Model Documentation

Introduction

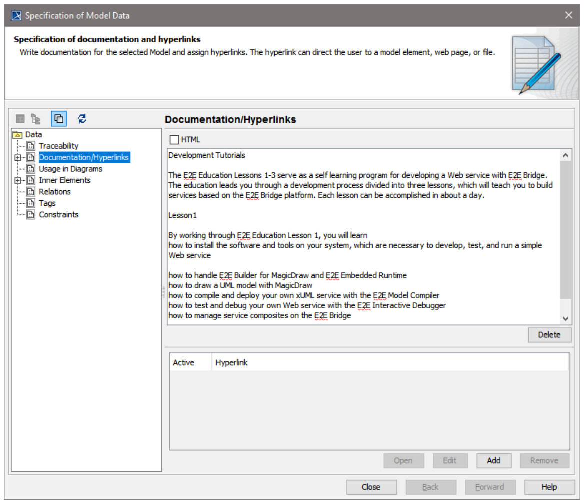
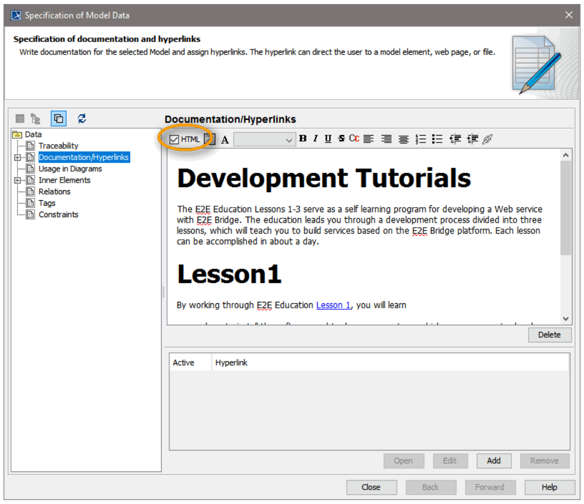
The Builder provides several documentation features to support different kinds of users:

- modelers who want to modify the service and need a quick overview on the service's use cases, backend access and data structures
- service consumers who want to call the service and need information on the service interface
- Bridge operators who need a basic understanding of the service's purpose and features.

For these documentation purposes, the Builder offers the possibility to add documentation and comment tags to model elements.

How to Add a Documentation Tag

Lookup the element you want to add documentation to in the containment tree or diagram and open the specification dialog by right-clicking the element and selecting **Specification** from the context menu.

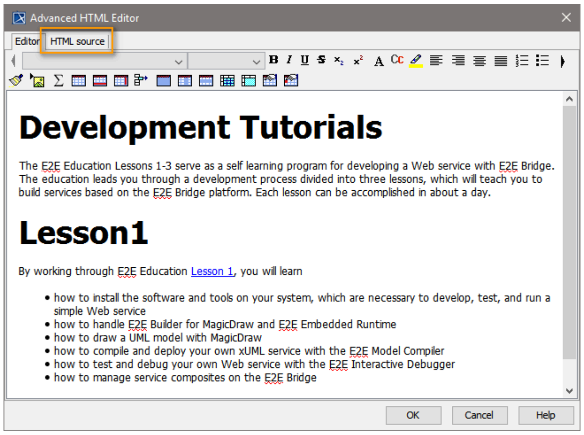
 Specification of Model Data **Specification of documentation and hyperlinks** Write documentation for the selected Model and assign hyperlinks. The hyperlink can direct the user to a model element, web page, or file. Documentation/Hyperlinks □ HTML Development Tutorials The E2E Education Lessons 1-3 serve as a self learning program for developing a Web service with E2E Bridge. The education leads you through a development process divided into three lessons, which will teach you to build services based on the E2E Bridge platform. Each lesson can be accomplished in about a day. Lesson1 By working through E2E Education Lesson 1, you will learn how to install the software and tools on your system, which are necessary to develop, test, and run a simple Web service how to handle E2E Builder for MagicDraw and E2E Embedded Runtime how to draw a UML model with MagicDraw how to compile and deploy your own xUML service with the E2E Model Compiler how to test and debug your own Web service with the E2E Interactive Debugger how to manage service composites on the E2E Bridge Data Traceability Documentation/Hyperlinks Usage in Diagrams Inner Elements Relations Tags Constraints Delete Active Hyperlink Open Edit Add Remove Close Back Forward Help	Click **Documentation/Hyperlinks** and start typing in the documentation field.
 Specification of Model Data **Specification of documentation and hyperlinks** Write documentation for the selected Model and assign hyperlinks. The hyperlink can direct the user to a model element, web page, or file. Documentation/Hyperlinks ☑ HTML **Development Tutorials** The E2E Education Lessons 1-3 serve as a self learning program for developing a Web service with E2E Bridge. The education leads you through a development process divided into three lessons, which will teach you to build services based on the E2E Bridge platform. Each lesson can be accomplished in about a day. **Lesson1** By working through E2E Education Lesson 1, you will learn Data Traceability Documentation/Hyperlinks Usage in Diagrams Inner Elements Relations Tags Constraints Delete Active Hyperlink Open Edit Add Remove Close Back Forward Help	By checking **HTML**, you can also insert HTML formatted documentation.

