

# Creating a Component Diagram

The following sections are ordered according the steps needed to create a component diagram and describe each creation step in detail.

1. [Assigning the Diagram Name](#)
2. [Defining the Composite](#)
3. [Defining the Frontend Service](#)
4. [Defining the SOAP Service Interface](#)
5. [Defining Backend Aliases](#)
6. [Finishing the Components Wizard](#)

## Step 1: Assigning the Diagram Name

After starting the Components Wizard, the first dialog opens, and you are prompted to enter the name of the component diagram.

	<p>You may use the default name suggested or enter a descriptive name. Changing the name will change the name of the top most tree node (marked in blue) in the navigation panel.</p> <p>Click <b>Next</b> to proceed or <b>Cancel</b> to abort.</p>
--	--

The Components Wizard can be canceled any time.

	<p>After clicking cancel, the Components Wizard prompts for confirmation.</p>
--	---

## Step 2: Defining the Composite


With the next step, you define the service composite. The service composite represents the repository of the Web service and will contain all necessary configuration information.

### On this Page:

- [Step 1: Assigning the Diagram Name](#)
- [Step 2: Defining the Composite](#)
- [Step 3: Defining the Frontend Service](#)
- [Step 4: Defining the SOAP Service Interface](#)
- [Step 5: Defining Backend Aliases](#)
- [Step 6: Finishing the Components Wizard](#)

### Related Pages:


- [Defining More Than One Deployment](#)
- [Frontend Components](#)
- [Creating an Alias](#)

«E2EComposite»


**EducationLesson3**

```
{category = "E2E Education",
controlPort = 21111,
killJVMThreadsOnShutdown,
wsdlPerService}
```

Clicking **Next** after having assigned the diagram name will bring you directly to the following dialog. Create the service composite here.


New Composite

**Create New Composite**
Enter name and documentation of the new composite.

Standard
Advanced
Test
Persistent State
JVM
S
◀
▶

Name
EducationLesson3

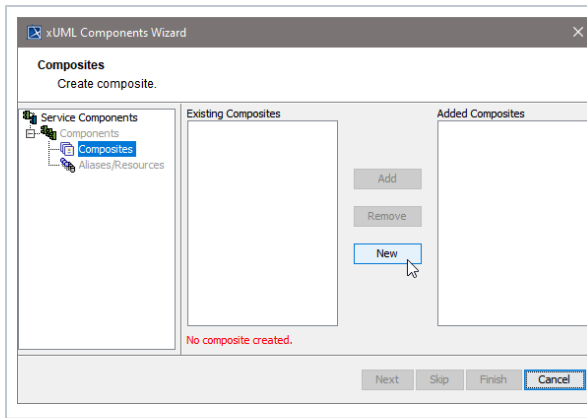
Control Port
21111

Documentation
Please enter documentation here

OK
Cancel

Assign a name to the composite. In the field **Control Port**, enter an unused port number (the value needs to be between 20'000 and 29'999).

Special characters and blanks are not allowed for composite names.



In case you accidentally clicked the **Cancel** button, the **Create new Composite** window can be re-opened by clicking the **New** button.

The **Create New Composite** dialog contains four more tabs.

On the **Advanced** tab of this dialog, you can configure advanced composite settings as tagged values to the composite.

Tagged Value	Description
<b>wsdlPerService</b>	If true (default=false) , each xUML service gets its own WSDL file. Additionally, all XML Schema elements and types having the same namespace are put into one schema file. These schema files are imported into the WSDLs to be shared among them. In this case it is also possible to mix RPC /soap-encoded services with Document /literal services.
<b>WSDL Namespace</b>	Target WSDL namespace of the generated WSDL file. Relevant only, if <b>wsdlPerService</b> is false (this is the default).

<b>Category</b>	Optional category to group similar xUML services.
<b>resolveInheritance</b>	If true, the inheritance hierarchy is resolved into flat messages. As of Bridge 7 , setting <b>resolveInheritance</b> to true is deprecated, because this will generate a different output structure than modeled. It also has hidden requirements to the element uniqueness.
<b>Version</b>	A service version number. This service version is visible in service context on the Bridge.
<b>Soap Version</b>	Specify the version of the SOAP protocol you want to use with the service.
<b>Class To XML Default Root Name</b>	Bridge 7 Specify which name to assign to the XML root element upon serializing. This setting can be overridden by using XML composer options as described on <a href="#">classToXML() Operation</a> . Refer to <a href="#">XML - UML Class Mapping</a> for more information on the topic of XML serialization.

