# Importing Java Class Files and Resources

In order to use the Java classes and properties resources in your UML model you need to import them first with the Builder. How to do this is described in detail on Importing Java™ Classes and Properties Resource Files.

Do not rename or move the imported Java class structure. The service may not compile or run correctly.

Since the imported classes base on compiled Java classes, the signature of the interface methods is not the same as the one defined in the original Java source code. The initial parameter names are derived from the JavaDoc, or – if not present – the native Java data type:
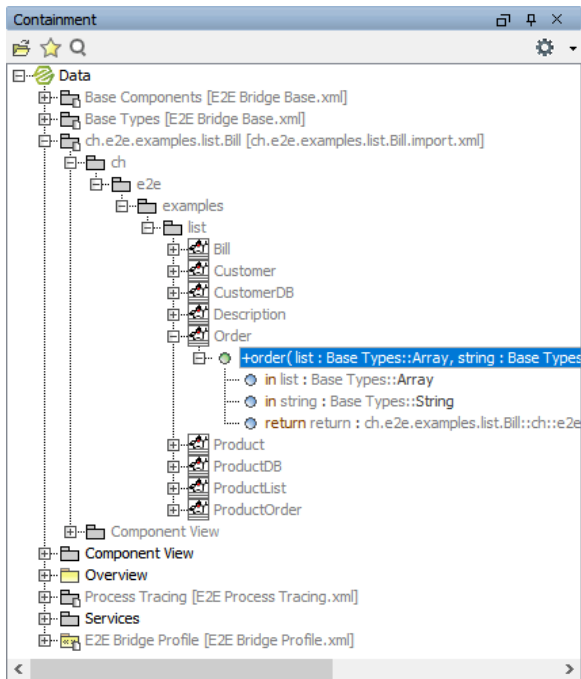
In the specification dialog of a parameter, you can see the native Java type of it. Refer to tagged value **nativeType** .
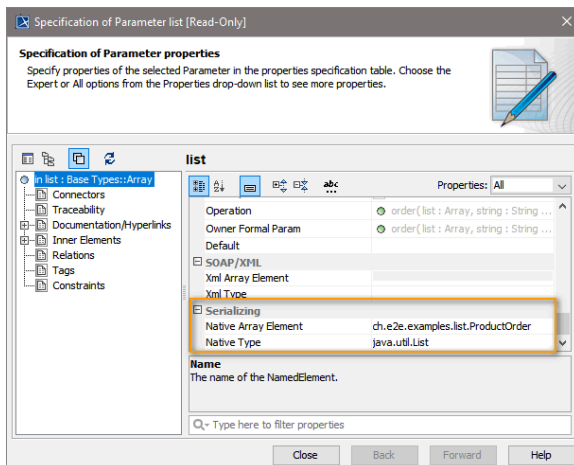
If you have imported a Java class that, after the import, has a method parameter of Base Type **Array**, you can find in the specification dialog the native type as well as the types of its array elements (refer to tagged values **nativeType** and **nativeArrayElement**).

If you have imported a Java class that after the import has a method parameter of type **Map** and if the definition of this Map is not using generics, you have been prompted to define the types of the key-value pair during import. The Importer will create a new class with the stereotype <<MapEntry>>, that will store these mappings.

All mappings will be used internally only when running the xUML service.

Often, the Java application requires properties resource files to run. You can import the resource files separately or together with the Java classes as described on Importing Java™ Classes and Properties Resource Files.

xUML Objects and Java Objects

Please consider the following facts on xUML and Java classes and their mapping when modelling xUML services:

- Objects passed between Java and xUML worlds are mapped. There are no references to the original instances retained.
- You cannot create a Java object in an xUML model. If you create an instance of a class having stereotype <<JavaClass>>, the object is still only an xUML object.