On this Page:

Related Pages:

- Project Organization
- Naming Conventions and Containment Tree Organisation
- Model Documentation
- Settings
- Mappings
- Sub-activities
- Logging
- Error Handling

- REST Interface Documentation and Tags
- Documentation of an xUML Service



Development Tutorials

The E2E Education Lessons 1-3 serve as a self learning program for developing a Web service with E2E Bridge. The education leads you through a development process divided into three lessons, which will teach you to build services based on the E2E Bridge platform. Each lesson can be accomplished in about a day.

Lesson1

By working through E2E Education Lesson 1, you will learn

- how to install the software and tools on your system, which are necessary to develop, test, and run a simple Web service
- how to handle E2E Builder for MagicDraw and E2E Embedded Runtime
- how to draw a UML model with MagicDraw
- how to compile and deploy your own xUML service with the E2E Model Compiler
- how to test and debug your own Web service with the E2E Interactive Debugger
- how to manage service composites on the E2E Bridge

By clicking , you can open an advanced HTML editor that has more formatting options and also allows to edit the HTML directly.

How to Add Comment Tags

Lookup the element you want to add a comment to in the containment tree and select **Create Element > Comment** from the context menu. Simply start typing to add the comment. Comments can be plain text only.

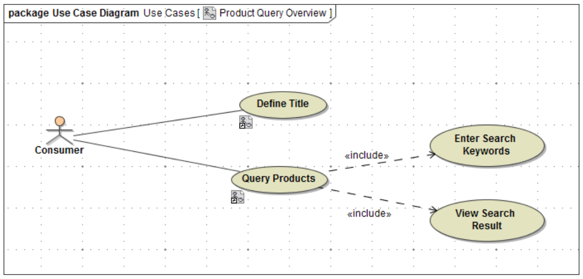
Service Documentation for Modelers

To give modelers a good overview on the service we recommend to

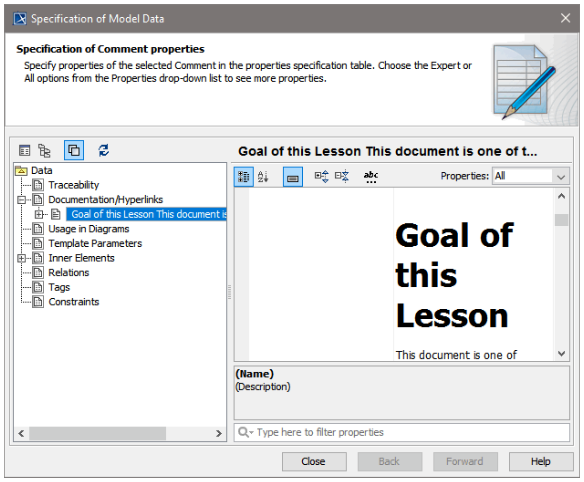
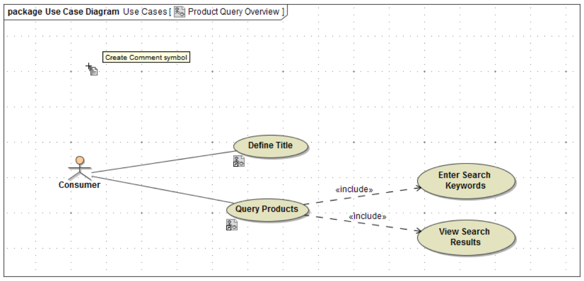
- provide an overview on what the service does by drawing a **use case diagram**
- show the main data structures and relations by drawing a **class diagram**
- give an overview on the main backend processing by drawing a **sequence diagram**
- provide a **change log** in the xml file that is displayed on the Bridge

