

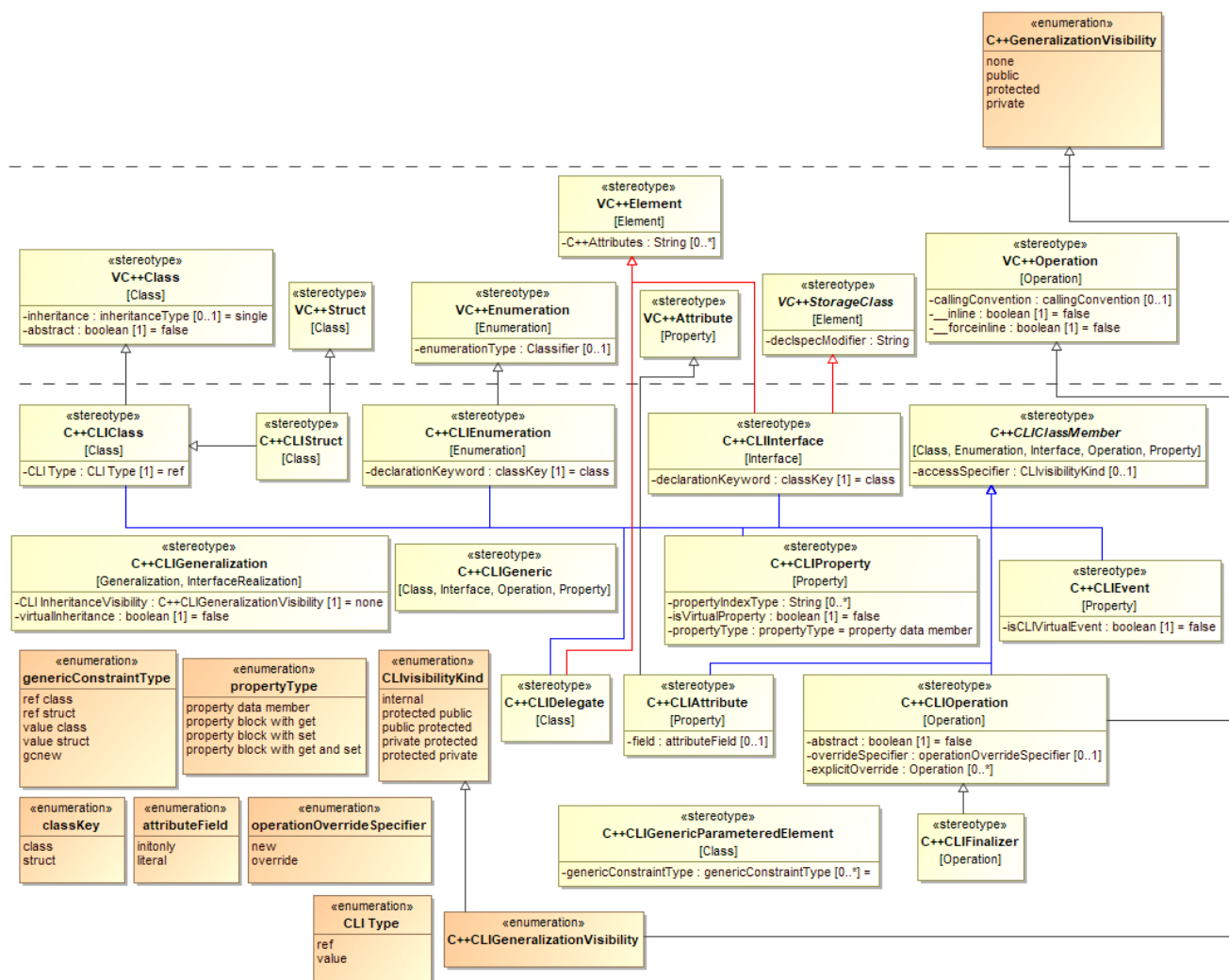
C++/CLI Profile



C++/CLI Profile

The profile table and description in this section do not include the tagged value inherited from C++ ANSI profile.

Stereotype



C++/CLI Stereotypes

C++CLI Class

Name	Meta Class	Constraints
C++CLIClass	Class	
Tag	Type	Description
CLI Type	CLI Type[1] = ref (Enumeration) See CLI Type	Represents the utilization of ref class or value class keywords for defining a CLR class.
Inherited Tag	Type	Description

inheritance «VC++Class»	inheritanceType[0..1] = single (Enumeration) See inheritanceType	Represents the utilization of VC++ keywords, __single inheritance , __multiple_inheritance , and __virtual_inheritance .
declspecModifier «VC++StorageClass»	String	For keeping an extended declaration modifier of __declspec See <i>Extended storage-class attributes with __declspec</i>
C++Attributes «VC++Element»	String	For keeping C++ Attributes
abstract «VC++Class»	boolean[1] = false	Represents the utilization of an abstract keyword.
accessSpecifier «C++CLIClassMember»	CLIVisibilityKind[0..1] (Enumeration) See CLIVisibilityKind	Represents the usage of visibility kind introduced in C++/CLI, including: • internal • protected public • public protected • private protected • protected private