<b>Startup Shutdown Trace Port</b>	Default port for tracing startup or shutdown activities is <b>30000</b> . You can change this default here, if necessary.
------------------------------------	---

For more information on these tagged values refer to [Frontend Components](#).

New Composite

Create New Composite

Enter name and documentation of the new composite.

Standard

Advanced

Test

Persistent State

JVM

S

◀

▶

☰

☐ generateTestService

Test Service Name

Test Port

55555

Test Encoding

rpc/soap

Test Proxy Host

Test Proxy Port

Test Proxy Protocol

https

Test Proxy Path


External Test Proxy Host


OK

Cancel

Specify the test settings here.

Tagged Value	Description
<b>generateTestService</b>	Enables or disables testing.  If true, a test SOAP service will be generated for all <<E2ETestable>> classes.
<b>Test Service Name</b>	Defines the name of the generated test service.
<b>Test Port</b>	Defines the number of the test service port.

<b>Test Encoding</b>	<p>Defines the encoding of the test SOAP operations. rpc/soap is recommended , because this matches well with the semantics of UML operations.</p> <div>  <p>If this value is set and other services of the composite have different encodings , the tag <b>wsdl:Service</b> must be set to true.</p> </div>
<b>Test Proxy Host</b>	Specifies the deployment host.
<b>Test Proxy Port</b>	Specifies the proxy port the test service is listening to.
<b>Test Proxy Protocol</b>	Specifies the used protocol.

<b>Test Proxy Path</b>	<p>Specifies the part of the overall URL that is mapped to the actual host name and port.</p> <div> If this tagged value is used, the modeler must ensure, that this value is unique within the proxy.</div>
<b>External Test Proxy Host</b>	<p>Specifies the host name as seen by the client. If a server certificate is used, it must be issued for this name. If this value is not set, the name of the node hosting the proxy is being used.</p>

For more information on testing in general refer to [xUML Service Model Debugging, Testing and Quality Assurance \(QA\) Concepts](#).

Using Persistent State features in the xUML service, you can make corresponding adjustments on the **Persistent State tab**, and configure the persistent state engine.

Tagged Value	Description
Workers	

New Composite

Create New Composite

Enter name and documentation of the new composite.

Standard

Advanced

Test

Persistent State

JVM

S

◀

▶

☰

Workers

3

Storage Medium

memory

Internal State Database Sync

<UNDEFINED>

External Database Alias

Owner

☐ switchOverEnabled

Switch Over Retry Timeout

600

Event Selection Algorithm

default

External Properties State DB Alias


<UNDEFINED>


OK

Cancel

**Workers** defines the maximum number of parallel threads used to process pending events. The default setting is 5, which is used if none or 0 workers are configured. The implications of more or less workers are discussed in [Performance Considerations of Persistent State](#).



 Each active worker requires one license slot (concurrent connection) to process activities. For more information on licensing and concurrent connections, refer to [License for Running xUML Services](#).

<b>Storage Medium</b>	<p><b>Storage Medium</b> defines where the persisted data and additional information like pending events and current states are kept. See section <a href="#">Data Storage</a> for a discussion of the options. When using a database instance as storage medium, make sure to configure components and deployment of the SQL adapter as described in section <a href="#">SQL Adapter</a>.</p>
<b>Internal State Database Sync</b>	<p><b>Internal State DB Synch</b> defines the level of file system synchronization performed on the internal persistent state database.</p> <div>  For production set up, we recommend using FULL synch. </div>
<b>External Database Alias</b>	<p>If you are using an external storage medium, you can specify the DB alias in <b>External State DB Alias</b>.</p>


<b>Owner</b>	<b>Owner</b> specifies the owner of the persistent state objects. Default owner (used when no divergent owner is specified) is the xUML service name, which is unique on each node instance.
<b>Switch Over Enabled</b>	During fail over, the E2E xUML Runtime will try to create a new database connection to a compensatory database (see <b>Switch Over Enabled</b> ). If this fails, the xUML Runtime will try to open a new connection every second until the timeout (in seconds) is reached. Default is <b>600 seconds</b> .

