Integrating with AMQP

Overview

Advanced Message Queuing Protocol (AMQP) is an open standard for passing business messages between applications or organizations. You can use Service Designer module to define an AMQP task. An AMQP task publishes the service request to an external system (Process Orchestrator) via a message broker.

The AMQP sends the service request to an exchange, which accepts the message from Service Catalog and routes them to message queues. RabbitMQ is an open source message broker software that implements the AMQP standard. An external system (Process Orchestrator) will access RabbitMQ with the required APIs to retrieve the messages and process them.

A new task type called Queue Service Request is available in the Service Designer module under the Plan tab to publish the service request data to a message broker. For more information on how to publish data using AMQP task, see Cisco Prime Service Catalog Designer Guide.

Message Queue

An exchange accepts messages from a producer application and routes them to message queues. The exchange that will be created for the pre, post, and the main AMQP tasks will be all of 'fanout' type with a default queue created for each of them and bound to each of them. The names of the default queues would be <topic-name>_queue. The topic name is the name that the user enters in the **Service Designer > Plan** tab for the 'Queue Service Request' task.

Note To prevent poodle attack, Prime Service Catalog integration with RabbitMQ server supports SSL protocol with TLSv1.2 version only. If TLSv 1.2 is not specified, then clients cannot connect to RabbitMQ server for consumption of messages and connections fail with this exception: javax.net.ssl.SSLException: Received fatal alert: protocol_version

Following is a sample code to create an exchange and queue and how to consume a message.

```
package amqpProject;


import com.rabbitmq.client.ConnectionFactory;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.QueueingConsumer;
```

```
public class SampleProcess {

  public static void main(String[] argv) throws Exception {
    String exchangeName = "testExchange";
    String queueName = exchangeName+"_queue";
    String brokerIpAddress = "10.142.10.77";
    String userName = "admin";
    String password = "cisco123";
    ConnectionFactory factory = new ConnectionFactory();
    factory.useSslProtocol("TLSv1.2"); !--Included to prevent Poodle attack
    factory.setHost(brokerIpAddress);
    factory.setUsername(userName);
    factory.setPassword(password);

    Connection connection = factory.newConnection();

    Channel channel = connection.createChannel();
    channel.exchangeDeclare(exchangeName, "fanout", true, false, null);

    channel.queueDeclare(queueName, true, false, false, null);
    channel.queueBind(queueName, exchangeName, "");

    QueueingConsumer consumer = new QueueingConsumer(channel);
    channel.basicConsume(queueName, true, consumer);
    while (true) {

      QueueingConsumer.Delivery delivery = consumer.nextDelivery();
      String message = new String(delivery.getBody());
      System.out.println(" [x] Received from "+queueName+":-");
      System.out.println(message);

    }
  }
}
```

REST-based nsAPIs

There is a set of REST API for returning the message broker details for a service and its tasks. The signature and the response returned by it are given below. The input required is the service name that will be accepted as a path parameter. All the AMQP tasks for that service with the exchange details will be returned to the caller.

Given below is the sample response for a Service specific API:

http://localhost:8088/RequestCenter/nsapi/messagebroker/service/<service name>

{"tasks":[{"name":"PRE:amqpTask1","connectionIdentifier":"am1","messageFormat":"JSON","exc
hangeName":"AQAB_Pre","payloadType":"All Message Details
(large)","isAutoComplete":"true","transformationType":"JOLT","outboundTransformationName":
"","inboundTransformationName":""},{"name":"amqpTask1","connectionIdentifier":"am1","messa
geFormat":"JSON","exchangeName":"post_exchange_0307","payloadType":"All Message Details
(large)","isAutoComplete":"true","transformationType":"JOLT","outboundTransformationName":
"","inboundTransformationName":""},{"name":"POST:amqpTask1","connectionIdentifier":"am1","
messageFormat":"JSON","exchangeName":"AQAB_Post","payloadType":"All Message Details
(large)","isAutoComplete":"false","transformationType":"JOLT","outboundTransformationName"
:"","inboundTransformationName":""}]}

Overview API

The Overview API gathers all the exchanges that are configured in the Service Designer module and makes a call to find out what are the exchanges and queues that are already created in the RabbitMQ server. The Overview API performs a lookup on AMQP server to get information on queues and the output is sent across in JSON format.

Using Overview API and noCache parameter:

- When you specify RabbitMQ information through the UI and use the Overview API without noCache parameter, the information is retrieved from both cache and database.
- However, when you directly access database to insert RabbitMQ information, you must use Overview API with noCache =1 to ensure that the latest information is fetched.

