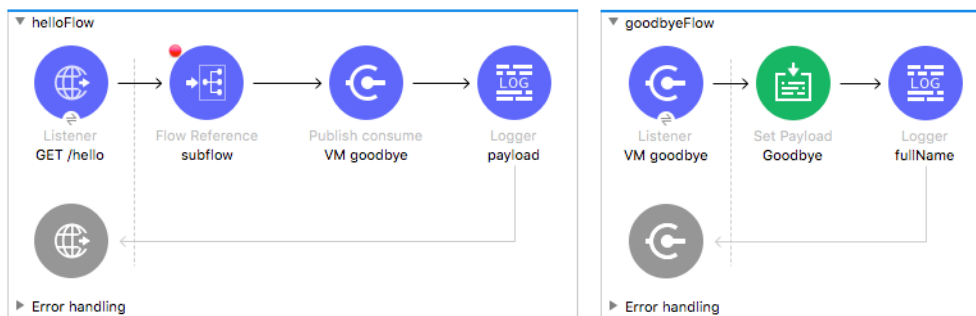


Walkthrough 7-2: Pass messages between flows using the VM connector

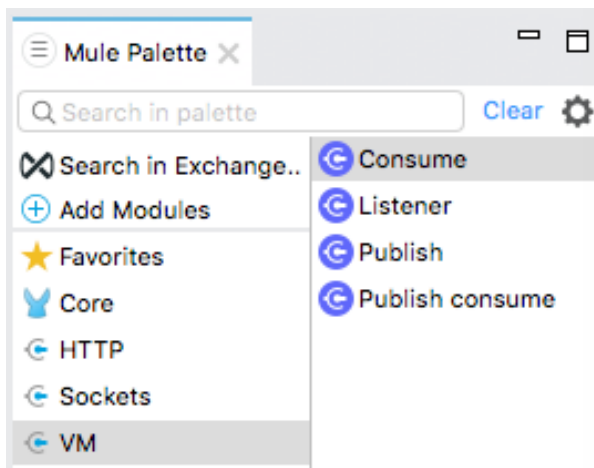
In this walkthrough, you pass messages between flows using asynchronous queues. You will:

- Pass messages between flows using the VM connector.
- Explore variable persistence with VM communication.
- Publish content to a VM queue and then wait for a response.
- Publish content to a VM queue without waiting for a response.

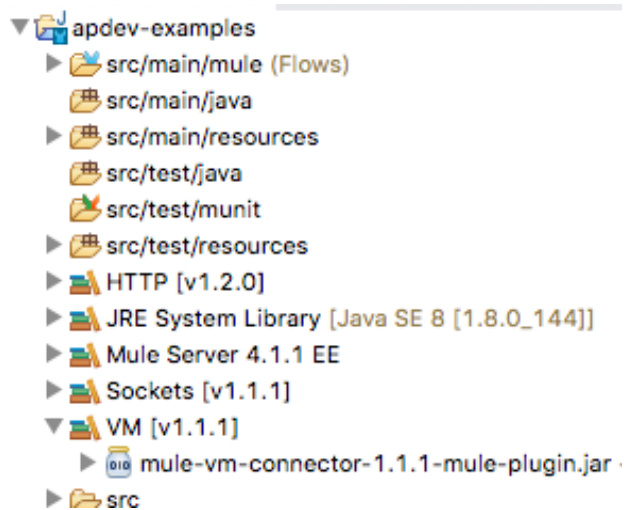


Add the VM module to the project

1. Return to apdev-examples.xml.
2. In the Mule Palette, select Add Modules.
3. Select the VM connector in the right side of the Mule Palette and drag and drop it into the left side.

