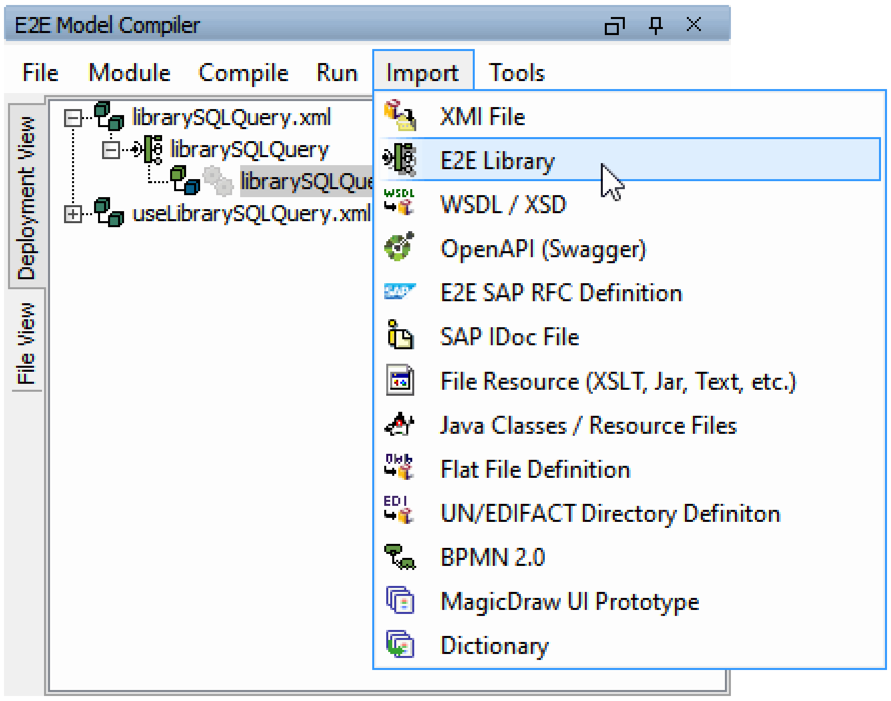
# Using xUML Service Libraries

**Example File (Builder project Advanced Modeling/E2ELibrary):**

| | |
|---|---|
| | <your example path>\Advanced Modeling\E2ELibrary\uml\libraryUseLibrarySQLQuery.xml |

To use the **SQLQueryLibrary.lrep** file in another UML model, it can be imported similar to all other external interfaces using the import dialog:

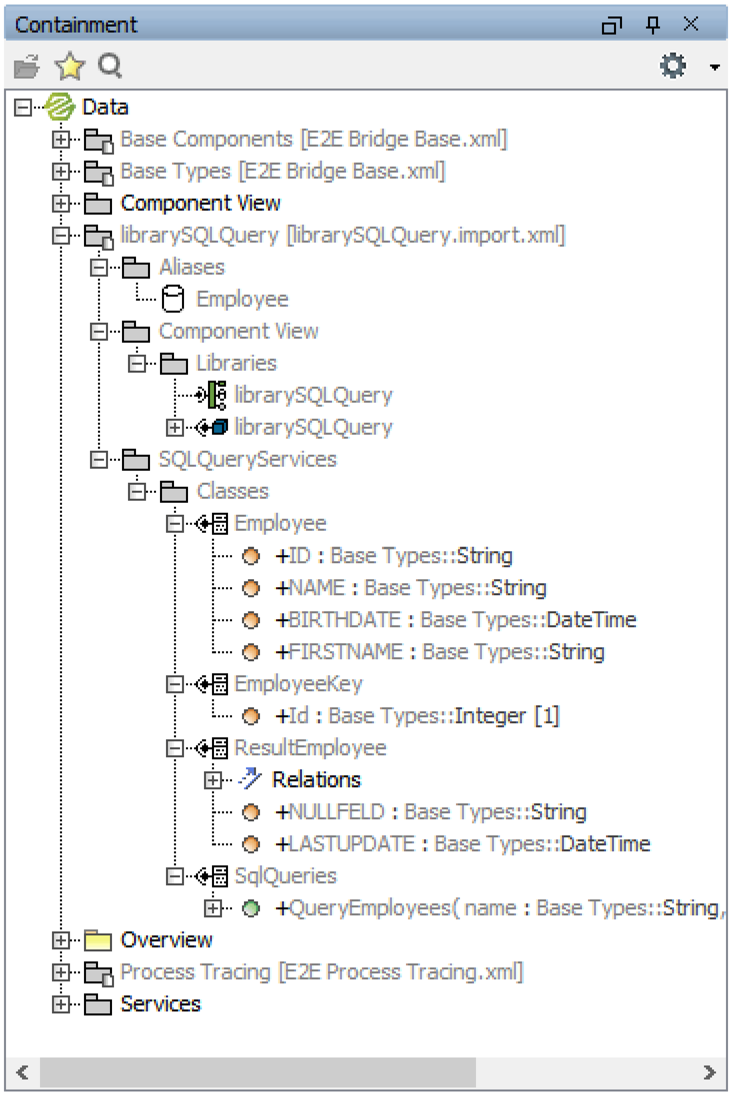


The xUML Service Library Importer guides you through all needed steps. For more details, refer to Importing xUML Service Libraries.

The importer creates a new model file in your current project in the folder **../uml/import**' and links that file as a module into the target model. This module contains proxy elements for all classes, interfaces, signals and enumerations that are available in the repository. Additionally, it holds proxy components for the xUML service library itself.

For example, importing the **librarySQLQuery.lrep** file leads to the following new objects in the containment tree of the target model, which are linked from the new file **librarySQLQuery.import.xml**:

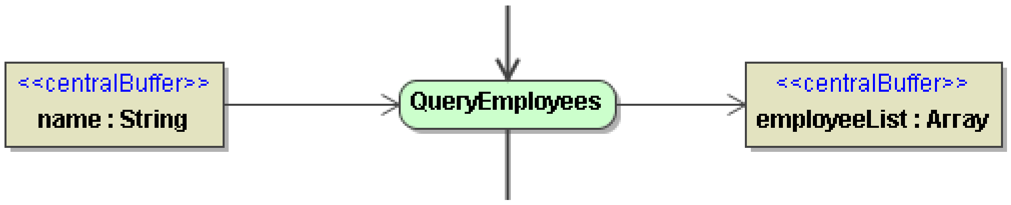
*Figure: Containment Tree Showing Imported Library*


Containment
Data
Base Components [E2E Bridge Base.xml]
Base Types [E2E Bridge Base.xml]
Component View
librarySQLQuery [librarySQLQuery.import.xml]
Aliases
Employee
Component View
Libraries
librarySQLQuery
librarySQLQuery
SQLQueryServices
Classes
Employee
+ID : Base Types::String
+NAME : Base Types::String
+BIRTHDATE : Base Types::DateTime
+FIRSTNAME : Base Types::String
EmployeeKey
+Id : Base Types::Integer [1]
ResultEmployee
Relations
+NULLFELD : Base Types::String
+LASTUPDATE : Base Types::DateTime
SqlQueries
+QueryEmployees( name : Base Types::String,
Overview
Process Tracing [E2E Process Tracing.xml]
Services

You get not only proxies for all <<E2ELibrary>> classes but for all dependent classes as well! For example, the operation QueryEmployees of class SqlQueries uses the class ResultEmployee. Thus, the importer must generate for this class an <<E2EClassProxy>> class too.
Furthermore, the xUML Service Library Importer creates a copy of the imported LREP-File locally in your current project in the **libs** folder. This copy is later used in the component diagram to link the xUML service library to your current model.