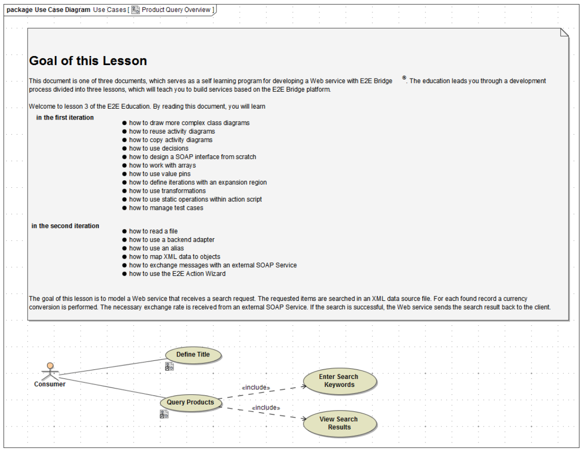
Use Case Diagram

Draw a simple use case diagram describing the main actors and main processing in a package called e. g. **Use Cases**.



We also recommend to make the documentation that previously has been added to the **Data** package visible by adding it to this diagram.
You can do this as follows:

 Specification of Model Data **Specification of Comment properties** Specify properties of the selected Comment in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties. Data Traceability Documentation/Hyperlinks Goal of this Lesson This document is Usage in Diagrams Template Parameters Inner Elements Relations Tags Constraints Goal of this Lesson This document is one of t... Properties: All Goal of this Lesson This document is one of (Name) (Description) Type here to filter properties Close Back Forward Help	Go to the specification dialog of the **D ata** package and select the documentation element.
 package Use Case Diagram Use Cases [Product Query Overview] Create Comment symbol Define Title Consumer Enter Search Keywords «include» Query Products «include» View Search Results	Drag and drop it to the diagram pane.



package Use Case Diagram Use Cases [Product Query Overview]

Goal of this Lesson

This document is one of three documents, which serves as a self learning program for developing a Web service with E2E Bridge ®. The education leads you through a development process divided into three lessons, which will teach you to build services based on the E2E Bridge platform.

Welcome to lesson 3 of the E2E Education. By reading this document, you will learn

in the first iteration

- how to draw more complex class diagrams
- how to reuse activity diagrams
- how to copy activity diagrams
- how to use decisions
- how to design a SOAP interface from scratch
- how to work with arrays
- how to use value pins
- how to define iterations with an expansion region
- how to use transformations
- how to use static operations within action script
- how to manage test cases

in the second iteration

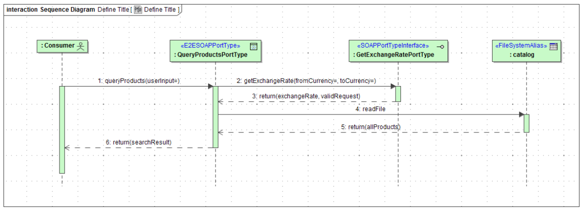
- how to read a file
- how to use a backend adapter
- how to use an alias
- how to map XML data to objects
- how to exchange messages with an external SOAP Service
- how to use the E2E Action Wizard

The goal of this lesson is to model a Web service that receives a search request. The requested items are searched in an XML data source file. For each found record a currency conversion is performed. The necessary exchange rate is received from an external SOAP Service. If the search is successful, the Web service sends the search result back to the client.

You can define to not display stereotype <<documentation>> (like shown above) via the symbol properties from the context menu.