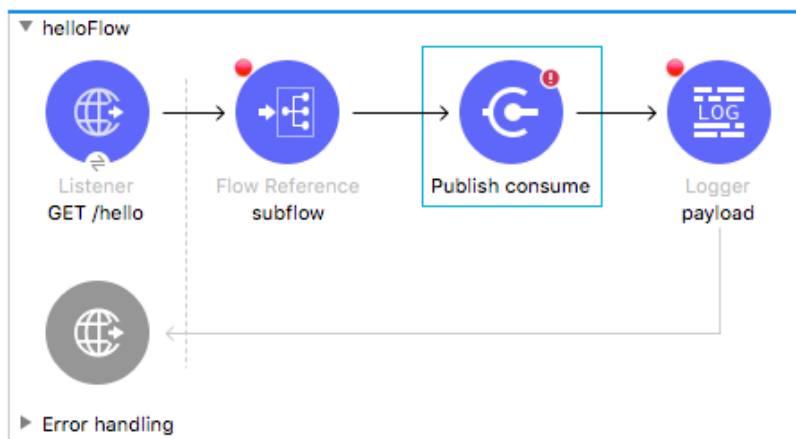


4. In the Project Explorer, locate the JAR file for the VM connector.

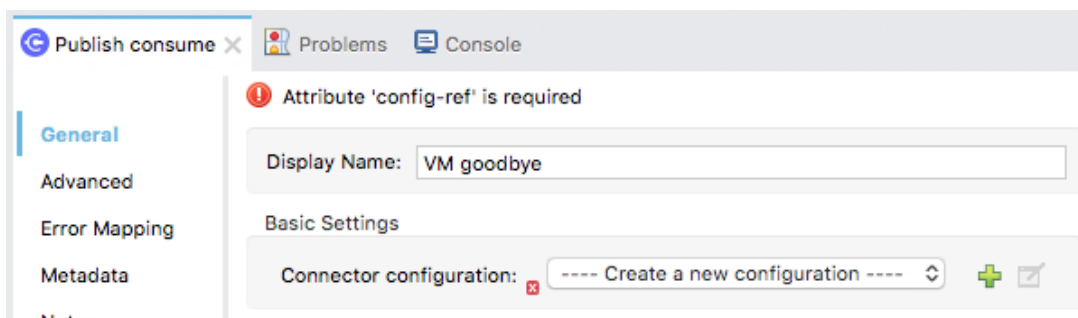


Add a VM Publish Consume operation

5. In helloFlow, delete the privateFlow Flow Reference.
6. Select VM in the Mule Palette.
7. Select the Publish consume operation and drag and drop it before the Logger in helloFlow.



8. In the Publish consume properties view, change the display name to VM goodbye.



9. Click the Add button next to connector configuration.
10. In the Global Element Properties dialog box, click the Add Queue button.

General

Queues

+ [icon] [icon] [icon] [icon]

Queue name	Queue type	Max outstanding messages
------------	------------	--------------------------

Send correlation id: AUTO (Default)

11. In the Queue dialog box, set the queue name to goodbye and click Finish.

Queue

Queue name: goodbye

Queue type: TRANSIENT (Default)

Max outstanding messages: 0

? Cancel Finish

12. In the Global Element Properties dialog box, click OK.

Global Element Properties

VM Config

Default configuration

General Advanced Notes

Name: VM_Config

Advanced

Connection

☐ Reconnection

Fails deployment when test connection fails: False (Default)