**Switch  
Over  
Retry  
Timeout**

This flag enables the automatic fail over mechanism for clustered persistent state databases. If the persistent state database becomes inoperative, the E2E xUML Runtime will try to open a connection to compensatory database of the cluster. See also option **Switch Over Retry Timeout**.



This option is available for clustered Oracle databases only.

<b>Event Selection Algorithm</b>	<p>Runtime 2019.2 Builder 7.4.0 The xUML Runtime processes the persistent state events in a defined order. Select the event selection algorithm the xUML Runtime should use to define this order of events. Refer to <a href="#">Performance Considerations of Persistent State &gt; Event Selection Algorithm</a> for more details on when to use which algorithm.</p> <div data-bbox="1075 875 1203 1751"> Services using this tagged value (all options but the default &lt;not specified&gt;) will not start on a Runtime below 2019.2.</div>
----------------------------------	--

<b>External Properties State DB Alias</b>	Runtime 2019.8 Builder 7.6.0 Specify the database alias of the external database you want to store external persistent data to. For more information on external data, refer to <a href="#">Persistent State Classes &gt; External Persistent Data.</a>
---	--

For more information [Persiste  
nt State Components.](#)

New Composite

### Create New Composite

Enter name and documentation of the new composite.

Standard | **Advanced** | Test | Persistent State | JVM | S | ◀ | ▶ | ☰

JVM Options

+

-

☒ killJVMThreadsOnShutdown

Parallel JVM Operations

10
 

↑

↓

OK

Cancel

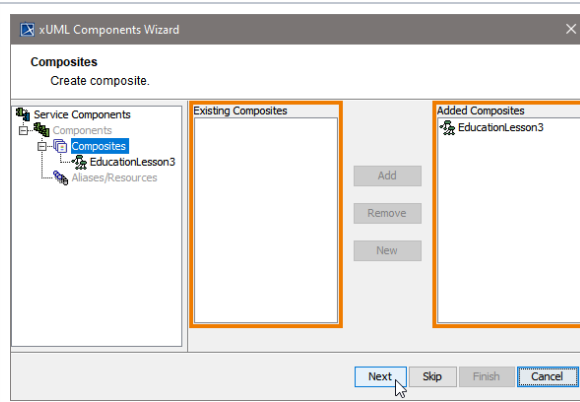
Using Java services in the xUML service, you can specify the JVM options here.

Tagged Value	Description
<b>JVM Options</b>	Java Virtual Machine options. The option strings has one of the following formats: "-D=...", "-X...", "...", The system properties java.class.path and java.library.path are set by the model compiler and can not be overridden in the model.
<b>killJVMThreadsOnShutdown</b>	If threads are still running on shutdown they are killed.
<b>Parallel JVM Operations</b>	Number of parallel open JavaVM threads within the runtime. Default: 100. If the limit is reached, the runtime tries for 60 seconds to obtain a free JVM thread. If it does not succeed, an error (JAVAADLM /19) is thrown.


For more information on Java Components refer to [Modeling the Java Components](#).

Click **OK** to proceed.

Up to this point, you have defined the xUML service component **EducationLesson3**.



In the customization panel, you can find the new xUML service component (e. g. **EducationLesson3**) in the list of added composites on the right side.

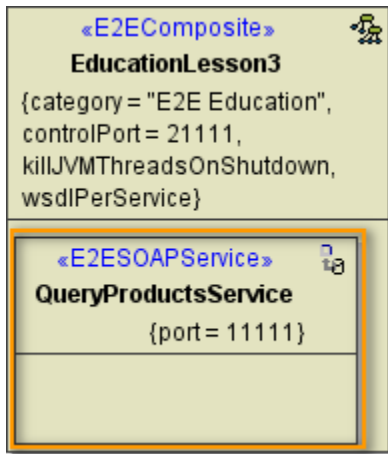
 The left side of the customization panel contains available components, classes, or interfaces. The right side always contains the used components.

