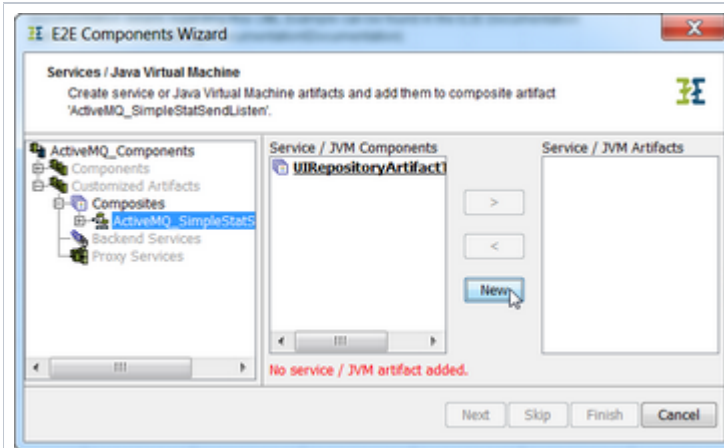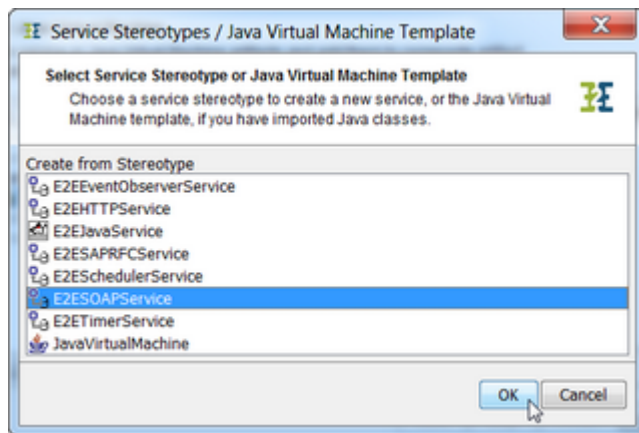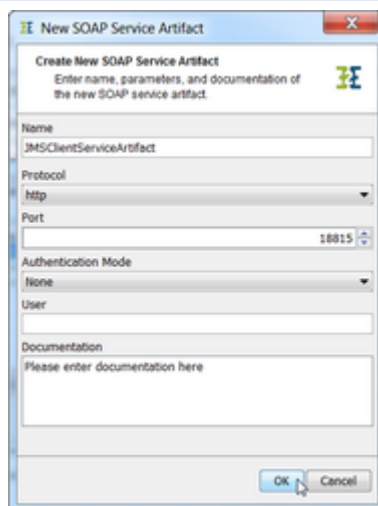# Defining the Components

Having defined the composite artifact (for instance ActiveMQ_SimpleStatSendListen), define a front end SOAP service and the Java Virtual Machine (JVM), that are both resident of the composite.