Reconnection strategy: None

General

Queues

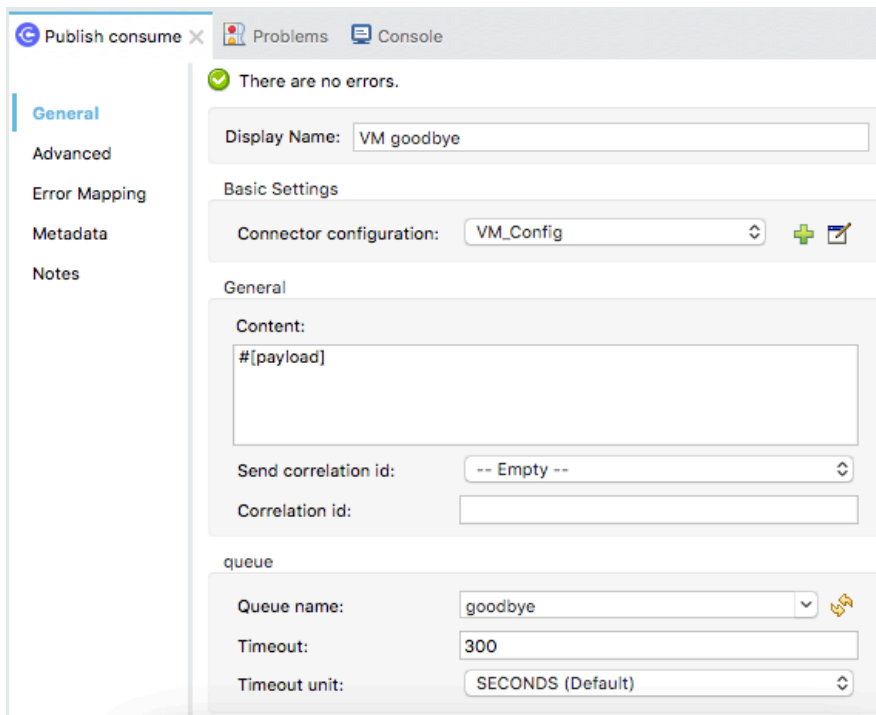
+ [icon] [icon] [icon] [icon]

Queue name	Queue type	Max outstanding messages
goodbye	TRANSIENT	0

Send correlation id: AUTO (Default)

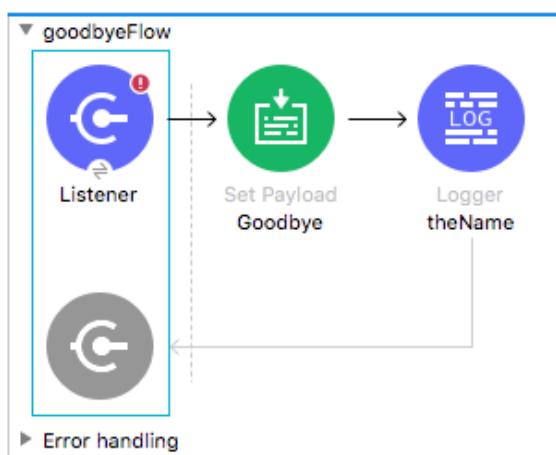
? Cancel OK

13. In the VM goodbye Publish consume properties view, set the queue name to goodbye.
14. Set the timeout to 300 seconds for debugging purposes.



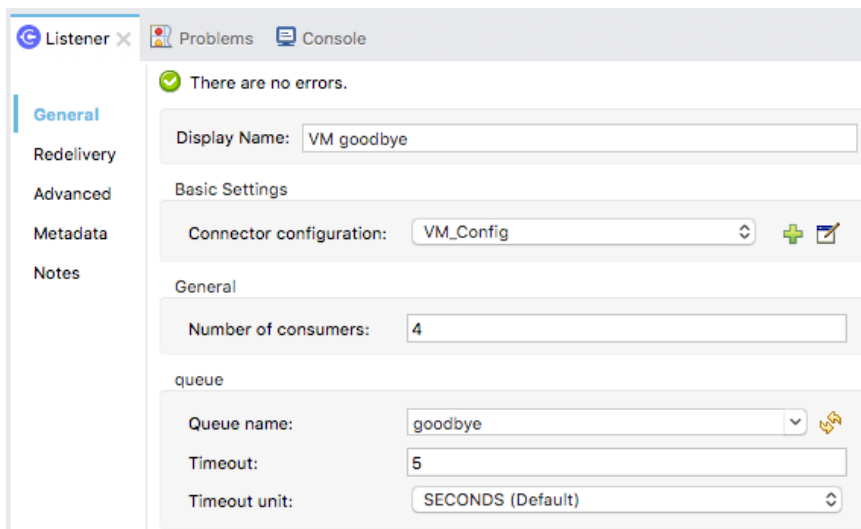
Add a VM Listener

15. Expand goodbyeFlow and delete its HTTP Listener.
16. Locate the Listener operation for the VM connector in the right side of the Mule Palette and drag and drop it in the source section of goodbyeFlow.