Click **Next** to proceed.

## Step 3: Defining the Frontend Service

In the next step, you define the frontend service that will be part of the xUML service.





Clicking **Next** after having defined the composite will bring you directly to the following dialog. Add a frontend service here.

<p><b>Service Stereotypes</b></p> <p><b>Select Service Stereotype</b> Choose a service stereotype to create a new service.</p> <p>Create from Stereotype</p> <ul style="list-style-type: none"> <li>E2EEventObserverService</li> <li>E2EHTTPService</li> <li><b>E2EJavaService</b></li> <li>E2ERESTService</li> <li>E2ESAPRFCService</li> <li>E2ESchedulerService</li> <li>E2ESOAPService</li> <li>E2ETimerService</li> </ul> <p>OK Cancel</p>	<p>Select a service type from the list.</p>
<p><b>xUML Components Wizard</b></p> <p><b>Services</b> Create service and add them to composite 'EducationLesson3':</p> <p>Service Components</p> <ul style="list-style-type: none"> <li>Components</li> <li>Composites</li> <li><b>EducationLesson3</b></li> <li>Aliases/Resources</li> </ul> <p>Existing Services</p> <p>Added Services</p> <p>Add Remove New</p> <p>Next Skip Finish Cancel</p>	<p>In other cases, e.g. if you want to add additional services, you can open the above dialog by clicking <b>New</b> on <b>Services</b> level of the tree.</p>

All possible frontend service stereotypes are listed:

<ul style="list-style-type: none"> <li>E2EEventObserverService</li> <li>E2EHTTPService</li> <li>E2EJavaService</li> <li>E2ESAPRFCService</li> <li>E2ESchedulerService</li> <li>E2ESOAPService</li> <li>E2ETimerService</li> </ul>	<ul style="list-style-type: none"> <li>several ActiveMQ versions</li> <li>GlassFishMQ-4.4</li> <li>Weblogic-10.3</li> <li>WebshpereMQ-6-NoJNDI</li> <li>WebshpereMQ-7</li> </ul>	<ul style="list-style-type: none"> <li>XSLTPProcessor</li> <li>ApacheFOP</li> <li>Saxon6 and 8</li> <li>Xalan2</li> <li>XalanTrax2</li> </ul>	<ul style="list-style-type: none"> <li>UIRepository</li> </ul>
---	--	---	--

Service Stereotypes

### Select Service Stereotype

Choose a service stereotype to create a new service.

Create from Stereotype

- E2EventObserverService
- E2HTTPService
- E2JavaService
- E2RESTService
- E2SAPRFCService
- E2SchedulerService
- E2SOAPService**
- E2TimerService

OK

Cancel

This example shows how to define a Web service. Select the service stereotype **E2ESOAPService**.

Proceed with **OK**.

New SOAP Service

### Create New SOAP Service

Enter name and documentation of the new SOAP service.

Standard

Advanced

Proxy

Name

QueryProductsService

Port

11111

Encoding

<UNDEFINED>

Timezone

Date Format String

Max Request Body Size

2048

Max Parallel Connections

50

☒ resolveHostnames

Documentation

Please enter documentation here

OK

Cancel

Now, customize the service artifact.

Tagged Value	Description
<b>Name</b>	Assign a <b>Name</b> .
<b>Port</b>	Assign a <b>Port</b> , to which the SOAP service is listening (the value needs to be between 10'000 and 19'999).
<b>Encoding</b>	Select an <b>encoding</b> . See <a href="#">Encoding of SOAP Operations</a> for more information on SOAP encoding of xUML services.
<b>Timezone</b>	Time zone string as specified in the <a href="#">time zone appendix</a> . timezone is used to print DateTime expressions.
<b>Date Format String</b>	Date formatting code as listed in <a href="#">Date and Time Formatting</a> . If nothing is defined, the XSD standard is used.

<b>Max Request Body Size</b>	Runtime 2021.2 Specifies the maximum size of the request in KB (1 KB = 1024 Bytes). This can be used to prevent DoS or similar attacks. When the payload of the service exceeds the given maximum, incoming request are rejected.
<b>Max Parallel Connections</b>	Maximum number of requests that can be processed in parallel.
<b>resolveHostnames</b>	Automatically resolve IP addresses to domain names.