The sample response from the RabbitMQ server is as follows:

http://localhost:8088/RequestCenter/nsapi/messagebroker/overview

{"connections":[{"connectionIdentifier":"am1","host":"10.78.0.247","port":5671},{"conne
ctionIdentifier":"am2","host":"10.78.0.247","port":5671},{"connectionIdentifier":"am3",
"host":"10.78.0.247","port":5671},{"connectionIdentifier":"AM4","host":"10.78.0.247","p
ort":5671},{"connectionIdentifier":"am5","host":"10.78.0.247","port":5671}]}

http://localhost:8080/RequestCenter/nsapi/messagebroker/overview?connectionIdentifier=am1

{"rabbitmq_version":"3.6.1","username":"admin","password":"UhP0zDGRoC2R7isv9qCQ6A==","i
pAddress":"10.78.0.247","recoveryInterval":300000,"vhost":"/","inboundQueue":"psc_inbou
nd_queue","authorization_key":"owL7ViRfE4Sce0aG1jSzZInkIVw5CsM7acQ5rlswpXzF/kForoUjlfrV
VUOuA+CqSFSTuJlLQ5GRUMmzzxbV2xtrMvhGed6x1WU08MbDLKydSSY5UQvoAS7aZ0dROdXvF+G4uD4nAlQ6HSI
c6dBct3M+dDJcm02z9OshvMaCmkvZa380B8/MbbBhu5Q3FntzWkAVY/FobU9gvlidoDt1Ty0CmAgfvPbP6joLXY
aTPHjaqjYaBaX2Y4m+1V7Wm3Rb+oLpHZkCVm7Pr1zlLByPs6d+qaQShIfr0yeXXbZrQNh3s8qHl+YbYFRtNin/J
EeLa6pY5J19m6zUV78n2BstRQ==","message-time-to-live":"300000","useSSL":true,"port":"5671
","exchanges":[{"name":"exchange2","vhost":"/","type":"fanout","created":"false"},{"nam
e":"HeatStack","vhost":"/","type":"fanout","created":"true"},{"name":"exchange3","vhost
":"/","type":"fanout","created":"false"}],"queues":[{"name":"exchange2_queue","vhost":"
/","created":"false"},{"name":"HeatStack_queue","vhost":"/","created":"true"},{"name":"
exchange3_queue","vhost":"/","created":"false"}],"certificate":"-----BEGIN
CERTIFICATE-----\nMIIC5DCCAcygAwIBAgIBATANBgkqhkiG9w0BAQsFADATMREwDwYDVQQDDAhNeVRlc3RDQ
TAeFw0x\nNjA2MTExMTUwMTJaFw0xNzA2MTExMTUwMTJaMCcxFDASBgNVBAMMCyQoaG9zdG5hbWUpMQ8wDQYD\n
VQQKDAZzZXJ2ZXIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCq8v2yJT7tv+nOwFSo\nEE1c0oVg3
skd86JN7jVJaz/mMOyJjDmf1147iUwZPMTTCB34ovYUXFkw+a+0ext2WRHgQLTMPvVO\nA86jwuPd/bhUxXg8je
EfE4V/1Seci9Xz+5VxqCybCNOzJQ12/vLXvIJK43U/+1GdXnpWxFaF0yd0\nht3iUy6mfUAHfNMI2SOfJwbbdUa
MyD0/Krsiu+X+vFQBUDmM7Y0priItiDVDq7rxug2UOPACzzMG\n5yJ5aJjNLSlJRwKKst/jjvesqHIgWNo0qKvk
TET3tIVsKDi1Fn9IdKQuuoI1n225+58cWSANmZ5M\n4BdSIf4z6QRuKliBRi55AgMBAAGjLzAtMAkGA1UdEwQCM
AAwCwYDVR0PBAQDAgUgMBMGA1UdJQQM\nMAoGCCsGAQUFBwMBMA0GCSqGSIb3DQEBCwUAA4IBAQC4L9fk4kuu/d
pLeFWNUhb5Uyk6AqbTRAyD\nlq1m11E09EhmTH7cnoFsz0ELBDppASIUADSb9jmUeNKJtjW94gq8luSem1Z8lQz

UCOCE6rsaznrw\nm9jJO7gXA5SSmy7PgdokdhbeTzlSYA3kkkR5ZE8M403Qv5cEPREqnshs6f6bQSm8tzSNETy3
OyHL\npUo7YVTBCfJMQ/e2nZxCJSDuEL6QaIL4kmEYeu8j/1RplBAofMkDfDe+yMx2MJYl+MopVggGexpa\nMqy
bCchrDTJ/8sI/R18Ld80TQ2Km70sQqvNVenCpGCtgGIcupRWaAOpsQH0PNaUK+lcwYRIvf0VS\n09QE\n-----E
ND CERTIFICATE--"}

Encrypt Credentials using Public Key GUID

Prime Service Catalog supports a secure way to return AMQP credentials, using the GUID of the Public Key passed to the nsAPI.

If the external system Public Key GUID is passed as a Query parameter, then credentials are encrypted (using PO Secure String Format) in the response with the Public Key associated to the GUID

If the external system Public Key GUID is not passed, encrypted string in DB is returned as is (note: external system cannot decrypt this)

Encrypted sensitive data is returned in the following AMQP overview APIs:

http://localhost:8088/RequestCenter/nsapi/messagebroker/overview?publicKeyGUID=...

Encryption is done using the public key configured at the connection level.

Generating AuthorizationKey API

For Service Item operations like create, update, and delete, Authorization Key is of more of importance than channel-id. But channel-id is also required in the message for uniquely identifying the external task. Authorization key generated using the below nsAPI is unique for a user. Authorization key is same for all the AMQP connections for a particular user even though its value seems to change every time the above nsAPI is called. This key is used for authorizing the user and to access various modules based on the permissions the user is granted.