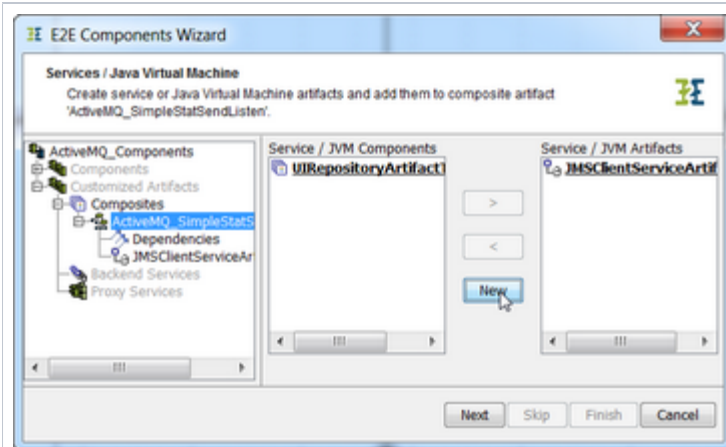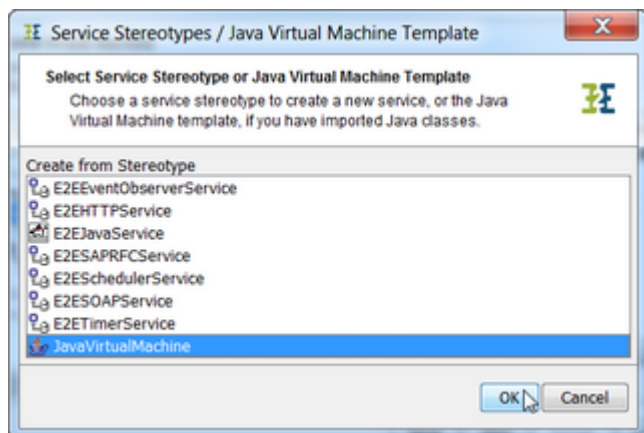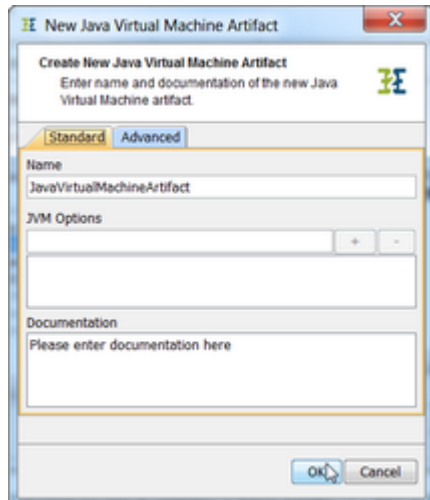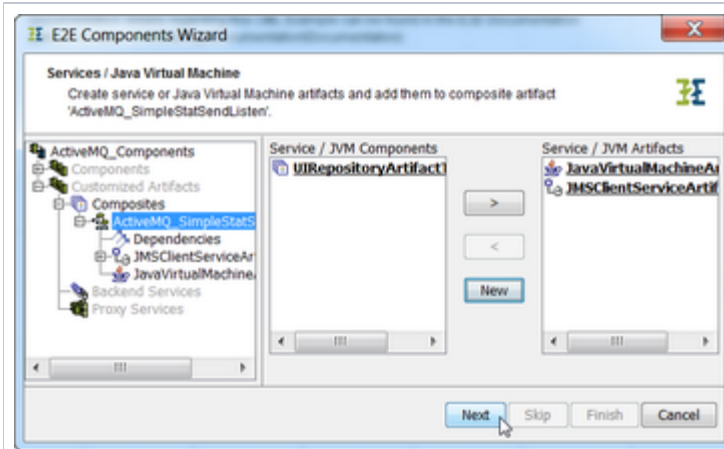|	
	Click New to insert a SOAP service definition.
	Select E2ESOAPService from the drop-down box and click OK.
	Assign a name (e.g. JMSClientService Artifact) and a port number. Click OK.

Click New once again to add the Java VM.

Select JavaVirtualMachine from the drop-down box and click OK.

You may assign a divergent name to the Java VM or just leave it like it is.
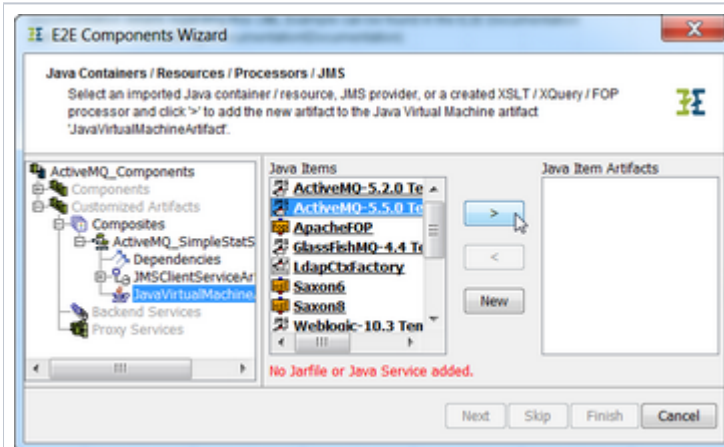
Click OK.

The component Wizard created the JVM artifact and displays it in the right pane.

Click Next to proceed.

