Stereotypes from Old Project Version

«C++EnumerationLiteral»

This example shows an EnumClass class with a literal value named literal. This literal value applies the «C++EnumerationLiteral» stereotype and sets C++Initializer tag value to 0.

Figure 86 -- «C++EnumerationLiteral» stereotype Example in Class Diagram

The «C++EnumerationLiteral» stereotype is shown in the figure below.

Figure 87 -- «C++EnumerationLiteral» stereotype

The C++Initializer tag value is shown in the figure below.

Figure 88 -- Enumeration Literal Tag Value

Old value	Translation
Enumeration Literal applies the «C++EnumerationLiteral» stereotype and sets C++Initializer tag value to 0.	Apply the «C++LiteralValue» and set the value tag value to 0. Remove the «C++EnumerationLiteral» stereotype and C++Initializer tag value from Enumeration Literal.

The Model shown in the figure below is a translation.

Figure 89 -- Translated Enumeration Literal in Class Diagram

«C++Namespace»

This example is the MyPackage package. It applies the «C++Namespace» stereotype in the old profile and sets the unique namespace name tag value to myNamespace.

Figure 90 -- «C++Namespace» stereotype Example in Class Diagram

The «C++Namespace» stereotype is shown in the figure below.

Figure 91 -- «C++Namespace» stereotype

The unique namespace name tag value is shown in the figure below.

Figure 92 -- unique namespace name tag value

Old value	Translation
Package applies the «C++Namespace» ste- reotype in the old profile (C++ Profile) and sets the unique namespace name tag value to myNamespace.	Apply the «C++Namespace» stereotype in a new profile (c++ ANSI profile) and set the unique namespace name tag value to myNamespace. Remove the «C++Namespace» stereotype (C++ Profile) and unique namespace name tag value.

«C++Typename»

The «C++Typename» stereotype can apply to Template Parameters. Therefore, Elements with template parameters could apply this stereotype.

Figure 93 -- Elements that can have template parameters

In version 12.0, all elements that have template parameters must apply the «C++TemplateParameter» stereotype.

If elements from old version apply the «C++Typename» stereotype, the translation will apply the «C++TemplateParameter» stereotype and set the type keyword tag value to typename. If not, the translation will apply the «C++TemplateParameter» stereotype and set type keyword tag value to class.

Figure 94 -- «C++Typename» stereotype

The «C++TemplateParameter» stereotype is shown in the figure below.

Figure 95 -- «C++TemplateParamater» stereotype

Old value	Translation
Template Parameter does not apply the «C++Typ ename» stereotype.	Apply the «C++TemplateParameter» stereotype and set type keyword tag value to class.
Template Parameter applies the «C++Typename» stereotype.	Apply the «C++TemplateParameter» stereotype and set type keyword tag value to typename. Remove the «C++Typename» stereotype.

«constructor» and «destructor» in UML Standard Profile

These examples are UMLStandardConstructorClass classes that have an operation named myOperation(), which applies the «constructor» stereotype in a UML Standard Profile and UMLStandardDestructorClass class with an operation named myOperation() which applies the «destructor» stereotype in UML Standard Profile.

Figure 96 -- «constructor» and «destructor» stereotype Example in Class Diagram

«constructor»

The «constructor» stereotype is shown in the figure below.

Figure 97 -- «constructor» stereotype

Old value	Translation
Operation applies the «constructor» stereotype in UML Standard Profile.	Apply the «C++Constructor» stereotype (c++ ANSI profile). Remove the «constructor» stereotype (UML Standard Profile).

«destructor»

The «destructor» stereotype is shown in the figure below.

Figure 98 -- «destructor» stereotype

Old value	Translation
Operation applies the «destructor» stereotype in UML Standard Profile.	Apply the «C++Destructor» stereotype (c++ ANSI profile). Remove the «destructor» stereotype (UML Standard Profile).

Related Pages:

- Translation Activity Diagram
- Language Properties
- Conversion with Array Type Modifiers
- Stereotypes from Old Project Version
- Thrown exception tag value translation
- Constructor and Destructor Name From Old Project Versions
- Data Type From Old Project Versions

- Code Engineering Sets
- Generating Code
- Reverse Options
- Global options for Code Engineering
- Files of Properties
- Java Code Engineering
- C++ Code Engineering
- CORBA IDL Mapping To UML