Authorization key can be obtained from the following nsAPI:

**http://<ServerURL>
/RequestCenter/nsapi/messagebroker/overview?connectionIdentifier=<particular amqp connection identifier>**

{"rabbitmq_version":"3.6.1","username":"admin","password":"UhP0zDGRoC2R7isv9qCQ6A==","ipAd
dress":"10.78.0.247","recoveryInterval":300000,"vhost":"/","inboundQueue":"psc_inbound_que
ue","authorization_key":"owL7ViRfE4Sce0aG1jSzZInkIVw5CsM7acQ5rlswpXzF/kForoUjlfrVVUOuA+CqS
FSTuJlLQ5GRUMmzzxbV2xtrMvhGed6x1WU08MbDLKydSSY5UQvoAS7aZ0dROdXvF+G4uD4nAlQ6HSIc6dBct3M+dDJ
cm02z9OshvMaCmkvZa380B8/MbbBhu5Q3FntzWkAVY/FobU9gvlidoDt1Ty0CmAgfvPbP6joLXYaTPHjaqjYaBaX2Y
4m+1V7Wm3Rb+oLpHZkCVm7Pr1zlLByPs6d+qaQShIfr0yeXXbZrQNh3s8qHl+YbYFRtNin/JEeLa6pY5J19m6zUV78
n2BstRQ==","message-time-to-live":"300000","useSSL":true,"port":"5671","exchanges":[{"name
":"exchange2","vhost":"/","type":"fanout","created":"false"},{"name":"HeatStack","vhost":"
/","type":"fanout","created":"true"},{"name":"exchange3","vhost":"/","type":"fanout","crea
ted":"false"}],"queues":[{"name":"exchange2_queue","vhost":"/","created":"false"},{"name":
"HeatStack_queue","vhost":"/","created":"true"},{"name":"exchange3_queue","vhost":"/","cre
ated":"false"}],"certificate":"-----BEGIN
CERTIFICATE-----\nMIIC5DCCAcygAwIBAgIBATANBgkqhkiG9w0BAQsFADATMREwDwYDVQQDDAhNeVRlc3RDQTAe
Fw0x\nNjA2MTExMTUwMTJaFw0xNzA2MTExMTUwMTJaMCcxFDASBgNVBAMMCyQoaG9zdG5hbWUpMQ8wDQYD\nVQQKDA
ZzZXJ2ZXIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCq8v2yJT7tv+nOwFSo\nEE1c0oVg3skd86JN7j
VJaz/mMOyJjDmf1147iUwZPMTTCB34ovYUXFkw+a+0ext2WRHgQLTMPvVO\nA86jwuPd/bhUxXg8jeEfE4V/1Seci9
Xz+5VxqCybCNOzJQ12/vLXvIJK43U/+1GdXnpWxFaF0yd0\nht3iUy6mfUAHfNMI2SOfJwbbdUaMyD0/Krsiu+X+vF
QBUDmM7Y0priItiDVDq7rxug2UOPACzzMG\n5yJ5aJjNLSlJRwKKst/jjvesqHIgWNo0qKvkTET3tIVsKDi1Fn9IdK
QuuoI1n225+58cWSANmZ5M\n4BdSIf4z6QRuKliBRi55AgMBAAGjLzAtMAkGA1UdEwQCMAAwCwYDVR0PBAQDAgUgMB
MGA1UdJQQM\nMAoGCCsGAQUFBwMBMA0GCSqGSIb3DQEBCwUAA4IBAQC4L9fk4kuu/dpLeFWNUhb5Uyk6AqbTRAyD\n
lq1m11E09EhmTH7cnoFsz0ELBDppASIUADSb9jmUeNKJtjW94gq8luSem1Z8lQzUCOCE6rsaznrw\nm9jJO7gXA5SS

my7PgdokdhbeTzlSYA3kkkR5ZE8M403Qv5cEPREqnshs6f6bQSm8tzSNETy3OyHL\npUo7YVTBCfJMQ/e2nZxCJSDu
EL6QaIL4kmEYeu8j/1RplBAofMkDfDe+yMx2MJYl+MopVggGexpa\nMqybCchrDTJ/8sI/R18Ld80TQ2Km70sQqvNV
enCpGCtgGIcupRWaAOpsQH0PNaUK+lcwYRIvf0VS\n09QE\n-----END CERTIFICATE--"}

Transforming JSON Using JOLT Work flow

From Prime Service catalog 12.0 onwards, inbound transformations are also supported. However, transformations are not supported for Service Item operations. For inbound messages transformation types supported are XSL and JOLT, and for outbound XSL, JOLT, and FTL are supported.

The below is the high-level work flow for creating a transformations are as follows:

Step 1 Add an AMQP connection from **Administration > Manage Connections > AMQP**. Select the For more information, see section Managing AMQP Connections in the *Cisco Prime Service Catalog Administration and Operation Guide*.

Step 2 Create outbound and inbound transformations in JOLT format. For details see Managing Transformations, page 5-18.

Step 3 In the service designer configure AMQP task for the service request. For details see section Configuring AMQP Tasks for Publishing Service Request to an External System of *Cisco Prime Service Catalog Designer Guide*.

