

3.4. Exchanging messages with other JBI services

{scrollbar}

On this page, we are going to INLINE

reconfigure Camel to send messages to the JBI endpoints we've just created

Modifying MyCamelRoute class

If you look at the code below, it might come as no surprise that the only thing you need to know to interact with the JBI services is a specific URI syntax to refer to these services.

```
public class MyRouteBuilder extends RouteBuilder { public void configure() { //send a message to a JBI endpoint... from("timer://tutorial?
fixedRate=true&delay=3000&period=10000") // 1 .setBody(constant("<message>Hello world!</message>")) // 2 .to("jbi:endpoint:urn:org:apache:servicemix:
tutorial:camel:jms:provider"); // 3 //...and receive messages sent by other JBI endpoints from("jbi:endpoint:urn:org:apache:servicemix:tutorial:camel:jms:
consumer") // 1 .to("log:tutorial-jbi") // 2 .convertBodyTo(DOMSource.class) // 3 .to("log:tutorial-domsouce") // 2 .convertBodyTo(String.class) // 3 .to("log:
tutorial-string"); // 2 } }
```

Send a message to a JBI endpoint

Let's first look in a bit more detail at the Camel route that sends data to the JBI endpoint:

1. we already know the `timer://...` endpoint from the previous pages
2. since JBI uses normalized (read: XML) messages, we add a start and end tag to the "Hello world!" message body
3. we refer to the JBI endpoint with `jbi:endpoint:<namespace>:<service>:<endpoint>`, so for our JMS consumer endpoint this becomes `jbi:endpoint:urn:org:apache:servicemix:tutorial:camel:jms:provider`

Namespace separator

If you specify your namespace as `xmlns:ns="http://servicemix.apache.org/tutorial/camel"` instead of using the `urn:-style` we are using in this tutorial, this slightly changes the JBI URI in Camel as well, as it would use the `/` as a separator. Example: `jbi:endpoint://http://servicemix.apache.org/tutorial/camel/jms/provider`

For more information see the [Camel JBI URI reference](#)

Receive a message from the JBI ESB

1. Whenever a Camel route is started with `from("jbi:endpoint:...")`, this route will publish an additional endpoint on the ESB. The syntax for the JBI endpoint remains the same, so this line will publish an internal endpoint named `consumer` on a service `jms` in namespace `org:apache:servicemix:tutorial` on the ESB. Because this matches the external JMS consumer endpoint we created on the previous page, this route will receive the messages from this external endpoint.
2. There is nothing new about the `log:` endpoints, we just use the Logger component to print out the message to the console.
3. One of the features that allows Camel to easily work with non-normalized message content, is the Type Converter support that is available. It allows you to transform the message content to another type on-demand. These 3 lines in our Camel route show you how easy it is to transform the JBI NormalizedMessage into a DOMSource or String.

Deploy and test the route

Just run the Maven build again and redeploy the JBI SA to test your changes to the Camel RouteBuilder. After deployment, you should start seeing these kind of messages in your ServiceMix console:

```
INFO - tutorial-jbi - Exchange[JbiMessage: org.apache.servicemix.jbi.messaging.NormalizedMessageImpl@bb2118{properties: {}}] INFO - tutorial-
domsource - Exchange[Message: javax.xml.transform.dom.DOMSource@179c8ad] INFO - tutorial-string - Exchange[Message: <message>Hello world!<
/message>]
```

This clearly shows you the Type Converters in action: the same message is printed as the original NormalizedMessage, a DOMSource and a plain String.

Now that we know how to integrate Camel with the JBI ESB, there is something else we can investigate: Camel itself also has a lot of optional components that you can use inside ServiceMix. Let us have a look at how to go about this on [the next page](#).

Further reading

- There is also another variant of the `jbi:` URI, which only specifies the service name instead of service and endpoint. For more information on the JBI endpoint URIs, have a look at <http://activemq.apache.org/camel/jbi.html>
- <http://activemq.apache.org/camel/type-converter.html> provides you with more detailed information on how Type Converters work

{scrollbar}