After having created an artifact of the JMS client port type definition, navigate to the artifact of the JavaVirtualMachine in the tree panel of the wizard (or click Next two times) to specify the JMS service.
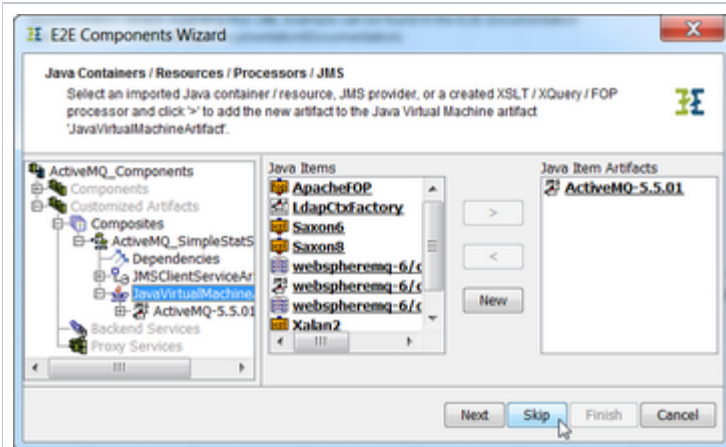

The component wizard lists all JMS service templates (among others).
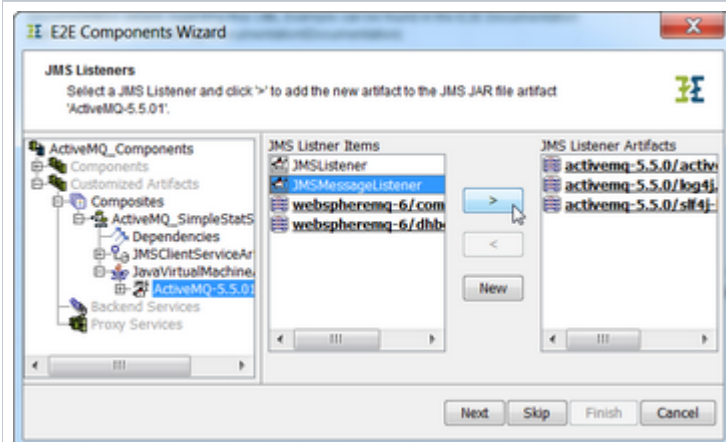
Select a template of the JMS service of your choice (e.g. ActiveMQ-5.5.0_Template) and click > to create an artifact.
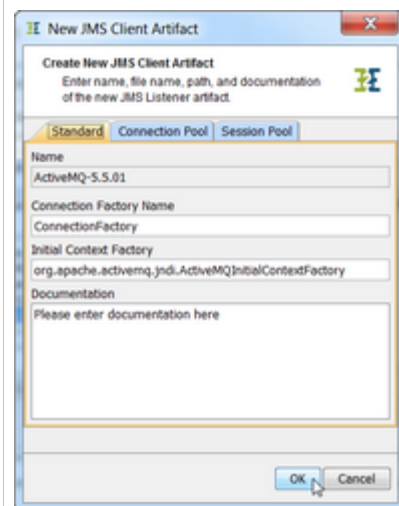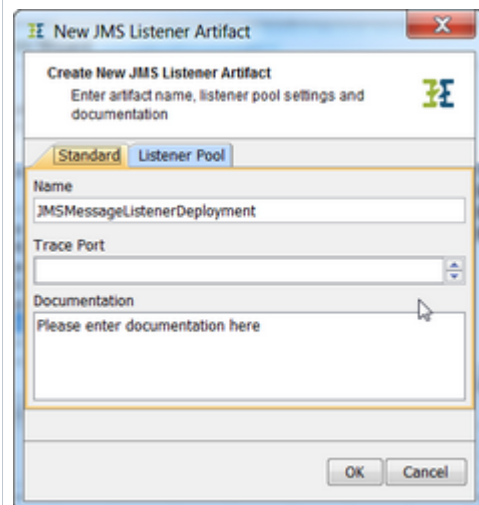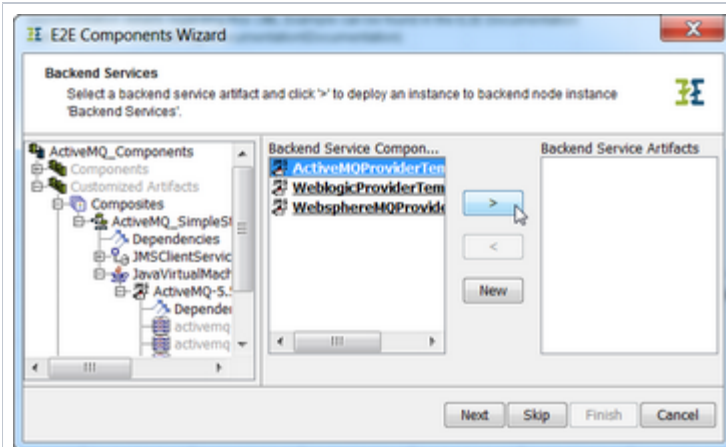