17. In the VM Listener properties view, change the display name to VM goodbye.
18. Set the connector configuration to the existing VM_Config

19. Set the queue name to goodbye.



Listener x Problems Console

General

Redelivery

Advanced

Metadata

Notes

There are no errors.

Display Name: VM goodbye

Basic Settings

Connector configuration: VM_Config

General

Number of consumers: 4

queue

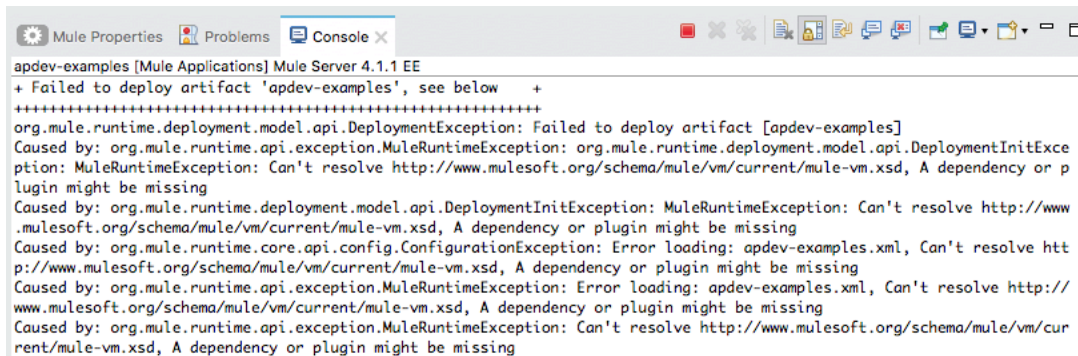
Queue name: goodbye

Timeout: 5

Timeout unit: SECONDS (Default)

Debug the application

20. Save the file to redeploy the project in debug mode; you should get an error that a dependency is missing.



```
apdev-examples [Mule Applications] Mule Server 4.1.1 EE
+ Failed to deploy artifact 'apdev-examples', see below +
+++++
org.mule.runtime.deployment.model.api.DeploymentException: Failed to deploy artifact [apdev-examples]
Caused by: org.mule.runtime.api.exception.MuleRuntimeException: org.mule.runtime.deployment.model.api.DeploymentInitException: MuleRuntimeException: Can't resolve http://www.mulesoft.org/schema/mule/vm/current/mule-vm.xsd, A dependency or plugin might be missing
Caused by: org.mule.runtime.deployment.model.api.DeploymentInitException: MuleRuntimeException: Can't resolve http://www.mulesoft.org/schema/mule/vm/current/mule-vm.xsd, A dependency or plugin might be missing
Caused by: org.mule.runtime.core.api.config.ConfigurationException: Error loading: apdev-examples.xml, Can't resolve http://www.mulesoft.org/schema/mule/vm/current/mule-vm.xsd, A dependency or plugin might be missing
Caused by: org.mule.runtime.api.exception.MuleRuntimeException: Error loading: apdev-examples.xml, Can't resolve http://www.mulesoft.org/schema/mule/vm/current/mule-vm.xsd, A dependency or plugin might be missing
Caused by: org.mule.runtime.api.exception.MuleRuntimeException: Can't resolve http://www.mulesoft.org/schema/mule/vm/current/mule-vm.xsd, A dependency or plugin might be missing
```

21. Stop the project.

22. Debug the project; the application should successfully deploy.

23. In Advanced REST Client, send the same request.

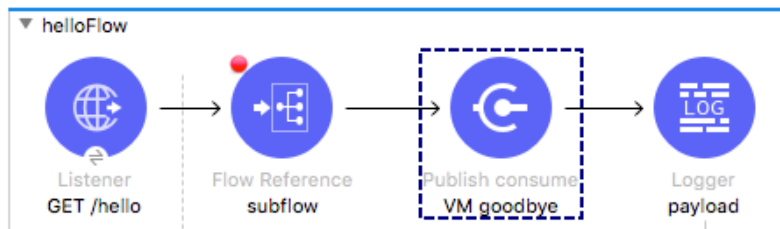
24. In the Mule Debugger, step through the application to the VM Publish consume.