The **Create New SOAP Service** dialog contains two more tabs.

On the **Advanced** tab, you can specify the following settings:

Tagged Value	Description
<b>WSDL Namespace</b>	Specifies the target namespace of the generated WSDL file. This is relevant only, if <b>wsdlPerService</b> in the xUML composite is true (see <a href="#">Frontend Components</a> ).
<b>Authentication Mode</b>	

**New SOAP Service**

**Create New SOAP Service**  
Enter name and documentation of the new SOAP service.

Standard **Advanced** Proxy

WSDL Namespace

Authentication Mode  
None

User

OK Cancel

Defines the authentication mode to be used with this SOAP service.

- If HTTPBasicAuth is used, you define that the service composite authenticates the user. User name and password must then be given in the tagged value user. For more details see [Basic Authentication with the E2E Bridge](#).
- If HTTPRemoteAuth is used, a reverse proxy needs to authenticate the user and pass it to the service. The service then expects a set REMOTE-USER HTTP header. Typically the Apache reverse proxy is used for that.

Independent of the authentication mode, the user is found in the principal object if the authentication succeeded.

<b>User</b>	Specifies user and password for the use of HTTP Basic authentication.
-------------	---

For detailed information on these tagged values refer to [S OAP Service Reference](#).

The screenshot shows a dialog box titled "New SOAP Service" with a close button (X) in the top right corner. Below the title bar, there is a section "Create New SOAP Service" with the instruction "Enter name and documentation of the new SOAP service." Below this, there are three tabs: "Standard", "Advanced", and "Proxy". The "Proxy" tab is selected. Under the "Proxy" tab, there are several input fields: "External Proxy Host", "Proxy Host", "Proxy Path", "Proxy Port", and "Proxy Protocol". The "Proxy Protocol" dropdown menu is currently set to "https". At the bottom of the dialog, there are "OK" and "Cancel" buttons.

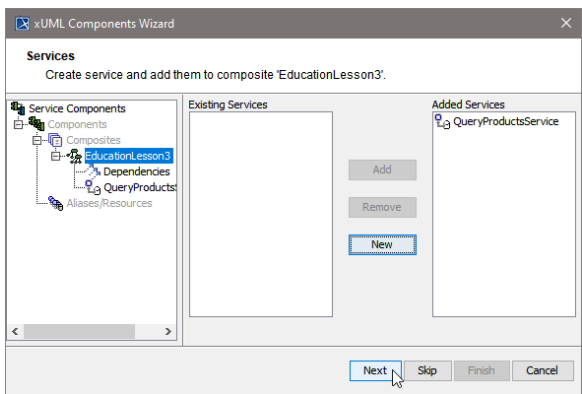
On the **Proxy** tab, you can specify the proxy details.

Tagged Value	Description
<b>External Proxy Host</b>	Specifies the host name as seen by the client. <ul style="list-style-type: none"> <li>If a server certificate is used, the certificate must be issued for this name.</li> <li>If this tagged value is not set, the name of the node hosting the proxy is being used.</li> </ul>
<b>Proxy Host</b>	Specifies the host for the proxy deployment. If <b>localhost</b> is used, the proxy will be created on the same host the service is deployed to. Use a dedicated host, if you want to run your proxy on a different host of the same Bridge domain. See also <b>externalProxyHost</b> .

	<b>Proxy Path</b>	<p>Specifies the part of the overall URL that is mapped to the actual host name and port.</p> <div><p><b>i</b> If <b>proxyPath</b> is used, the modeler must ensure that this value is unique within the proxy.</p></div>
	<b>Proxy Port</b>	<p>Specifies the port the proxy is listening to.</p>
	<b>Proxy Protocol</b>	<p>Specifies the used proxy protocol.</p>
<p>For detailed information on these tagged values refer to <a href="#">SOAP Service Reference</a>.</p>		

Proceed with **OK**.

Now, in the component diagram the service component is placed within the service composite component.