C++CLIStruct

Name	Meta Class	Constraints
C++CLIStruct	Class	
Inherited Tag	Type	Description
managedType «C++CLIClass»	CLI Type[1] = ref (Enumeration) See CLI Type	Represents the utilization of a ref class or value class keywords for defining a CLR class.
abstract «C++CLIClass»	boolean[1] = false	Represents the utilization of an abstract keyword.
inheritance «VC++Class»	inheritanceType[0..1] = single (Enumeration) See inheritanceType	Represent the utilization of VC++ keywords, __single inheritance , __multiple_inheritance , and __virtual_inheritance .
declspecModifier «VC++StorageClass»	String	For keeping an extended declaration modifier of __declspec See <i>Extended storage-class attributes with __declspec</i>
C++Attributes «VC++Element»	String	For keeping C++ Attributes

C++CLIEnumeration

Name	Meta Class	Constraints
C++CLIEnumeration	Enumeration	
Tag	Type	Descriptions
declarationKeyword	classKey[1] = class (Enumeration) See classKey	Represents the utilization of enum class or enum struct keywords. See enum class, enum struct
Inherited Tag	Type	Description
type «VC++Enumeration»	Classifier[0..1]	Represents the type of enumeration literal

C++CLIInterface

Name	Meta Class	Constraints
C++CLIInterface	Interface	
Tag	Type	Description
declarationKeyword	classKey[1] = class (Enumeration) See classKey	Represents the utilization of interface class or interface struct . keywords. See ref class, ref struct, value class, value struct, interface class, interface struct
Inherited Tag	Type	Description
declspecModifier «VC++StorageClass»	String	For keeping an extended declaration modifier of __declspec See <i>Extended storage-class attributes with __declspec</i>
C++Attributes «VC++Element»	String	For keeping C++ Attributes

C++CLIClassMember

Name	Meta Class	Constraints
C++CLIClassMember	Operation, Property	This stereotype is an abstract stereotype for keeping access - Specifier tag definition.
Tag	Type	Descriptions
accessSpecifier	CLVisibilityKind[0..1] (Enumeration) See CLVisibilityKind	Represents the usage of visibility kind introduced in C++/CLI including: • internal • protected public • public protected • private protected • protected private

C++CLIAttribute

Name	Meta Class	Constraints
C++CLIAttribute	Property	
Tag	Type	Descriptions
field	attributeField[0..1] (Enumeration) See attributeField	Represents the usage of initonly or literal keywords.
Inherited Tag	Type	Description
accessSpecifier «C++CLIClassMember»	CLVisibilityKind[0..1] (Enumeration) See CLVisibilityKind	Represents the usage of visibility kind introduced in C++/CLI including: • internal • protected public • public protected • private protected • protected private

declspecModifier «VC++StorageClass»	String	For keeping extended declaration modifier of __declspec See <i>Extended storage-class attributes with __declspec</i>
C++Attributes «VC++Element»	String	For keeping C++ Attributes

C++CLIOperation

Name	Meta Class	Constraints
C++CLIOperation	Operation	
Tag	Type	Descriptions
abstract	boolean[1] = false	Represents the utilization of abstract keyword.
overrideSpecifier	operationOverrideSpecification[0..1] (Enumeration) See Override Specifiers	Represents the utilization of C++/CLI operation override specification, new and override .
explicitOverride	Operation[0..*]	Represents the usage of explicit override feature.
Inherited Tag	Type	Description
accessSpecifier «C++CLIClassMember»	CLIVisibilityKind[0..1] (Enumeration) See CLIVisibilityKind	Represents the usage of visibility kind introduced in C++/CLI including: • internal • protected public • public protected • private protected • protected private
callingConvention «VC++Operation»	callingConvention[0..1] See callingConvention	Represents the utilization of keywords. • __cdecl • __clrcall • __stdcall • __fastcall • __thiscall
C++Attributes «VC++Element»	String	For keeping C++ Attributes

C++CLIProperty

«C++CLIProperty» is used to define a CLR property, which has the appearance of an ordinary data member, and can be written to or read from using the same syntax as a data member.