Click OK.

Click Next.

Select the JMS Listener you defined in your model and click > to create an artifact.
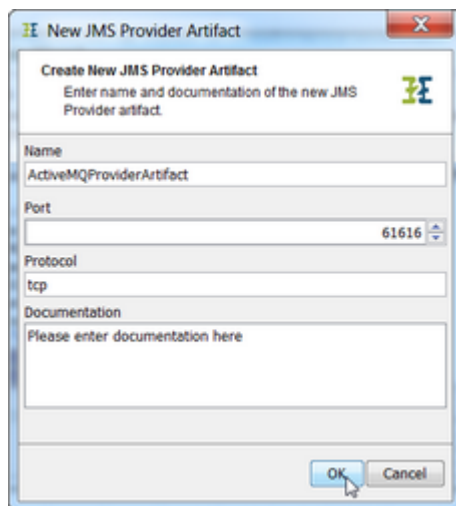

Assign a name or leave the name suggested.

Click OK.

Navigate to the backend node in the tree panel of the wizard (or click Next multiple times until you get there).
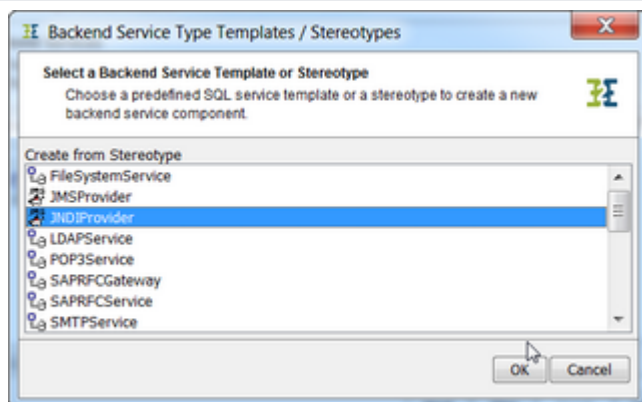
Select the JMS provider template of your choice (e.g. ActiveMQProvider Template) and click > to create an artifact.
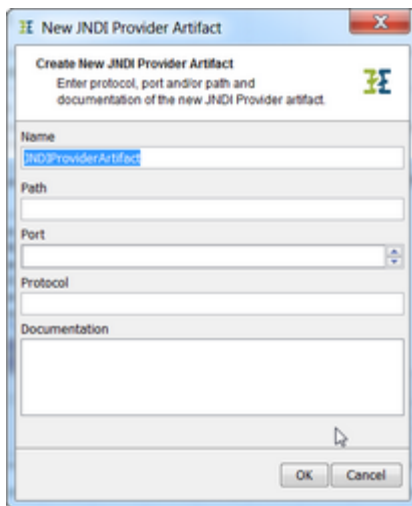

The default values of the parameters correspond to a default installation of the JMS Provider.