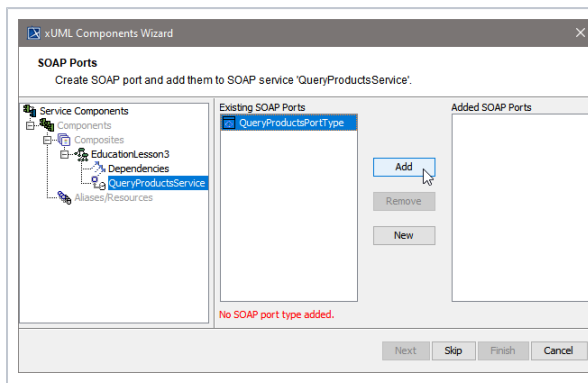
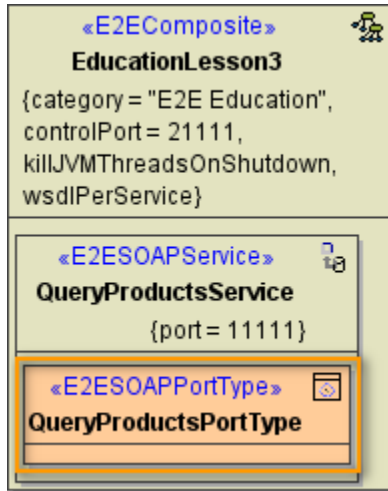


In the customization panel on the right, you will find the new service (e.g. **QueryProductsService**).

Click **Next** to proceed.

## Step 4: Defining the SOAP Service Interface

In the next step, you will define the interface of the SOAP service. Through this interface, the Web service is accessible from the outside world.



The Components Wizard lists all port type classes on the left. Select a port type and click **Add** to add the port type to the composite.

**Edit SOAP Port Type**

Edit name, operation(s) (mandatory), and documentation (mandatory) of the SOAP port type.

Name

QueryProductsPortType

Namespace

Path

Documentation

The port type provides an operation to query an XML file containing product c  
The search result is displayed in the client application.

OK Cancel

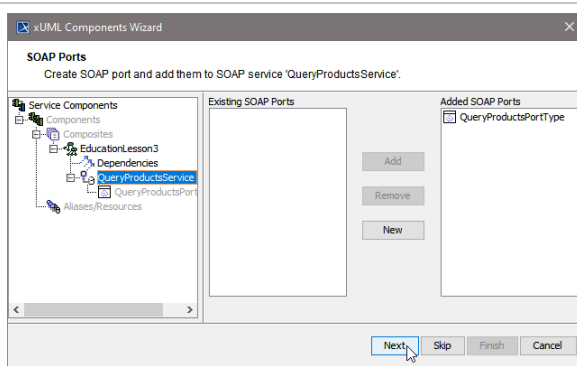
In the following dialog, assign a name to the port type or use the default name suggested.

Special character  
s and  
blanks  
are not  
allowed  
for  
compon  
ent names.

- Provide a **Names**  
**pace** if  
necessar  
y.
- Provide a **path**, if  
you want  
to make  
this  
compon  
ent  
independ  
ent of  
the  
package  
structure.  
Without  
a given  
path,  
port  
types are  
identified  
by port  
type  
name  
and  
package  
URL.

Proceed with  
**OK**.



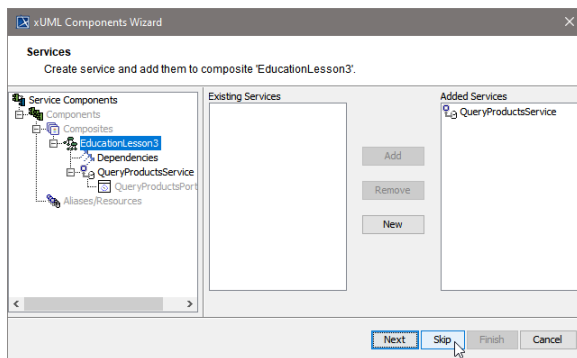


In the artifact part of the customization panel, you will find the added port type **QueryProductsPortType**.

As an interface can only be used once in a composite, the port type **QueryProductsPortType** is not displayed anymore in the SOAP Port Types list on the left.