Points to remember:

- You can define default values for the following which can be overridden in AMQP task parameter popup page.
  - "Public Key
  - "Message Type i.e the message format (XML/JSON) in which outbound and inbound message processing would happen by default for the particular connection.
- Message type can be set at the AMQP connection level or at the task level, but the task level setting takes precedence.
- Different transformations are fetched based on the transformation type selected i.e., if XML is selected only XSL transformations will be available for outbound transformation, inbound transformation selection and for JSON it can be either JOLT or FTL transformations.
- FTL transformation is supported only for outbound messages.

Sample JSON Transformation Using JOLT

JSON inbound AMQP input:

```
{
    "message": {
        "updateData": {
            "dataValue": [
```

```
                {
                    "namePO": "TextField.Text",
                    "valuePO": [
                        "QoE"
                    ],
                    "multiValuedPO": false
                }
            ]
        },
        "addCommentsPO": {
            "comment": [
                "test comment 1",
                "test comment 2",
                "test comment 3"
            ]
        },
        "takeAction": {
            "actionPO": "DONE"
        },
        "channelId": "51515F28-E36A-478C-8773-E35B215CF36C"
    }
}
```

JOLT Transformation spec:

```
[
  {
    "operation": "shift",
    "spec": {
      "message": {
        "updateData": {
          "dataValue": {
            "*": {
              "namePO": "message.updateData.dataValue.[&1].name",
              "valuePO": "message.updateData.dataValue.[&1].value",
              "multiValuedPO": "message.updateData.dataValue.[&1].multiValued"
            }
          }
        },
        "*": "&1.&0",
        "channelId": "message.channelId"
      }
    }
  },
  {
    "operation": "shift",
    "spec": {
      "message": {
        "takeAction": {
          "actionPO": "message.&1.action"
        },
        "*": "&1.&0",
        "channelId": "message.channelId"
      }
    }
  },
  {
    "operation": "shift",
    "spec": {
      "message": {
        "addCommentsPO": "message.addComments",
        "*": "&1.&0",
        "channelId": "message.channelId"
```

```
      }
    }
  }
]
```

JSON Output:

Below is the output generated by Prime Service Catalog inbound AMQP code based on the above transformation and input, which is finally processed.

```
{
  "message" : {
    "addComments" : {
      "comment" : [ "test comment 1", "test comment 2", "test comment 3" ]
    },
    "channelId" : "51515F28-E36A-478C-8773-E35B215CF36C",
    "takeAction" : {
      "action" : "DONE"
    },
    "updateData" : {
      "dataValue" : [ {
        "multiValued" : false,
        "name" : "TextField.Text",
        "value" : [ "QoE" ]
      } ]
    }
  }
}
```

Sample AMQP Inbound XML

```
<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
<update-data>
<data-value multi-valued="false">
<name>amqpDict1.field1</name>
<value>a_newer</value>
</data-value>
</update-data>
</message>

<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
<add-comments>
<comment>testing new comments</comment>
</add-comments>
</message>


<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
<take-action action="done">
</take-action>
</message>
```

Sample AMQP Inbound JSON

```
{
  "message" : {
    "addComments" : {
      "comment" : [ "test comment 1", "test comment 2", "test comment 3" ]
    },
```

```
    "channelId" : "51515F28-E36A-478C-8773-E35B215CF36C",
    "takeAction" : {
      "action" : "DONE"
    },
    "updateData" : {
      "dataValue" : [ {
        "multiValued" : false,
        "name" : "TextField.Text",
        "value" : [ "QoE" ]
      } ]
    }
  }
}
```

Inbound Message

Two types of operations are supported for inbound messages from the third-party system—requisition operations and service item operations.

The most important element within the nsXML is the channel-id, an ID that uniquely identifies the external task. This ID is provided to the third-party system and needs to appear in their response if the corresponding data update is to be successfully applied by the business engine.

Requisition Operations for AMQP

Requisition operations supported for AMQP are similar to Service Link requisition operations. However, in AMQP inbound messages supports XML and JSON formats. For details on each of the operations, see section Inbound nsXML Message, page 5-25.

take-action

The take-action operation marks the delivery task as completed.

```
<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
<take-action action="done">
</take-action>
</message>
```

update-data

The dictionary fields data in the requisition are updated with the new values.

```
<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
<update-data>
<data-value multi-valued="false">
<name>amqpDict1.field1</name>
<value>a_newer</value>
</data-value>
</update-d n ata>
</message>
```

add-comment

An add-comments message is used to add comments to the System Comments section of the requisition.

```
<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
<add-comments>
<comment>testing new comments</comment>
```

```
</add-comments>
</message>
```

Service Item Operations for AMQP

Note Transformations are not supported for Service Item operations.