Sequence Diagram

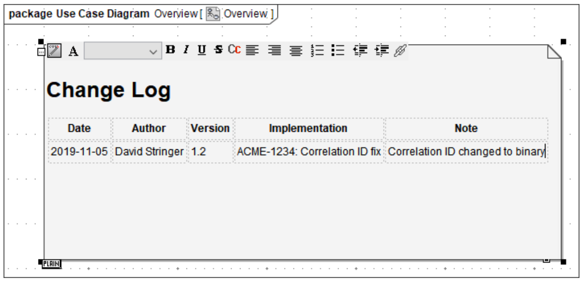
Draw a sequence diagram to give an overview on the main backend processing.



interaction Sequence Diagram Define Title [Define Title]
: Consumer
«E2ESOAPPortType»
: QueryProductsPortType
«SOAPPortTypeInterface»
: GetExchangeRatePortType
«FileSystemAlias»
: catalog
1: queryProducts(userInput=)
2: getExchangeRate(fromCurrency=, toCurrency=)
3: return(exchangeRate, validRequest)
4: readFile
5: return(allProducts)
6: return(searchResult)

Change Log

Keep a change log table in package **Overview**. Builder 7.4.0 The Builder templates already provide a simple change log table in a use case diagram **Overview**.



Change Log

Date	Author	Version	Implementation	Note
2019-11-05	David Stringer	1.2	ACME-1234: Correlation ID fix	Correlation ID changed to binary

All information provided here will be visible in the service documentation on the Bridge for Bridge operators to see (see Service Documentation for Bridge Operators below).

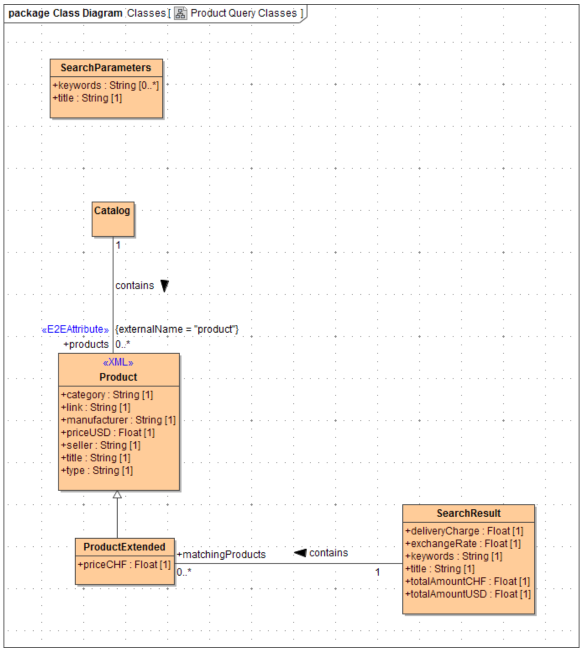
If this default change log table does not suit your needs, you can change it for dedicated models, or you can change the template itself. By changing the template, you can get consistent change logs throughout all services.
To edit the template, open file **<your user directory>/.e2ebuilder/modelDocumentationTemplate.html** with an editor. This file contains the HTML layout of the change log table:

```
<html>
    <head>
    </head>
    <body>
        <h2>Change Log</h2>
        <table class="table">
            <tr>
                <th>Date</th>
                <th>Author</th>
                <th>Version</th>
                <th>Implementation</th>
                <th>Note</th>
            </tr>
            <tr>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
            </tr>
        </table>
    </body>
</html>
```

Change the table or adorn the HTML with styling as you need.

Class Diagram

Draw class diagrams for the main structures in dedicated packages (see Containment Tree Organization).



Service Documentation for Service Consumers

MagicDraw provides the possibility to add documentation and comment tags to model elements. Documentation and comment tags added to service model elements (service, operation, parameter) are explaining the service interface. They are meant to be read by the service consumer and will be generated to

- to the YAML file for REST services
  For more information see REST Interface Documentation and Tags.
- the WSDL file for SOAP services

