

POSEIDON C# CODE GENERATION PLUGIN GUIDE

General Rules

Tagged Values

These tagged value keys are supported when the value is set to “*true*” within appropriate context:

- internal
- protected internal
- volatile
- override
- sealed
- extern
- internal
- virtual

Additional Stereotypes

- event
- readonly
- delegate

Modelling Element Rules

Class

- Uses the standard UML ‘Class’
- Supports single inheritance only

Class Signature

- Additional visibilities for class signatures are set when the tagged values below are ‘*true*’:
 1. internal
 2. sealed

Class Attributes

- Additional visibilities for class attributes are set when the tagged values below are ‘*true*’:
 1. internal
 2. protected internal
 3. volatile

Class Operations

- Additional visibilities for class operations are set when the tagged values below are ‘*true*’:
 1. internal
 2. protected internal
 3. override
 4. sealed
 5. extern
 6. virtual

Everything else will use the checked visibility radio buttons.

Interface

- Uses the standard UML 'Interface'
- Supports single inheritance only

Interface Signature

- Additional visibilities for interface signatures are set when the tagged value below is 'true':
 1. i n t e r n a l

Interface Members

All interface members implicitly have publ i c access. It is a compile-time error for interface member declarations to include any modifiers. In particular, interface members cannot be declared with the modifiers abstract, publ i c, protected, i n t e r n a l , pri v a t e, vi r t u a l , overri d e, or stati c.

Everything else will use the checked visibility radio buttons.

Structure

- Uses the standard UML 'Class' with the <<Struct>> stereotype
- Supports single inheritance only

Structure Signature

- Additional visibilities for structure signatures are set when the tagged value below is 'true':
 1. i n t e r n a l

Struct types are never abstract and are always implicitly sealed. The abstract and seal ed modifiers are therefore not permitted in a struct declaration. Since inheritance isn't supported for structs, the declared accessibility of a struct member cannot be protected or protected i n t e r n a l .

Structure Members

Function members in a struct cannot be abstract or vi r t u a l , and the overri d e modifier is allowed only to override methods inherited from the type System. Val ueType. A struct may be passed by reference to a function member using a ref or out parameter.

Everything else will use the checked visibility radio buttons.

Enumeration

- Uses the standard UML 'Class' with an <<Enum>> stereotype
- By default, it generates an enum as type i n t .
- Enum does not participate in generalizations or specifications
- Enum cannot have navigable opposite association ends, operations, or inner classifiers
- Anything else will default to i n t .

Enumeration Signature

- Additional visibilities for enumeration signatures are set when the tagged value below is 'true':
 1. i n t e r n a l

Everything else will use the checked visibility radio buttons.

Delegate

- Use standard UML Class with `<<delegate>>` stereotype
- Delegate does not participate in generalization and specification

Delegate Signature

- Additional visibilities for delegate signatures are set when the tagged value below is 'true':
 1. internal

Everything else will use the checked visibility radio buttons.

C# Event

C# events are supported with an operation that has the stereotype `<<event>>`.

Operations

There are some translations on the return type of C# operations:

- in/out will be translated to ref
- in will be translated to blank (" ")
- out will be translated to out
- root will be translated to new