25. Look at the payload, attributes, and variables.

Mule Debugger X

Name	Value
▶ Attributes	org.mule.extension.http.api.HttpRequestAttributes
▶ Component Path	helloFlow/processors/1
▶ DataType	SimpleDataType{type=java.lang.String, mimeType=**/*}
▶ Message	
▶ Payload (mimeType=**/*)	Hello
▶ Variables	size = 1
▶ 0	firstName=Maxwell

apdev-examples X



26. Step into goodbyeFlow.

27. Look at the payload, attributes, and variables (or lack thereof).

Mule Debugger X

Name	Value
▶ Attributes	org.mule.extensions.vm.api.VMM
▶ correlationId	0-dad4f120-4424-11e8-9298
▶ queueName	goodbye
▶ serialVersionUID	3923721884012142263
▶ timestamp	2018-04-19T15:57:41.030
▶ Component Path	goodbyeFlow/processors/0
▶ DataType	SimpleDataType{type=java.lang.
▶ Message	
▶ Payload (mimeType="... Hello	

apdev-examples X

goodbyeFlow

```
graph LR; Listener[Listener VM goodbye] --> SetPayload[Set Payload Goodbye]; SetPayload --> Logger[Logger theName];
```

The diagram shows a flow named 'goodbyeFlow'. It starts with a 'Listener VM goodbye' component, followed by a 'Set Payload Goodbye' component. The 'Set Payload Goodbye' component is highlighted with a dashed blue box. The flow ends with a 'Logger theName' component.

28. Step through the flow until the event returns to helloFlow.
29. Look at the payload, attributes, and variables.

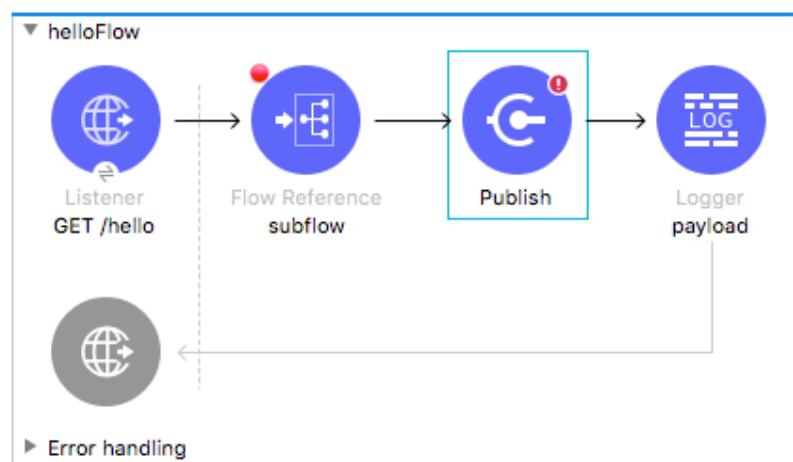
The screenshot shows the Mule Debugger interface with the 'Mule Debugger' tab active. Below it, the 'apdev-examples' application is open, showing the 'helloFlow' flow. The flow consists of four steps: a Listener (GET /hello), a Flow Reference (subflow), a Publish consume (VM goodbye), and a Logger (payload). The Logger step is highlighted with a dashed blue box. The Mule Debugger window shows the following details:

Name	Value
Attributes	org.mule.extensions.vm.api.VMMessageAttribu
Component Path	helloFlow/processors/2
DataType	SimpleDataType{type=java.lang.String, mimeT
Encoding	UTF-8
Message	
Payload (mimeType="applica...	GOODBYE Maxine
Variables	size = 1
0	firstName=Maxwell

