Creating and sending signals

On this page

Creating a signal instance

You can create Signal Instances by using the following API. The signatures of this API are as follows

```
SignalInstance createSignal(Signal signal)
SignalInstance createSignal(String keyword)
```

The following example demonstrates how to create a Signal Instance

```
ALH.createSignal("start");
ALH.createObject("statemachine::signals::start");
ALH.createSignal(s); ---> if 's' is an instance of a Signal.
```

The parameter of this API can be either a String or a Signal. If it is a String, the system will find a Signal whose name or qualified name contains the String.

Sending a signal instance to a specific target object

You can send an existing Signal Instance (or create a new one and then send it) to a target object with the following APIs

```
void sendSignal(String signalName, Object_ object)
void sendSignal(String signalName, String targetName)
void sendSignal(SignalInstance signal, Object_ object)
void sendSignal(SignalInstance signal, String targetName)
void sendSignal(String signalName, Object_ target, String portName)
void sendSignal(SignalInstance signal, Object_ target, String portName)
```

The conditions that apply when creating an Instance are as follows

- If a signal name contains "::", it will find the signal from a qualified name. The signal will be found if its qualified name is ended with signalName.
- If an object is an instance of an Object_, send the signal to that Object_ directly.
- If an object is an instance of a String, there are two possible cases as follows
    - It will find the target object(s) from all waiting objects whose part/property names match the target's String parameter.
    - It will find the target object(s) through connected ports, given that the port is the name of the current object.

The following example shows how to send a specific signal to a specific target object in ALH API

```
ALH.sendSignal("play", o);  "o" references to a target object.
ALH.sendSignal("system::play", o);  Find a signal using a qualified name.
ALH.sendSignal("play", "Speaker");  Send to all waiting objects that have "Speaker"
as their type name.
ALH.sendSignal("play", "Player", "out2");  "Player" is an object, and "out2" is a port name.
```

All parameters must not be null. Otherwise, the ScriptEngine errors will be thrown.

Getting the last signal instance from a runtime object

You can use the following API to retrieve the last signal instance from a runtime object. It is equivalent to $signal$.

```
SignalInstance getLastSignal(Object_ o)
```

The following code fragment shows how to get the last signal instance in the event pool of a specified object using ALH API

```
ALH.getLastSignal(o);
```

It returns the Instance of the following

```
fUML.Semantics.CommonBehaviors.Communications.SignalInstance.
```