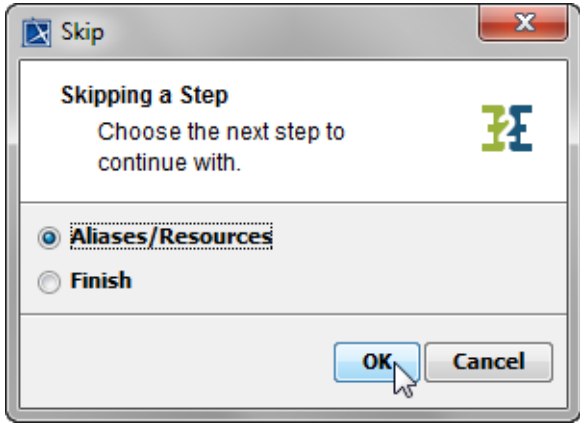
Each port type class (representing the interface of the SOAP service) can only be used once in a composite.

Click **Next** to proceed.



In the **Customized Artifacts** part of the tree panel, the xUML service is selected again to give you the option to define further frontend services (see [Defining the Frontend Service](#)).

If you do not need any further elements, click **Skip**.

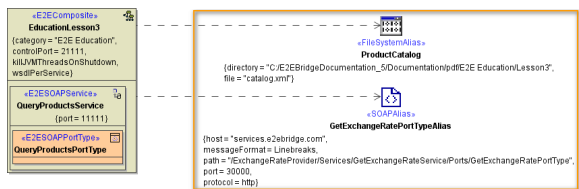


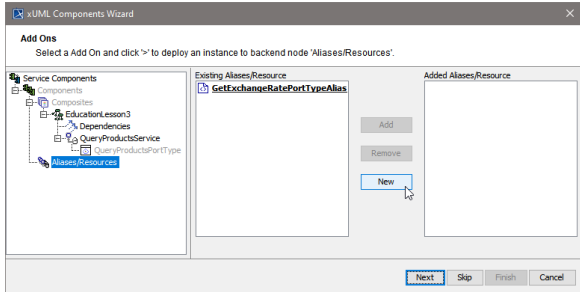
For definition of **Backend Services (Aliases/Resources)** see below ([Defining Backend Aliases](#)).

For **Finishing** the Components Wizard see [Finishing the Components Wizard](#).

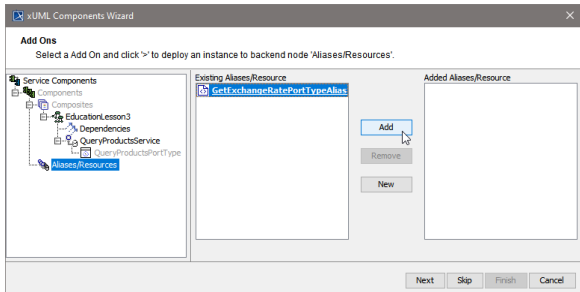
## Step 5: Defining Backend Aliases

In this step, backend aliases are defined. The Components Wizard provides predefined Backend Components.





Click **New** to create a new backend alias ...

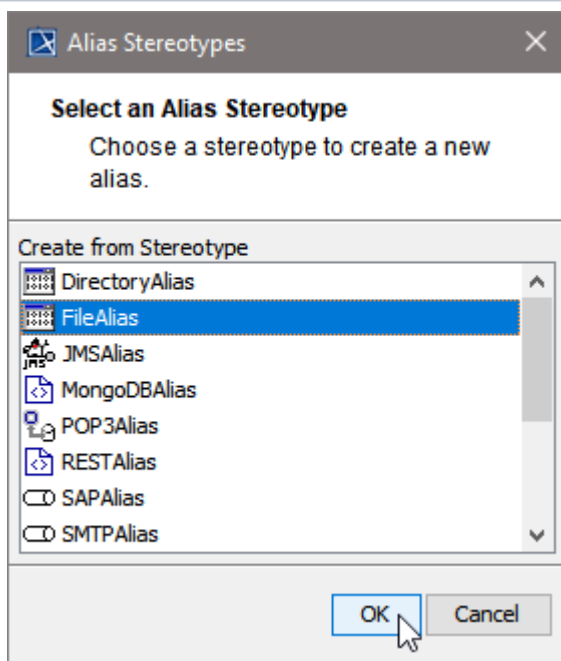


...or select an existing alias and click **Add** to add it to the component diagram.