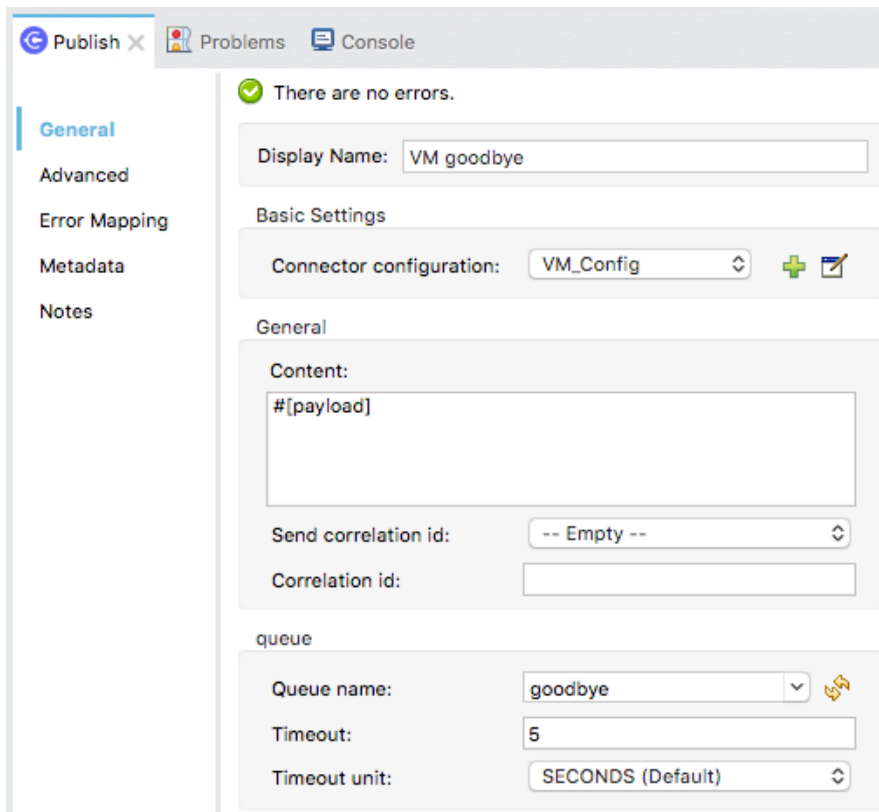
30. Step through the rest of the application.
31. Switch perspectives.

Change the VM Publish Consume operation to Publish

32. Delete the VM Publish consume operation in helloFlow.
33. Drag a VM Publish operation from the Mule Palette and drop it before the Logger.



34. In the Publish properties view, set the display name to VM goodbye.
35. Set the connector configuration to the existing VM_Config.
36. Set the queue name to goodbye.



Debug the application

37. Save the file to redeploy the project in debug mode.
38. In Advanced REST Client, send the same request.
39. In the Mule Debugger, step through the application to the VM Publish operation.
40. Step again; you should step to the Logger in helloFlow.

41. Look at the value of the payload.

The screenshot shows two panels. The top panel, titled 'Mule Debugger', displays a table of message details:

Name	Value
Attributes	org.mule.extension.http.api.HttpRequestAttributes
Component Path	helloFlow/processors/2
DataType	SimpleDataType{type=java.lang.String, mimeType=('*/*')}
Message	
Payload (mimeType=('*/*'))	Hello
Variables	size = 1

Below the table, the payload 'Hello' is displayed. The bottom panel, titled 'apdev-examples', shows a flow diagram with four components: 'Listener GET /hello', 'Flow Reference subflow', 'Publish VM goodbye', and 'Logger payload'. The 'Logger payload' component is highlighted with a dashed blue box.

42. Step again; you should see execution stopped in goodbyeFlow.

43. Step to the end of the application.

44. Return to Advanced REST Client; you should see a response of Hello – not GOODBYE.

The screenshot shows the Advanced REST Client interface. At the top, a green status bar indicates '200 Success' and '160792.30 ms'. Below this, there are icons for copy, download, source code, and view. The response body displays the text 'Hello'.

45. Return to Anypoint Studio and switch perspectives.

46. Stop the project.