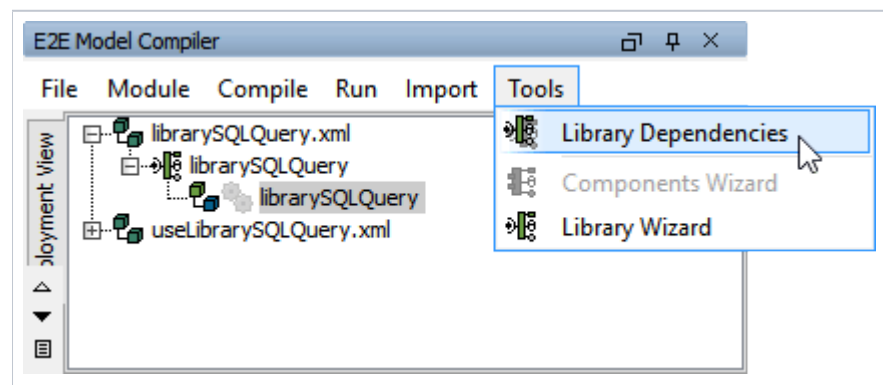
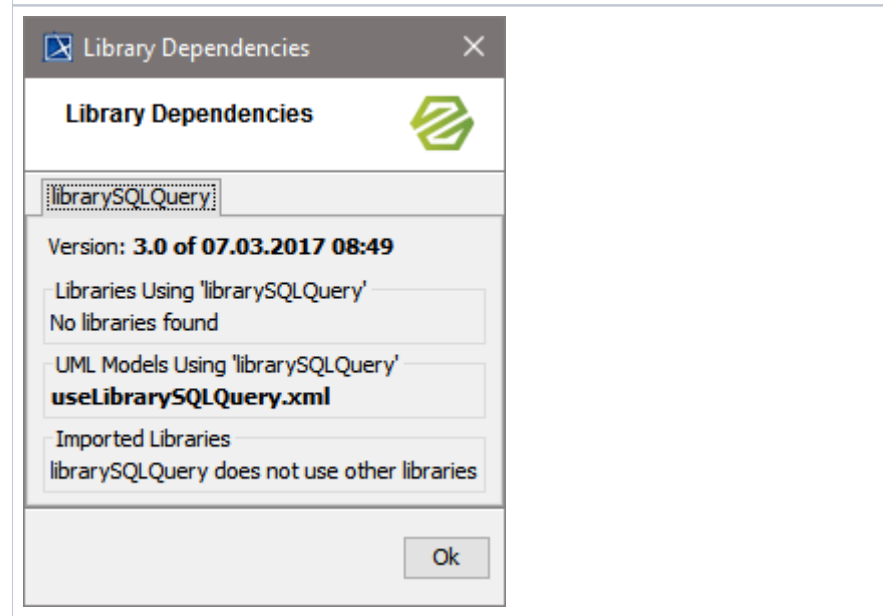


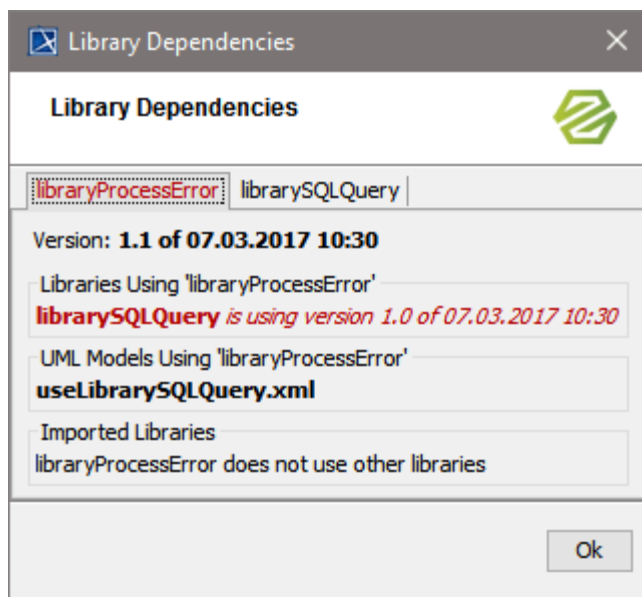
xUML Service Library Dependencies

Sometimes it is necessary to know which xUML service libraries are used in which models in your current Builder project.

	<p>To get this information you can use the Library Dependencies Tool in the xUML Model Compiler.</p>
	<p>The GUI shows the dependencies for each imported xUML service library in your project.</p>

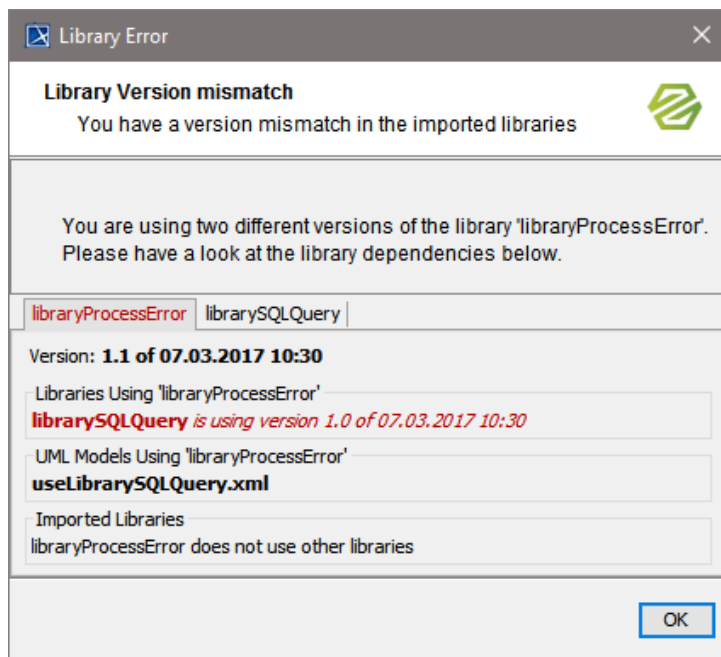
When library uses a library, and both of these libraries are used in a model, both library usages must use the same library version - otherwise, there is a version clash. The **Library Dependencies Tool** indicates such clashes highlighted in red, if there are any.

Let's say, we have build an additional library called **libraryProcessError** which implements a generic error handling. Both, **useLibrarySQLQuery** and **librarySQLQuery** are using **libraryProcessError**, but not in the same version; the library **librarySQLQuery** is using an older version of **libraryProcessError** than **useLibrarySQLQuery**. The GUI shows us the version clash as follows:



Let's say, you have build an additional library **libraryProcessError** which implements a generic error handling. Both, **useLibrarySQLQuery** and **librarySQLQuery** are using **libraryProcessError**, but not in the same version: library **librarySQLQuery** is using an older version of **libraryProcessError** than **useLibrarySQLQuery**.

The **Library Dependencies Tool** displays a version clash.



Such a situation will lead to a compiler error for **useLibrarySQLQuery** later on. In the details view of the compiler error you can see the version mismatch.

To solve this version conflict we have to compile **librarySQLQuery** first, and then re-import it into **useLibrarySQLQuery** again.