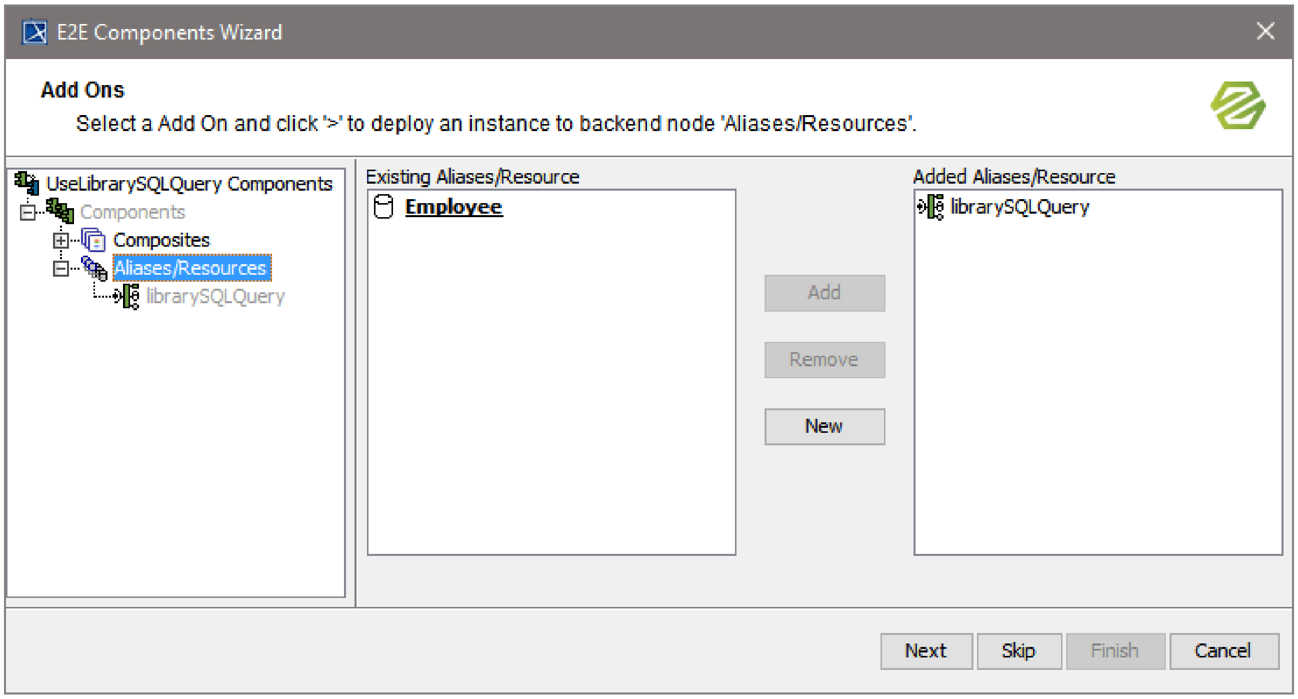
Since the model that is using an xUML service library just links the local files created by the xUML Service Library Importer, the library must be imported again into the Builder project whenever the library has changed and you want to use these changes.
After you imported these classes, you can use them as you would use local defined UML classes.

*Figure: Usage of Imported Library*



However, before you deploy your main model the first time you must tell the system to which composite the imported library shall be linked. This is done with the **Component Wizard** in the component diagram of your main model. Go to the backend services and select one of the imported library artifacts. Then, the screen may look like this:

*Figure: Linkage of Library to xUML Service*


E2E Components Wizard

**Add Ons**

Select a Add On and click '>' to deploy an instance to backend node 'Aliases/Resources'.

- UseLibrarySQLQuery Components
  - Components
    - Composites
    - Aliases/Resources
      - librarySQLQuery

Existing Aliases/Resource

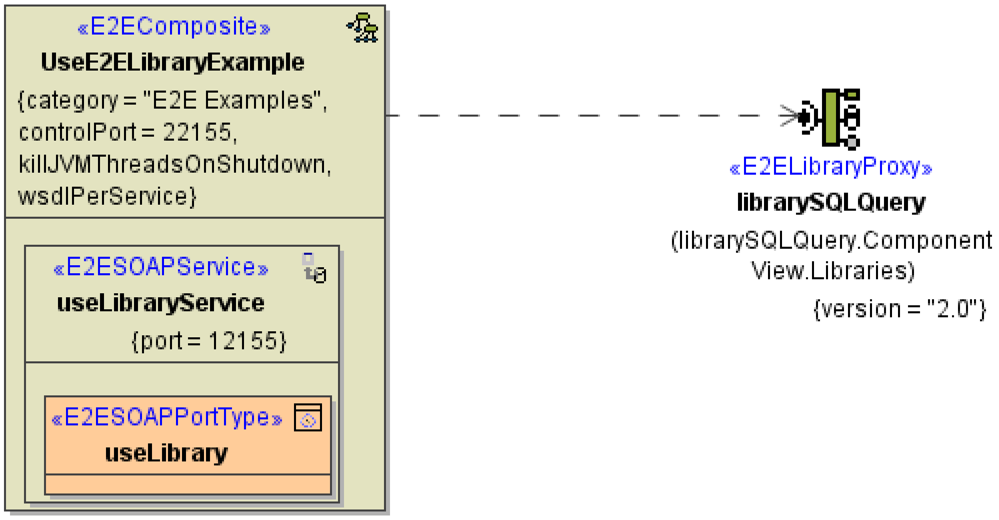
- Employee

Added Aliases/Resource

- librarySQLQuery

In the next step, you must define a dependency that links this artifact with the composite it is going to be linked to. After finishing this wizard, your new component diagram now looks like:

*Figure: Component Diagram when using a E2ELibrary*



After using the wizard you get a component representing the external service similar to the situation you get after importing a SOAP service.

Now you are ready to test your model.