Name	Meta Class	Constraints
C++CLIProperty	Property	The stereotype must not be applied to the same attribute as «C++Attribute» or other stereotypes derived from «C++Attribute»
Tag	Type	Descriptions
propertyIndexType	String[0..*]	index type keeps the list of property index types.
isVirtualProperty	boolean[1] = false	Specify whether the property. is virtual or not.

propertyType	PropertyType[1] = property data member	Specify property type between: - property data member - property block with get - property block with set - property block with get and set
Inherited Tag	Type	Description
accessSpecifier «C++CLIClassMember»	CLIVisibilityKind[0..1] (Enumeration) See CLIVisibilityKind	Represents the usage of visibility kind introduced in C++/CLI including: - internal - protected public - public protected - private protected - protected private

C++CLIDelegate

Name	Meta Class	Constraints
C++CLIDelegate	Class	Declaration of delegate can be only in a managed type. The declaration of delegate cannot have attribute and operation. Represents the utilization of delegate keyword.
Tag	Type	Descriptions
C++CLIEvent	boolean[1] = false	Specify whether the event is virtual or not.
Inherited Tag	Type	Description
accessSpecifier «C++CLIClassMember»	CLIVisibilityKind[0..1] (Enumeration) See CLIVisibilityKind	Represents the usage of visibility kind introduced in C++/CLI including: - internal - protected public - public protected - private protected - protected private

C++CLIGeneric

Name	Meta Class	Constraints
C++CLIGeneric	Class, Interface, Operation, Property	Represents the usage of a generic keyword.
C++CLIGenericParameterizedElement	Class	
Tag	Type	Descriptions
genericConstraintType	genericConstraintType[0..*] (Enumeration) See genericConstraintType	Specifies generic constraint type, including - ref class - ref struct - value class - value struct - gnew

C++CLIGeneralization

Name	Meta Class	Constraints
C++CLIGeneralizationParameteredElement	Class	
Tag	Type	Descriptions
genericConstraintType	genericConstraintType[0..*] (Enumeration) See genericConstraintType	Specifies generic constraint type, including • ref class • ref struct • value class • value struct • gnew

C++CLIGeneralization

Name	Meta Class	Constraints
C++CLIGeneralization	Generalization	
Tag	Type	Descriptions
CLIInheritanceVisibility	C++CLIGeneralizationVisibility[1] = none (Enumeration) See C++CLIGeneralizationVisibility	
virtualInheritance	boolean[1] = false	

Enumeration

CLI Type

CLI Type will be used as value of tag named "**CLI Type**" under «C++CLIClass». The possible values are **ref** and **value**.

Literal	Description
ref	For defining a CLR reference class
value	For defining a CLR value class

operation OverrideSpecifier

operationOverrideSpecifier will be used as value of tag named "**overrideSpecifier**" under «VC++Operation». The possible values are **new** and **override** which are keywords that can be used to qualify override behavior for derivation.

Literal	Description
new	Indicates the use of a new keyword to qualify override behavior for derivation. In VC++, new is a keyword indicating that a virtual member will get a new slot in the table; the function does not override a base class method.
override	Indicates the use of the override key to qualify override behavior for derivation. In VC++, override is a keyword indicating that a member of a managed type must override a base class or a base interface member. If there is no member to override, the compiler will generate an error.

attributeField

attributeField will be used as the value of the tag named "field" under «C++CLIAttribute». The possible values are **initonly** and **literal**, keywords used to qualify the field type of attribute.

Literal	Description
---------	-------------

initonly	Represents the utilization of initonly keyword. initonly indicates that variable assignment can only occur as part of the declaration or in a static constructor in the same class.
literal	Represents the utilization of literal keyword. It is the native equivalent of static const variable. Constraint: Is Static and Is Read Only must be set to true .

classKey

Literal	Description
class	Represents the keyword that has the word, class .
struct	Represents the keyword that has the word, struct .

Remark:

- **enum class** and **enum struct** are equivalent declarations.
- **interface class** and **interface struct** are equivalent declarations.

CLIVisibilityKind

Literal	Description
internal	Represents the internal visibility.
protected public	Represents the protected public visibility.
public protected	Represents the public protected visibility.
private protected	Represents the private protected visibility.
protected private	Represents the protected private visibility.

genericConstraintType

Literal	Description
ref class	Represents the usage of the ref class keyword in a generic constraint clause.
ref struct	Represents the usage of the ref struct keyword in a generic constraint clause.
value class	Represents the usage of the value class keyword in a generic constraint clause.
value struct	Represents the usage of the value strict keyword in a generic constraint clause.
gcnew	Represents the usage of the gcnew keyword in a generic constraint clause.

C++CLIGeneralizationVisibility

C++CLIGeneralizationVisibility is an enumeration inheriting from C++GeneralizationVisibility and CLIVisibility- Kind. Its literals include:

- **none,**
- **public,**
- **protected,**
- **private,**
- **internal,**
- **protected public,**
- **public protected,**
- **private protected,** and
- **protected private.**