For the service item operations, authorization key must be generated and included in the messages. Use the below nsAPI to generate the authorization key:

```
http://<ServerURL>
/RequestCenter/nsapi/messagebroker/overview?connectionIdentifier=<particular amqp
connection identifier>
```

For more details on this API, see section Generating AuthorizationKey API, page 4-4. Below are the sample Service Item operations for AMQP.

create

In create messages, the attribute values are added to the service items.

```
<message channel-id="A984F860-DBE7-48EB-B3A2-F6A0159B093F"
authorization-key="liDBm1NeRVbVmwbBmAd+cca0/Z8jf863aYOKl3QXSWcYwn5aGuPxn09CjDzv5EKR7/CcJx5
+2gu1MhUopAuiX3WwjX/DP1Xw3nmFLMz6dmaiiq0+5v4XOKmNqf1J3GvBJYjTxAkH1VbCQ2y1fNxE8/cP5wyd3mU6M
LlD9tjc/Iro950gLlTq+9C2/QMis6ya52O2D8F652jnHnWbZrDf6zPdOS1FpCKkg05YSVpqifGgqjEh3RTyPs9w0Qc
NcxWcEqD1vvBMNW0VAL/YaW+MNmoUnpglR0cUeOJ3WFr8/1l/uyNyhfvAYfDUznVCNfVDtkMzCWhA4eH25XQUtmKwG
w==">
<create>
<serviceitem>
<name>AnandServiceItem</name>
<serviceItemData>
<serviceItemAttribute name="Name">Hannah</serviceItemAttribute>
<serviceItemAttribute name="DOB">1995-01-19</serviceItemAttribute>
<serviceItemAttribute name="Age">25</serviceItemAttribute>
</serviceItemData>
</serviceitem>
</create>
</message>
```

update

In update messages, omitting a service item attribute results in no change to the attribute value. When an attribute is explicitly specified in the message but contains no value, the value of the attribute for the service item is set to blank for text fields and zero for numeric fields.

```
<message channel-id="A984F860-DBE7-48EB-B3A2-F6A0159B093F"
authorization-key="liDBm1NeRVbVmwbBmAd+cca0/Z8jf863aYOKl3QXSWcYwn5aGuPxn09CjDzv5EKR7/CcJx5
+2gu1MhUopAuiX3WwjX/DP1Xw3nmFLMz6dmaiiq0+5v4XOKmNqf1J3GvBJYjTxAkH1VbCQ2y1fNxE8/cP5wyd3mU6M
LlD9tjc/Iro950gLlTq+9C2/QMis6ya52O2D8F652jnHnWbZrDf6zPdOS1FpCKkg05YSVpqifGgqjEh3RTyPs9w0Qc
NcxWcEqD1vvBMNW0VAL/YaW+MNmoUnpglR0cUeOJ3WFr8/1l/uyNyhfvAYfDUznVCNfVDtkMzCWhA4eH25XQUtmKwG
w==">
 <update>
   <serviceitem>
   <name>AnandServiceItem</name>
   <serviceItemData>
   <serviceItemAttribute name="Name">Anand2update</serviceItemAttribute>
   <serviceItemAttribute name="RAM">Primarymemory3</serviceItemAttribute>
   <serviceItemAttribute name="MemoryInt">759001</serviceItemAttribute>
   <serviceItemAttribute name="Model512">Lenovo T440 updated</serviceItemAttribute>
   <serviceItemAttribute name="Money">80000</serviceItemAttribute>
   <serviceItemAttribute name="ManufTime">2016-05-01 16:45</serviceItemAttribute>
   </serviceItemData>
```

```
   </serviceitem>
  </update>
```

delete

Delete service item requests require only the names for the service item type and instance. Additional service item attribute and subscription information is ignored.

```
<message channel-id="A984F860-DBE7-48EB-B3A2-F6A0159B093F"
authorization-key="liDBm1NeRVbVmwbBmAd+cca0/Z8jf863aYOKl3QXSWcYwn5aGuPxn09CjDzv5EKR7/CcJx5
+2gu1MhUopAuiX3WwjX/DP1Xw3nmFLMz6dmaiiq0+5v4XOKmNqf1J3GvBJYjTxAkH1VbCQ2y1fNxE8/cP5wyd3mU6M
LlD9tjc/Iro950gLlTq+9C2/QMis6ya52O2D8F652jnHnWbZrDf6zPdOSlFpCKkg05YSVpqifGgqjEh3RTyPs9w0Qc
NcxWcEqD1vvBMNW0VAL/YaW+MNmoUnpglR0cUeOJ3WFr8/1l/uyNyhfvAYfDUznVCNfVDtkMzCWhA4eH25XQUtmKwG
w==">
   <delete>
      <serviceitem>
      <name>ServiceItem</name>
      <serviceItemData>
      <serviceItemAttribute name="Name">Anand3</serviceItemAttribute>
      </serviceItemData>
   </serviceitem>
   </delete>
</message>
```

hierarchy

In hierarchy messages, the attribute values are nested in the service items.

Outbound Message

The message format published is in the nsXML, JSON, or FTL formats similar to the format that is currently used by Service Link.

```
{
        "message": {
                "create": {
                        "hierarchy": true,
                        "serviceitem": [ {
                                "name": "Computer",
                                "serviceItemData": [{
                                        "Name": "Dell-HP Computer",
                                        "OS": "Linux 7",
                                        "RAM": "7878",
                                        "HD": "520",
                                        "Mouse": {
                                                "Name": "HPE"
                                        },
                                        "Network": {
                                                "Name": "My Mac Network",
                                                "IPAddress": "3.3.3.3",
```

```
                                                    "MACAddress": "12.12.12.12",
                                                    "AdapterType": "MAC"
                                                }
                                            }]
                                    }]
                            },
                            "channelId": "6A55AF3D-4EEC-402C-9055-CB79081C138F"
                    }
            }
```

Sample XML Outbound Message

The sample nsXml Data Structure is given below.

