General Meshlet Configurations for DataSync

The file that controls the configuration settings for a **Perspectium Meshlet** is a yaml file, generally named **application.yml** or **application-dev.yml**. This file is included as part of a meshlet's installation. To make changes to the configuration settings for your meshlet, open the yaml file in a text editing application and modify the configurations described in the sections below.

What's on this page?

- Spring - RabbitMT Configuration
- Perspectium - Message Configuration
- Example of a meshlet configuration yaml file

The configurations described below are the **essential** configurations that must be set for your meshlet. Additional configurations will depend on your integration; see the respective documentation (such as DataSync for Snowflake or ServiceBond for Jira) for further details.

Spring - RabbitMT Configuration

Meshlets connect to the **Perspectium Integration Mesh** over AMQP/AMQPS. To configure your connection to the Integration Mesh, you will need to include the following configurations under **spring.rabbitmq**:

Key	Default Value	Description
host		Host name for connecting to the Perspectium Integration Mesh. ```spring:\n rabbitmq:\n host: localhost```
port		The AMQP or AMQPS port for connecting to the Perspectium Integration Mesh. The value for this configuration is generally **5672** for connecting over AMQP or **5671** for a secure connection over AMQPS (SSL). ```spring:\n rabbitmq:\n port: 5671```
username		Username for connecting to the Perspectium Integration Mesh. ```spring:\n rabbitmq:\n username: testuser```
password		Password for connecting to the Perspectium Integration Mesh. ```spring:\n rabbitmq:\n password: password123```

vhost		Your virtual space (vhost) in the Perspectium Integration Mesh. This value will generally be provided to you by Perspectium Support.
		
		``
		spring:
		    rabbitmq:
		        vhost: /
		``

exclu sive	true	Sets the meshlet to be the exclusive consumer of messages in the queue specified in the below **perspectium.message. inboundQueue** configuration. When this value is true, any other meshlet that tries to connect to the queue will be denied access until this meshlet is no longer connected. Using this value as true prevents other meshlets (such as a test or dev meshlet) from consuming messages on accident.
		
		This can also be useful for high availability when you are running two meshlets at once. The first meshlet ran will continue to access and consume the queue while the second meshlet will continually retry every 30 seconds. If the first meshlet should die and lose connection, then the second meshlet will start connecting to consume messages.
		
		``
		spring:
		    rabbitmq:
		        exclusive: true
		``

ssl:
 	
ena bled
	false	Enables a secure connection over AMQPS (SSL) to RabbitMQ. The **spring.rabbitmq.port** configuration above will also need to be an AMQPS port that accepts secure connections.
		
		``
		spring:
		    rabbitmq:
		        ssl:
		            enabled: true
		``

Perspectium - Message Configuration

To configure Perspectium-related properties including which queues to connect to in the Integration Mesh, you will need to include the following configurations under **perspectium.message**:

Key	Default Value	Description

inbou ndQu eue		Represents the queue in the Integration Mesh that the meshlet will consume messages from.
		
		``
		perspectium:
		    message:
		        inboundQueue: psp.in.meshlet.servicenow.incident.psp_demo
		``
		
		Queues are generally in the following format:
		DataSync:
		
		``
		psp.in.meshlet.<application>.<customer/key>
		``
		
		Example: psp.in.meshlet.servicenow.psp_demo.
		ServiceBond:
		
		``
		psp.in.meshlet.<application>.<process>.<customer/key>
		``
		
		Example: psp.in.meshlet.servicenow.incident.psp_demo.
		For more information on how to name your queues, contact Perspectium Support.

outbo undQ ueue		Represents the queue in the Integration Mesh that the meshlet will be publishing messages to. This is normally a ServiceNow instance, such as another application that is being integrated with ServiceNow sending messages (records) to the instance.
		
		``
		perspectium:
		    message:
		        inboundQueue: psp.out.servicenow.dev123456
		``

wrap CDATA	false	Enabling this configuration will wrap all outbound message fields in a CDATA section. This is useful for when you have a ServiceBond integration with an API that receives XML payloads.
		
		``
		perspectium:
		    message:
		        wrapCDATA: false
		``

insta nceK ey		Enabling this feature will limit the meshlet to only process messages that have a **key** matching the value specified in this configuration. Perspectium applications generate messages that include a **key** field that represents the key of where the messages was generated from. In the case of ServiceNow, this is the instance name i.e. **dev12345**.
		
		With this feature, the meshlet will only process messages with the matching key and skip all other messages. This allows you to filter out any messages so the meshlet is reading messages from only one source.
		
		``
		perspectium:
		    message:
		        instanceKey: dev12345
		``
		
		If this configuration is not specified, the meshlet will process all messages it consumes from its inbound queue.

Example of a meshlet configuration yaml file

The example below shows a basic configuration of the meshlet configuration yaml file (i.e. application.yml or application-dev.yml).

```yaml
spring:
   rabbitmq:
      host: localhost
      port: 5672
      username: georvioloan
      password: password123
      vhost: /
      exclusive: false
perspectium:
   message:
      inboundQueue: psp.in.meshlet.servicenow.incident.psp_demo
      outboundQueue: psp.out.servicenow.dev123456
      wrapCDATA: false
```