier-value" value="SYNCHRONIZED" />
      <property name="update-semantics" value="replace" />
      <property name="submit-asset" value="true" />
```

```
        <property name="submit-note" value="Synchronize asset status
with SOLA" />
      </properties>
    </listener>
    …
  </listeners>
```

ARTIFACT SOURCE

A SOLAArtifactSource is defined to allow assets to reference SOLA service detail pages and WSDL documents.

SOLAArtifactSource Details

- **Behavior:**
  Used for displaying either service detail pages or a service's WSDL from SOLA. There are two general formats for URLs using this artifact source soa://*sola*/service and soa://*sola*/wsdl. Where *sola* is the name of the artifact source, and "service" or "wsdl" refers to the service detail page or service WSDL that is being referenced.
- **Usage Context:**
  Used on assets representing services.
- **Class:** com.logiclibrary.integrations.sola.SOLAArtifactSource
- **Properties:**
  - *federated-system-name*
    This is the name of the SOLA federated-system
    (Required)
- **Example:**

```
    <artifact-source name="sola"
class="com.logiclibrary.integrations.sola.SOLAArtifactSource">
      <properties>
        <property name="federated-system-name" value="SOLA" />
      </properties>
    </artifact-source>
```

VALIDATORS

Lifecycle Manager allows synchronous validation with various SOLA entities. The validation occurs whenever an asset is submitted for publish or when it is explicitly invoked by a user.

SOLAValidator

- **Behavior:**
  Before a service is published to SOLA, the validator can warn users of potential conflicts within SOLA. This validator will check if there will be any issues creating the service (or project if it doesn't already exist) in SOLA. It also perform name limitations (such as length or reserved characters). Items that will prevent publish are marked as severe errors, those that indicate potential issues will be warnings.
- **Class:** com.logiclibrary.integrations.sola.SOLAValidator
- **Properties:**
  - *federated-system-name*
    This property is used to specify the SOLA federated system that is contacted to perform validation.
- **Example:**

```
    <asset-validator name="SOLAValidator"
class="com.logiclibrary.integrations.sola.SOLAValidator">
      <properties>
        <property name="federated-system-name" value="SOLA" />
      </properties>
    </asset-validator>
```