- Sample message structure for Data; No Service Details(default; small)

```
<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="F203ACC7-E6CA-A2EA-E040-007F0101140E"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<task-started task-type="task">
<task>
<actual-duration>0.0</actual-duration>
<completed-date/>
<context-id>69</context-id>
<context-type>Requisition Entry</context-type>
<due-date>2014-04-25 20:00:00</due-date>
<effort>10.0</effort>
<estimated-date/>
<expected-duration>10.0</expected-duration>
<flag-id>0</flag-id>
<is-sharable>true</is-sharable>
<is-shared>true</is-shared>
<next-action-id>2</next-action-id>
<performer-actual-duration>0.0</performer-actual-duration>
<priority>2</priority>
<scheduled-start-date>2014-04-24 18:00:00</scheduled-start-date>
<start-date>2014-04-23 14:11:13</start-date>
<state-id>2</state-id>
<subject>queueServiceRequest1</subject>
<task-id>266</task-id>
</task>
<requisition>
<services>0</services>
<actual-cost>0.0</actual-cost>
<actual-duration>0.0</actual-duration>
<closed-on/>
<customer>
<company-address/>
<email>internal@newscale.com</email>
<fax/>
<first-name>admin</first-name>
<home-ou>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</home-ou>
```

```
<home-phone/>
<last-name>admin</last-name>
<login-name>admin</login-name>
<person-id>1</person-id>
<personal-address/>
<supervisor-name/>
<timezone>Pacific Standard Time</timezone>
<work-phone/>
</customer>
<due-on>2014-04-25 20:00:00</due-on>
<expected-cost>0.0</expected-cost>
<expected-duration>0.0</expected-duration>
<external>false</external>
<initiator>
<company-address/>
<email>internal@newscale.com</email>
<fax/>
<first-name>admin</first-name>
<home-ou>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</home-ou>
<home-phone/>
<last-name>admin</last-name>
<login-name>admin</login-name>
<person-id>1</person-id>
<personal-address/>
<supervisor-name/>
<timezone>Pacific Standard Time</timezone>
<work-phone/>
</initiator>
<organizational-unit>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</organizational-unit>
<requisition-entry>
<closed-date/>
<data-values>
<data-value multi-valued="false">
<name>amqpDict1.field1</name>
<value>a</value>
</data-value>
<data-value multi-valued="false">
<name>amqpDict1.field2</name>
<value>b</value>
</data-value>
<data-value is-secure="true" multi-valued="false">
<name>amqpDict1.field3</name>
<value>ut3u4RC699wz7Jd2O+4cix23m9XXgAC/EsDiHp1ERsOpdKSKdrtgMjFWcVoO96aCFSkCgwa2tkBo
vKoOHzaillNiJ47+aY8zCWKwd17tCjmMLEkeQlTvrDvIHR/DT1iWSmN08DI9+Ns7hY3A/g6ijUoM
gcuMczH+5F/pGtgupLZF/L7FwOwu4VcKVWM/2N5tuXGz+1aHTRAAAAByQXr/yD/75Ysy57LnQLxc
</value>
</data-value>
<data-value is-secure="true" multi-valued="false">
<name>amqpDict1.field4</name>
<value>ut3u4RC699wz7Jd2O+4cix23m9XXgABReynDp5AdBgRi5ivhS05Unv98BgWgxc5YNcZrCihhhH/1
rzZqhjiIjAoRelIjADDb3IQP72armXsLRTvh0/fusd4jLdVIm4q1s+GaSTt6F3oqQeZ4RLhVUopo
p2zNAxmjaGj629C8gWREes3Z8EjDvXg5K1YVi90fXJV1jDoADhAAAABP1hct5TNV2d8R1cN/qDtK
</value>
</data-value>
</data-values>
<due-date>2014-04-25 20:00:00</due-date>
<item-number>1</item-number>
<price-per-unit>0.0</price-per-unit>
```

```
<priced>true</priced>
<quantity>1</quantity>
<rejected>false</rejected>
<rejected-date/>
<requisition-entry-id>69</requisition-entry-id>
<revision-number>105</revision-number>
<service>
<estimated-cost>0.0</estimated-cost>
<name>queueService1</name>
<parameters>
<default-duration>0.0</default-duration>
<priority>2</priority>
<start-date/>
<start-mode>0</start-mode>
</parameters>
<pricing-schema>0</pricing-schema>
<quantity>1</quantity>
<service-id>2</service-id>
<version>105</version>
<standard-duration>0.0</standard-duration>
</service>
<start-after/>
<start-date>2014-04-23 14:10:48</start-date>
<start-mode>0</start-mode>
<status>1</status>
</requisition-entry>
<requisition-id>64</requisition-id>
<started-on>2014-04-23 14:10:47</started-on>
<status>1</status>
</requisition>
<context>
<requisitionentryref itemnumber="1"/>
</context>
</task-started>
</message>
```

