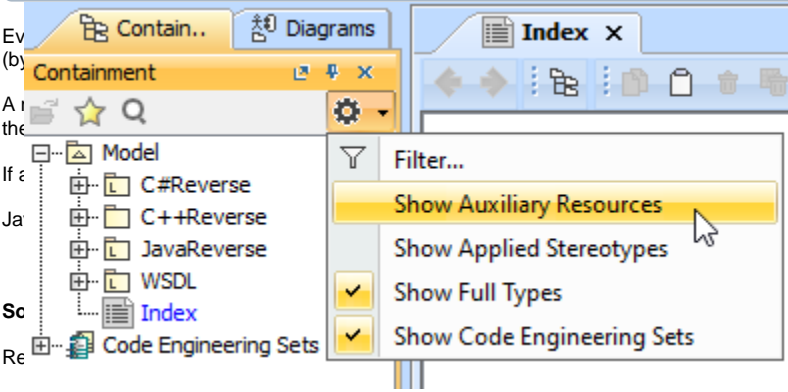


Java Referenced Types

Java built-in types are used from the UML Standard Profile, which is automatically loaded with every new project.

**Show Auxiliary Resources**
The UML Standard Profile by default is hidden. If you want to see it, select **Show Auxiliary Resources** in the [Model Browser](#).



The screenshot shows the 'Model Browser' window with a tree view on the left containing 'Model', 'C#Reverse', 'C++Reverse', 'JavaReverse', 'WSDL', 'Index', and 'Code Engineering Sets'. A context menu is open over the 'Index' node, showing options: 'Filter...', 'Show Auxiliary Resources' (highlighted), 'Show Applied Stereotypes', 'Show Full Types', and 'Show Code Engineering Sets'.


...created into a project and referenced in the CES reference path

...class path is the boot class path taken from the JVM on which ... in **Sorting reversed classes according to the classpath**.

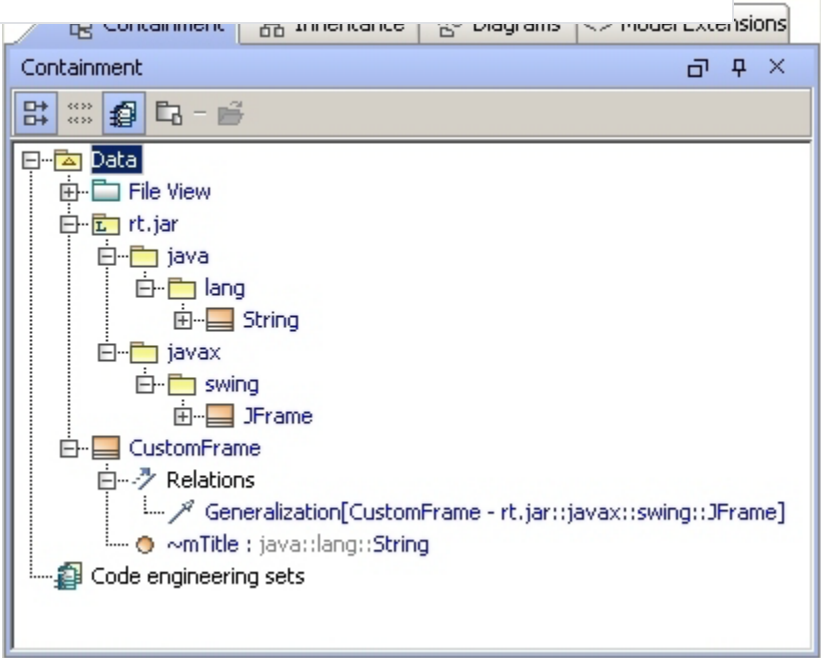
...the *Default* package.

...types.

The figure below shows how the reverse result of the following java code is represented:

**Example**
Java code

```
import javax.swing.*; public class CustomFrame extends JFrame { String mTitle; }
```



The screenshot shows a UML model window titled 'Containment'. It displays a tree structure under a 'Data' package. The tree includes 'File View', 'rt.jar', 'java' (containing 'lang' with 'String'), 'javax' (containing 'swing' with 'JFrame'), and 'CustomFrame'. A 'Relations' package is also shown, containing a 'Generalization' relationship between 'CustomFrame' and 'JFrame'. A note indicates the attribute 'mTitle' is of type 'java::lang::String'. At the bottom, there is a 'Code engineering sets' package.

Sample: The String, JFrame and CustomFrame classes location after reverse

- Related Pages:**
- [Java Referenced Types](#)
 - [Mapping to UML Rules](#)