

POSEIDON DELPHI CODE GENERATION PLUGIN GUIDE

General Rules

Classifiers

- Class
- Interface
- Enumeration
- Record
- Set
- Sub Range
- Array
- Exception

Tagged Values

- In classifier tag = 'uses' with value = string that represents unit(s) name to be included in specified unit declaration, separated by comma.

Description:

This will handle string that represents unit(s) name to be included in specified unit declaration. Blank uses tagged value will be defaulted to 'SysUnits'.

Example:

UnitA, UnitB, UnitC

- In attribute tag = 'published' with value = 'true'.

Description:

This will handle published visibility of classifier attribute.

- In operation tag = 'published' with Value = 'true'.

Description:

This will handle published visibility of classifier operation.

- In exception tag = 'published' with Value = 'true'.

Description:

This will handle published visibility of Exception.

- In classifier tag = 'setval ue' with Value = string that represents the value of Set separated by comma.

Description:

This will handle the way to input the value of Set type.

Example:

1,9

- In Classifier Tag = 'subrangeval ue' with Value = string that represents the value of Sub Range separated by comma.

Description:

This will handle the way to input the value of Sub Range type.

Example:

1,9

- In Classifier Tag = 'arrayval ue' with Value = string that represents the value of Array separated by comma.

Description:

This will handle the way to input the value of Array type.

Example:

1,9

- In Classifier Tag = 'arraytype' with value = string that represents the type of Array.

Description:

This will handle the way to input the type of Array type.

- In Operation Tag = 'virtual' with Value = 'true'

Description:

This will handle the way to set the specified operation into a 'virtual' type operation.

- In Operation Tag = 'dynamic' with Value = 'true'

Description:

This will handle the way to set the specified operation into a 'dynamic' type operation.

- In Operation Tag = 'override' with Value = 'true'

Description:

This will handle the way to set the specified operation into a 'override' type operation.

- In Operation Tag = 'overload' with Value = 'true'

Description:

This will handle the way to set the specified operation into a 'overload' type operation.

Note: all string input in Tag column is restricted to be case sensitive.

Stereotypes

Additional stereotypes used in Delphi code generation:

- In attribute stereotype = 'Const'
Description:
This will handle the way to specify a 'const' type attribute.
- In operation stereotype = 'function'
Description:
This will handle the way to specify a 'function' type operation.
- In operation stereotype = 'procedure'
Description:
This will handle the way to specify a 'procedure' type operation.
- In attribute stereotype = 'property'
Description:
This will handle the way to specify a 'property' type attribute.
- In classifier stereotype = 'Enum'
Description:
This will handle the way to specify an 'Enumeration' type classifier.
- In classifier stereotype = 'Record'
Description:
This will handle the way to specify a 'Record' type classifier.
- In classifier stereotype = 'Set'
Description:
This will handle the way to specify a 'Set' type classifier.
- In classifier stereotype = 'SubRange'
Description:
This will handle the way to specify a 'Sub Range' type classifier.
- In Classifier Stereotype = 'Array'
Description:
This will handle the way to specify an 'Array' type classifier.
- In Classifier Stereotype = 'Exception'
Description:
This will handle the way to specify an 'Exception' type classifier.

Modelling Element Rules

Class

- Use standard UML 'Class'
- Participates in generalization, association and specification
- Only support single inheritance

Interface

- Use standard UML 'Interface'
- Participates in generalization
- Does not participate in association and specification
- Only support single inheritance

Enumeration

- Use standard UML 'Class' with <<Enum>> stereotype
- Does not participate in generalization and specification
- Can not have navigable opposite association end and operation

Record

- Use standard UML 'CI ass' with <<Record>> stereotype
- Does not participate in generalization and specification
- Can have navigable opposite association end,
- Can not have any operations

Set

- Use standard UML 'CI ass' with <<Set>> stereotype
- Does not participate in generalization and specification
- Can not have navigable opposite association end and operation

Sub Range

- Use standard UML 'CI ass' with <<SubRange>> stereotype
- Does not participate in generalization and specification
- Can not have navigable opposite association end and operation

Array

- Use standard UML 'CI ass' with <<Array>> stereotype
- Does not participate in generalization and specification
- Can not have navigable opposite association end and operation

Exception

- Use standard UML 'CI ass' with <<Exception>> stereotype
- The same typical with Class.

Specific Rules

Attribute with '*non-1*' multiplicity will generate an Array type that is defaulted to 'int' type with Lower Bound and Upper Bound value based on Lower Bound and Upper Bound value of specified multiplicity.

Example:

Attribute with multiplicity: 1..2 will generate : Array[1..2] of int;

- Blank input of 'uses' Tag, will be defaulted to 'SysUti I s'.
- Blank input of 'setval ue' Tag will be defaulted to 'a'..'z'
- Blank input of 'subrangeval ue' Tag will be defaulted to 'a'..'z'
- Blank input of 'arrayval ue' Tag will be defaulted to '1'..'10'
- Blank input of 'arraytype' Tag will be defaulted to int
- Blank input of 'procedure' and 'functi on' Tag will be defaulted to procedure