- Sample message structure with secure strings in it.

```
<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="F203ACC7-E6CA-A2EA-E040-007F0101140E"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<task-started task-type="task">
<task>
<actual-duration>0.0</actual-duration>
<completed-date/>
<context-id>69</context-id>
<context-type>Requisition Entry</context-type>
<due-date>2014-04-25 20:00:00</due-date>
<effort>10.0</effort>
<estimated-date/>
<expected-duration>10.0</expected-duration>
<flag-id>0</flag-id>
<is-sharable>true</is-sharable>
<is-shared>true</is-shared>
<next-action-id>2</next-action-id>
<performer-actual-duration>0.0</performer-actual-duration>
<priority>2</priority>
<scheduled-start-date>2014-04-24 18:00:00</scheduled-start-date>
<start-date>2014-04-23 14:11:13</start-date>
<state-id>2</state-id>
<subject>queueServiceRequest1</subject>
<task-id>266</task-id>
</task>
<requisition>
```

```
<services>0</services>
<actual-cost>0.0</actual-cost>
<actual-duration>0.0</actual-duration>
<closed-on/>
<customer>
<company-address/>
<email>internal@newscale.com</email>
<fax/>
<first-name>admin</first-name>
<home-ou>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</home-ou>
<home-phone/>
<last-name>admin</last-name>
<login-name>admin</login-name>
<person-id>1</person-id>
<personal-address/>
<supervisor-name/>
<timezone>Pacific Standard Time</timezone>
<work-phone/>
</customer>
<due-on>2014-04-25 20:00:00</due-on>
<expected-cost>0.0</expected-cost>
<expected-duration>0.0</expected-duration>
<external>false</external>
<initiator>
<company-address/>
<email>internal@newscale.com</email>
<fax/>
<first-name>admin</first-name>
<home-ou>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</home-ou>
<home-phone/>
<last-name>admin</last-name>
<login-name>admin</login-name>
<person-id>1</person-id>
<personal-address/>
<supervisor-name/>
<timezone>Pacific Standard Time</timezone>
<work-phone/>
</initiator>
<organizational-unit>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</organizational-unit>
<requisition-entry>
<closed-date/>
<data-values>
<data-value multi-valued="false">
<name>amqpDict1.field1</name>
<value>a</value>
</data-value>
<data-value multi-valued="false">
<name>amqpDict1.field2</name>
<value>b</value>
</data-value>
<data-value is-secure="true" multi-valued="false">
<name>amqpDict1.field3</name>
<value>ut3u4RC699wz7Jd2O+4cix23m9XXgAC/EsDiHp1ERsOpdKSKdrtgMjFWcVoO96aCFSkCgwa2tkBo
vKoOHzaillNiJ47+aY8zCWKwd17tCjmMLEkeQlTvrDvIHR/DT1iWSmN08DI9+Ns7hY3A/g6ijUoM
gcuMczH+5F/pGtgupLZF/L7FwOwu4VcKVWM/2N5tuXGz+1aHTRAAAAByQXr/yD/75Ysy57LnQLxc
```

```
</value>
</data-value>
<data-value is-secure="true" multi-valued="false">
<name>amqpDict1.field4</name>
<value>ut3u4RC699wz7Jd2O+4cix23m9XXgABReynDp5AdBgRi5ivhS05Unv98BgWgxc5YNcZrCihhhH/1
rzZqhjiIjAoRelIjADDb3IQP72armXsLRTvh0/fusd4jLdVIm4q1s+GaSTt6F3oqQeZ4RLhVUopo
p2zNAxmjaGj629C8gWREes3Z8EjDvXg5K1YVi90fXJV1jDoADhAAAABP1hct5TNV2d8R1cN/qDtK
</value>
</data-value>
</data-values>
<due-date>2014-04-25 20:00:00</due-date>
<item-number>1</item-number>
<price-per-unit>0.0</price-per-unit>
<priced>true</priced>
<quantity>1</quantity>
<rejected>false</rejected>
<rejected-date/>
<requisition-entry-id>69</requisition-entry-id>
<revision-number>105</revision-number>
<service>
<estimated-cost>0.0</estimated-cost>
<name>queueService1</name>
<parameters>
<default-duration>0.0</default-duration>
<priority>2</priority>
<start-date/>
<start-mode>0</start-mode>
</parameters>
<pricing-schema>0</pricing-schema>
<quantity>1</quantity>
<service-id>2</service-id>
<version>105</version>
<standard-duration>0.0</standard-duration>
</service>
<start-after/>
<start-date>2014-04-23 14:10:48</start-date>
<start-mode>0</start-mode>
<status>1</status>
</requisition-entry>
<requisition-id>64</requisition-id>
<started-on>2014-04-23 14:10:47</started-on>
<status>1</status>
</requisition>
<context>
<requisitionentryref itemnumber="1"/>
</context>
</task-started>
</message>
```