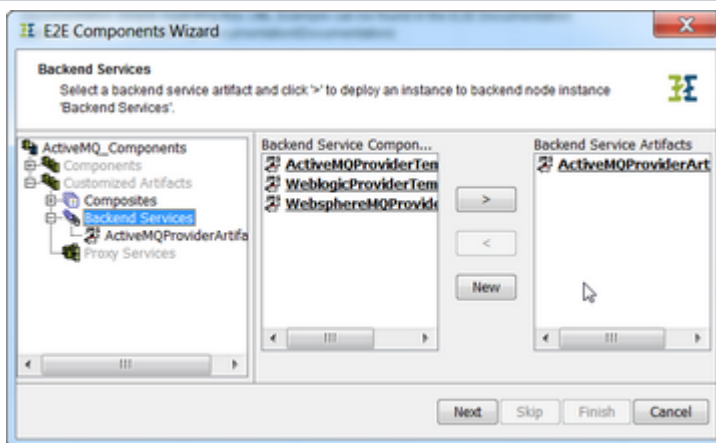Adjust the settings according to your installation.
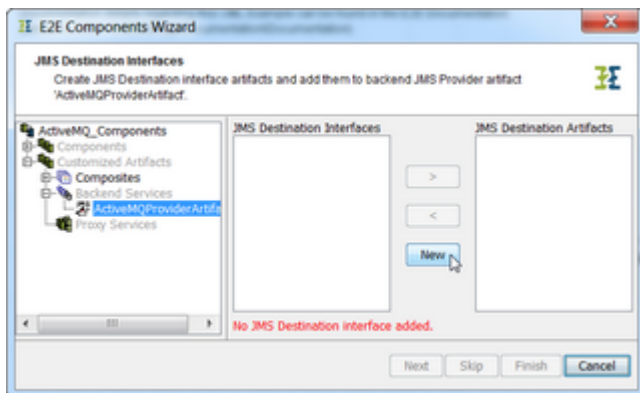
Click OK.

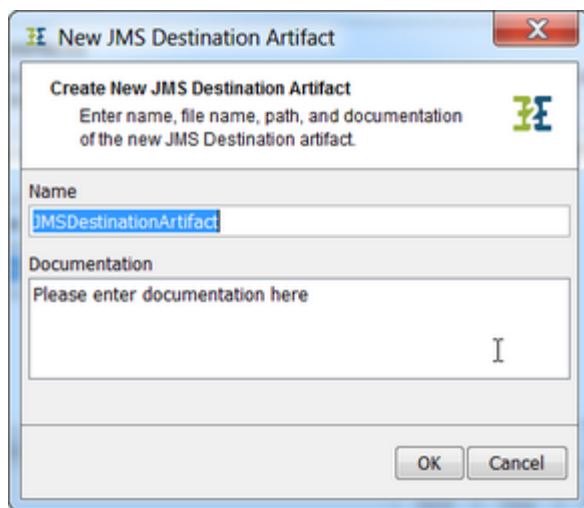Alternatively, click New to define a new JMS Provider (e.g. JNDI Provider).

Not using a template, you have to administer the settings manually.

Click Next.

At this level, Click New to define a JMS destination interface component and artifact.

Assign a name and click OK.
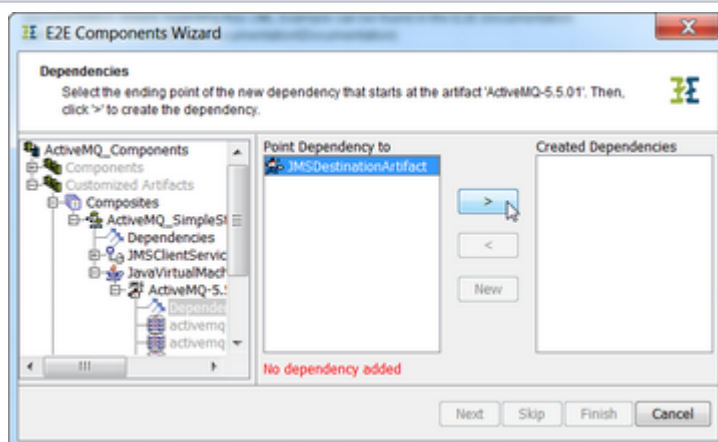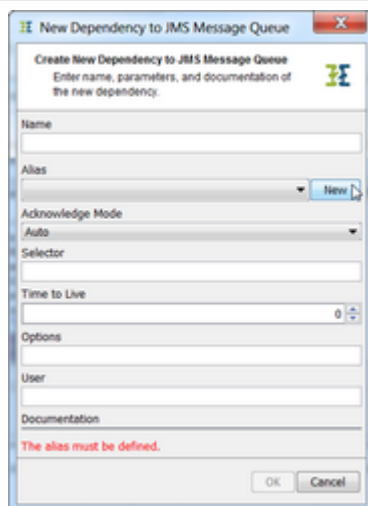

In the tree panel. navigate to dependencies node of the Java Virtual Machine (or click Next multiple times).