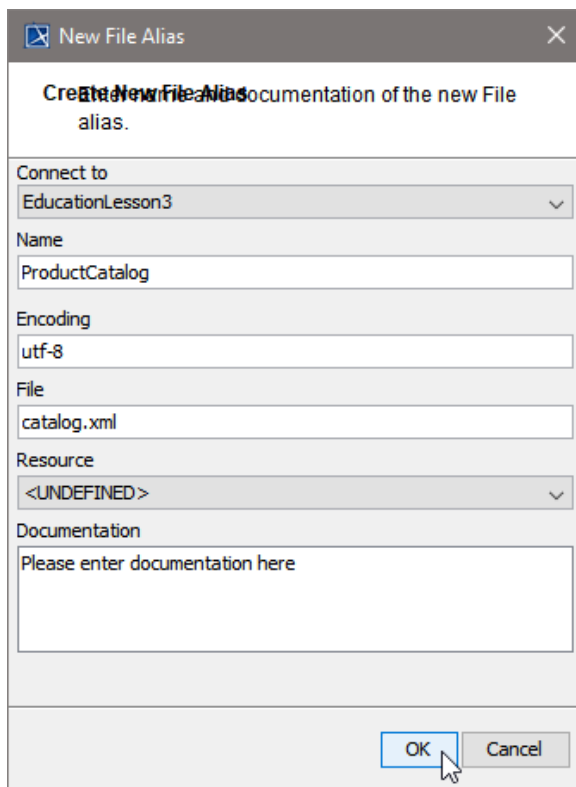
For adding a new alias, choose one of the following possible backend service stereotypes:

Backend Aliases

- FileSystemAlias
- JMSAlias
- POP3Alias
- SAPAlias
- SMTPAlias
- SOAPAlias
- SQLAlias
- URLAlias



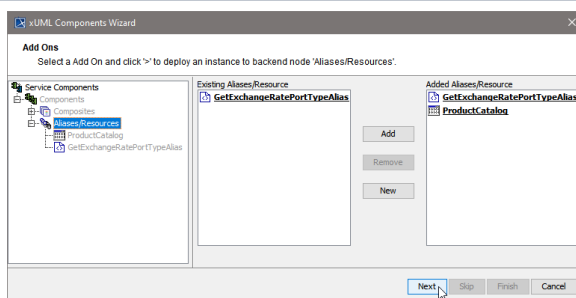
Choose a backend alias stereotype from the list of available stereotypes, e.g. **FileAlias** and click **OK**.



Assign a name to the alias or leave the default name suggested.

Specify **File Name** and **Path**, or, if you wanted to read data from an imported resource, select the resource from the **Resource** drop-down box instead.

Click **OK**.

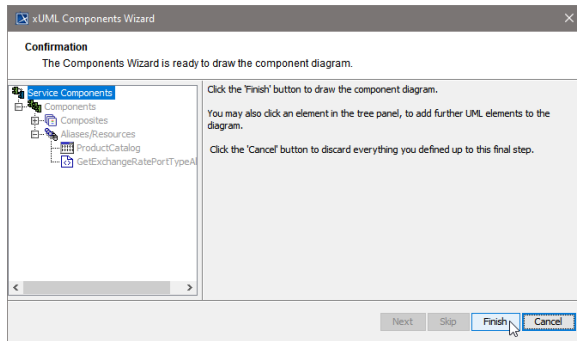


In right part of the customization panel, you will find the new file system alias (e.g. **ProductCatalog**).

Click **Next**.

## Step 6: Finishing the Components Wizard

This is the final step of the Components Wizard where you confirm to draw the component diagram. If the definitions are not complete yet, you can select an element node in the tree panel and add further UML elements to the diagram.



The Components Wizard generates the component diagram. If no errors occurred, it is necessary to save the model. On saving, the deployment data of the Model Compiler gets refreshed and you can proceed with e.g. compiling the model.