Sample JSON Outbound Message

```
{
  "message": {
    "taskStarted": {
      "task": {
        "actualDuration": 0.0,
        "calendarEntries": {},
        "checkLists": {},
        "completedDate": "",
        "contextId": 17,
        "contextType": "Requisition Entry",
```

```
"dueDate": "2016-11-29 19:00:00",
"effort": 10.0,
"estimatedDate": "",
"expectedDuration": 10.0,
"flagId": 0,
"isSharable": true,
"isShared": true,
"nextActionId": 2,
"performer": {
  "companyAddress": "",
  "email": "",
  "fax": "",
  "firstName": "Default Service Delivery",
  "homeOu": {
    "name": "<s ID=\"847\"/>",
    "organizationalUnitId": 1
  },
  "homePhone": "",
  "lastName": "Queue",
  "loginName": "",
  "personId": 2,
  "personalAddress": "",
  "supervisorName": "",
  "timezone": "Pacific Standard Time",
  "workPhone": ""
},
"performerRole": {
  "name": ""
},
"performerActualDuration": 0.0,
"priority": 2,
"queue": {
  "companyAddress": "",
  "email": "",
  "fax": "",
  "firstName": "Default Service Delivery",
  "homeOu": {
    "name": "<s ID=\"847\"/>",
    "organizationalUnitId": 1
  },
  "homePhone": "",
  "lastName": "Queue",
  "loginName": "",
  "personId": 2,
  "personalAddress": "",
  "supervisorName": "",
  "timezone": "Pacific Standard Time",
  "workPhone": ""
},
"scheduledStartDate": "2016-11-28 17:00:00",
"startDate": "2016-11-26 12:32:36",
"stateId": 2,
"subject": "amqpTask1",
"supervisor": {
  "companyAddress": "",
  "email": "",
  "fax": "",
  "firstName": "Default Service Delivery",
  "homeOu": {
    "name": "<s ID=\"847\"/>",
    "organizationalUnitId": 1
  },
  "homePhone": "",
  "lastName": "Queue",
```

```
    "loginName": "",
    "personId": 2,
    "personalAddress": "",
    "supervisorName": "",
    "timezone": "Pacific Standard Time",
    "workPhone": ""
  },
  "supervisorRole": {
    "name": ""
  },
  "taskId": 67,
  "waiting": false
},
"requisition": {
  "services": 0,
  "actualCost": 0.0,
  "actualDuration": 0.0,
  "closedOn": "",
  "comments": {},
  "costCenterCode": "",
  "customer": {
    "companyAddress": "",
    "email": "internal@newscale.com",
    "fax": "",
    "firstName": "admin",
    "homeOu": {
      "name": "<s ID=\"847\"/>",
      "organizationalUnitId": 1
    },
    "homePhone": "",
    "lastName": "admin",
    "loginName": "admin",
    "personId": 1,
    "personalAddress": "",
    "supervisorName": "",
    "timezone": "Pacific Standard Time",
    "workPhone": ""
  },
  "dueOn": "2016-11-26 12:32:35",
  "expectedCost": 0.0,
  "expectedDuration": 0.0,
  "external": false,
  "initiator": {
    "companyAddress": "",
    "email": "internal@newscale.com",
    "fax": "",
    "firstName": "admin",
    "homeOu": {
      "name": "<s ID=\"847\"/>",
      "organizationalUnitId": 1
    },
    "homePhone": "",
    "lastName": "admin",
    "loginName": "admin",
    "personId": 1,
    "personalAddress": "",
    "supervisorName": "",
    "timezone": "Pacific Standard Time",
    "workPhone": ""
  },
  "invocations": {},
  "organizationalUnit": {
    "name": "<s ID=\"847\"/>",
    "organizationalUnitId": 1
```

```
},
"requisitionEntry": [
  {
    "closedDate": "",
    "dataValues": {
      "dataValue": [
        {
          "name": "amqpDict1.field1",
          "value": [
            "a"
          ],
          "multiValued": false
        },
        {
          "name": "amqpDict1.field2",
          "value": [
            "b"
          ],
          "multiValued": false
        }
      ]
    },
    "dueDate": "2016-11-29 19:00:00",
    "itemNumber": 1,
    "pricePerUnit": 0.0,
    "priced": true,
    "quantity": 1,
    "rejected": false,
    "rejectedDate": "",
    "rejector": {
      "companyAddress": "",
      "email": "",
      "fax": "",
      "firstName": "",
      "homeOu": {},
      "homePhone": "",
      "lastName": "",
      "loginName": "",
      "personId": 0,
      "personalAddress": "",
      "supervisorName": "",
      "timezone": "",
      "workPhone": ""
    },
    "requisitionEntryId": 17,
    "revisionNumber": 3,
    "service": {
      "estimatedCost": 0.0,
      "form": {
        "fields": {}
      },
      "name": "amqpService1",
      "parameters": {
        "defaultDuration": 0.0,
        "priority": 2,
        "startDate": "",
        "startMode": 0
      },
      "pricingSchema": 0,
      "quantity": 1,
      "serviceId": 142,
      "version": 3,
      "standardDuration": 0.0
    },
```

```
            "startAfter": "",
            "startDate": "2016-11-26 12:32:36",
            "startMode": "0",
            "status": 1,
            "bundle": false
          }
        ],
        "requisitionId": 17,
        "startedOn": "2016-11-26 12:32:35",
        "status": 1
      },
      "context": {
        "requisitionentryref": {
          "itemnumber": 1
        }
      },
      "taskType": "task"
    },
    "channelId": "42346EF5-D2A7-EC03-E050-11AC02003EE1"
  }
}
```