Select the JMS destination and click > to create a dependency from the Java Virtual Machine to the JMS destination.

If you have not defined an alias yet, click New to define one.

Otherwise, select the appropriate alias.

New Alias

Enter name and documentation of the new alias.

Name

JMSTopic

Documentation

Please enter documentation here

OK   Cancel

Assign a name an click OK.

New Dependency to JMS Message Queue

Create New Dependency to JMS Message Queue
Enter name, parameters, and documentation of the new dependency.

Name

Alias

JMSTopic   New
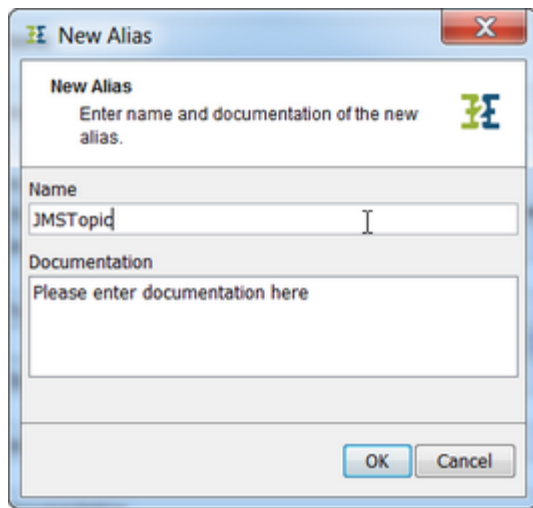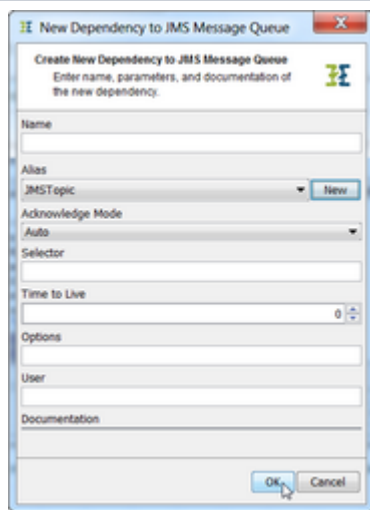
Acknowledge Mode

Auto

Selector

Time to Live

0

Options

User

Documentation

OK   Cancel

Now supply the parameters the JMS Provider needs to feed the listener.

In field Options specify the topic the listener should listen to, in the way <JMSDestination>.topic =<topicName>.

In field User supply the <user>/<password> of the JMS service.

If the JMS client does not listen to all messages of the specified topic, you may enter a Selector, too

---

In the finished component diagram, all these definitions may also be changed directly on the JMS Dependency. **The list of options depends on the JMS provider's requirements.**

For a detailed description of the JMS message header fields and parameters refer to JMS Message Header Fields and JMS Parameters in Detail.

If the JMS service communicates with the message queue dynamically, there is no need to specify the JMS dependency parameters. However, they may be created for documentation purposes
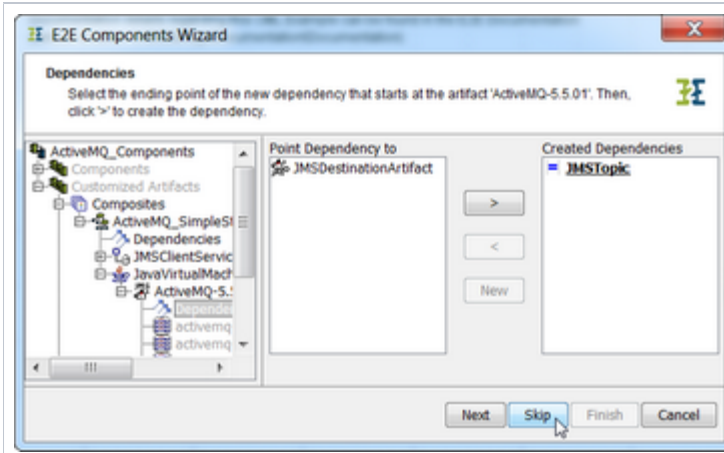
Click Skip as there is no need to make further definitions and finish the component wizard.