


Java Parameterized Type

Parameterized types are mapped to the general UML Classifier connected with a UML Template Binding to the UML Classifier. The supplier of this binding link is the UML Classifier with UML Template Parameters and represents the Java generic type with type parameters. The client of the UML Template Binding is the UML Classifier of the same UML type as the supplier. Java type parameters are mapped directly to the UML Template Parameter Substitution of the UML Template Binding.

Parameterized type mapping table


Java Element	MagicDraw-UML element
Parameterized type	This is called bounded elements in UML, and is connected to the UML Classifier by the UML Template Binding as a client. The client must be of the same type as the supplier. The supplier must have at least one UML Template Parameter.
Parameter for type	Mapped to the UML Template Parameter Substitution of the UML Template Binding. Each UML Template Parameter from the supplier must be substituted by the UML Template Parameter Substitution. The type of the type parameter is any reference to the UML Classifier from the model set as "Actual" value of the UML Template Parameter Substitution.
Type modifiers of the parameter	Mapped to the MagicDraw property "Type modifiers" of the UML Template Parameter Substitution
Wildcard	A UML Class with the name "?" from the Java Profile is used as an "Actual" value in the UML Template Parameter Substitution.
Wildcard with bounds	Java bounding type is mapped to the UML Classifier and used as the "Actual" value in UML Template Parameter Substitution. "? extends" or "? super" bounding is mapped to the appropriate tag (JavaArgumentBound) value "extends" or "super" of the «JavaTypeArgument» stereotype applied to the UML Template Parameter Substitution.

Example

When you have a type with type variables represented in the model, you can create a parameterized type for Java. For this you must create an empty UML Classifier of the same type as the template classifier (type with type variables). Then create a UML Template Binding and create a UML Template Parameter Substitution for the UML Template Parameters.

Figure 39 -- Created parameterized type for "java.util.List<String>"

Related Pages: To create a "java.util.List<String>" type, you must create a UML Interface first, with a UML Template Parameter representing "java.util.List".

Unknown macro: 'list-children' Figure 36 -- UML Interface representing "java.util.List" with template parameter

Then create another UML Interface and connect it with the List interface with UML Template Binding.

Note that to draw a UML Template Binding, the client element must have at least one UML Template Parameter created

Figure 37 -- UML Template Binding between template class and bounded element

You must open the UML Template Binding specification dialog and create a UML Template Parameter Substitution for the appropriate UML Template Parameter.