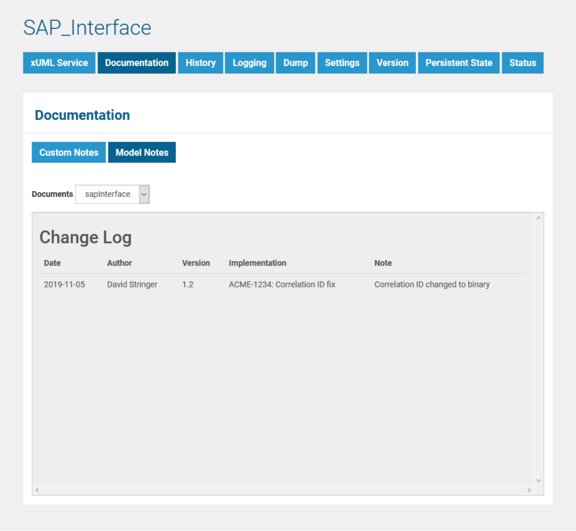
```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="urn:GetTitleService"
                  xmlns:http="http://schemas.xmlsoap.org/soap/http"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:soapenc="http://schemas.xmlsoap.org/soap
/encoding/"
                  name="GetTitleService"
                  targetNamespace="urn:GetTitleService">
   <wsdl:documentation>Documentation of Service GetTitleService<
/wsdl:documentation>
   <wsdl:types/>
   <wsdl:message name="GetTitlePortType_getTitle_Request">
      <wsdl:part name="inputTitle" type="xs:string">
         <wsdl:documentation>&lt;html&gt;
  &lt;head&gt;
        &lt;style&gt;
            p {padding:0px; margin:0px;}
        &lt;/style&gt;
    &lt;/head&gt;
  &lt;body&gt;
    &lt;p&gt;
&lt;b&gt;&lt;span style="font-size:24pt;"&gt;HTML Documentation of
Parameter in&lt;/span&gt;&lt;/b&gt;

    &lt;/p&gt;

&lt;/body&gt;
&lt;/html&gt;</wsdl:documentation>
         <wsdl:documentation>Comment of parameter in</wsdl:
documentation>
      </wsdl:part>
   </wsdl:message>
   <wsdl:message name="GetTitlePortType_getTitle_Response">
```

```
      <wsdl:part name="outputTitle" type="xs:string"/>
   </wsdl:message>
   <wsdl:portType name="GetTitlePortType">
      <wsdl:documentation>The port type provides an operation to
copy a string and convert it to upper case. That string is used as
title in the client application.</wsdl:documentation>
      <wsdl:operation name="getTitle">
         <wsdl:documentation>The operation defines an interface to
copy a simple string, convert it to upper case and return it to the
client application.</wsdl:documentation>
         <wsdl:documentation>Comment on operation GetTitle</wsdl:
documentation>
         <wsdl:input message="tns:GetTitlePortType_getTitle_Request"
/>
         <wsdl:output message="tns:
GetTitlePortType_getTitle_Response"/>
      </wsdl:operation>
   </wsdl:portType>
   <wsdl:binding name="GetTitlePortType_SOAPBinding" type="tns:
GetTitlePortType">
      <soap:binding style="rpc" transport="http://schemas.xmlsoap.org
/soap/http"/>
      <wsdl:operation name="getTitle">
         <soap:operation soapAction=""/>
         <wsdl:input>
            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap
/encoding/"
                       namespace="urn:Services.GetTitleService.Ports.
GetTitlePortType"
                       use="encoded"/>
         </wsdl:input>
         <wsdl:output>
            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap
/encoding/"
                       namespace="urn:Services.GetTitleService.Ports.
GetTitlePortType"
                       use="encoded"/>
         </wsdl:output>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="GetTitleService">
      <wsdl:documentation>Documentation of Service GetTitleService<
/wsdl:documentation>
      <wsdl:documentation>Comment on Service GetTitlService</wsdl:
documentation>
      <wsdl:port binding="tns:GetTitlePortType_SOAPBinding" name="
GetTitlePortType">
         <soap:address location="http://${wsdl_host_urn:
Component_View.Composites.EducationLesson1.GetTitleService}:10011
/GetTitleService/GetTitlePortType"/>
      </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```

Service Documentation for Bridge Operators

Like with all other model elements, you can also add documentation to the **Data** package. This documentation will be available for reading and editing on the Bridge.
See Documentation of an xUML Service for more information on that feature.

If service modelers provided a change log table with their deployed services, this change log will be visible in the **Model Notes** section of the **Documentation** tab.



Change Log

Date	Author	Version	Implementation	Note
2019-11-05	David Stringer	1.2	ACME-1234: Correlation ID fix	Correlation ID changed to binary