

# Creating xUML Service Libraries

This section describes how to define xUML service libraries.

Example File (Builder project Advanced Modeling/E2ELibrary):



<your example path>\Advanced Modeling\E2ELibrary\uml\LibrarySQLQuery.xml

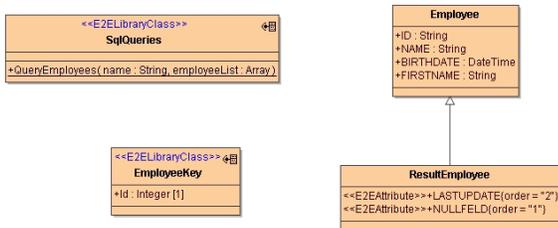
## On this Page:

- [Library Classes](#)
- [Library Components](#)
  - [Library SOAP Service](#)
- [Compiling the Library](#)

## Library Classes

As a first step, the modeler has to define the external usable classifiers explicitly in the model. This is done by giving all classes that shall be reused an `<<E2ELibraryClass>>` stereotype.

Figure: E2ELibrary Classes



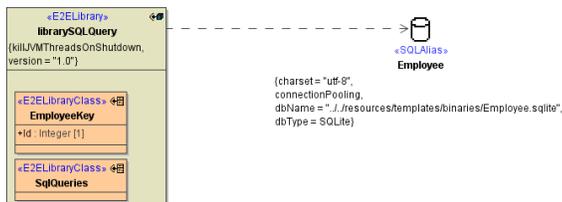
Furthermore, interfaces, signals and enumerations can be specified as external usable classifiers as well by giving them the stereotypes `<<E2ELibraryInterface>>`, `<<E2ELibrarySignal>>` respectively `<<E2ELibraryEnumeration>>`.

## Library Components

The next step is to put the external usable classifiers into an `<<E2ELibrary>>` component, as shown in the next figure. This approach is very similar to an `<<E2ESOAPService>>` component holding an `<<E2ESOAPPortType>>` class. Of course, one difference is that SOAP services must be placed in an `<<E2EComposite>>` component, which is not necessary for libraries because libraries do not run on their own.

The following example shows a library that abstracts a SQL access in a simple class called **SqlQueries**:

Figure: E2ELibrary Component Diagram



The above diagram is an example of a **Library Component Diagram**. Its structure is very similar component diagrams used to define E2E xUML services. The only significant difference comes from the following new stereotypes:

- `<<E2ELibrary>>` components represent the implementation of the library.
- `<<E2ELibraryClass>>`, `<<E2ELibraryInterface>>`, and `<<E2ELibrarySignal>>` classifiers represent the external usable interfaces and data structures nested in E2E libraries.

This component diagram can be generated by using the **xUML Service Library Wizard**. To start this wizard click the icon.

If you create a new xUML service library using the wizard, you will see that the name field when creating a new library is disabled. This is due to the fact, that the library name is equal to the libraries file name, just without the extension. So editing the library name would only be possible by changing the actual model file name.

This library uses a database backend and therefore also needs, as in a normal Builder Project with a backend, an alias.

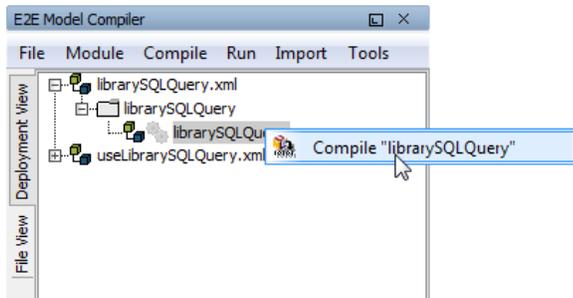
## Library SOAP Service

Apart from the library components, a library may also contain a service (SOAP or REST) and a corresponding component diagram. This can be used for testing purposes, to test the library functionality independently from a library usage service.

A SOAP service definition within a library needs to have set **WSDL per service** to **true**. Otherwise the library will not be imported correctly to the usage service and you cannot deploy the service.

## Compiling the Library

After defining the library you can compile it in the deployment view:



The result of the compile is the library repository **SQLQueryLibrary.Irep**. This file resides in the **../repository/** folder of the current project, because it is a repository that can now be distributed.