

Modeling with Microsoft Specific Profiles

CLR Data Type Keyword

ref class, ref struct, value class, value struct, interface class, interface struct

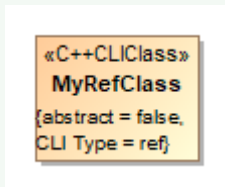
- class_access ref class name modifier : inherit_access base_type {};



Code:

```
ref class MyRefClass
{
};
```

UML model with applied Stereotype «C++CLIClass» and Tag value **CLI Type** = ref



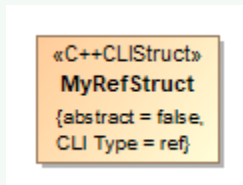
- class_access ref struct name modifier : inherit_access base_type {};



Code:

```
ref struct MyRefStruct
{
};
```

UML model with applied Stereotype «C++CLIStruct» and Tag value **CLI Type** = ref



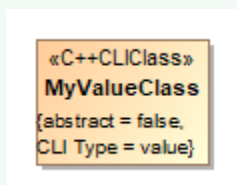
- class_access value class name modifier : inherit_access base_type {};



Code:

```
value class MyValueClass
{
};
```

UML model with applied Stereotype «C++CLIClass» and Tag value **CLI Type** = value



- class_access value struct name modifier : inherit_access base_type {};

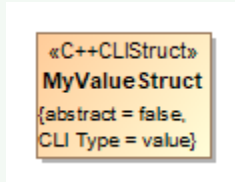




Code:

```
value struct MyValueStruct
{
};
```

UML model with applied Stereotype «C++CLIStruct» and Tag value **CLI Type** = value



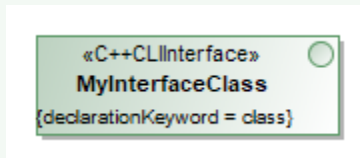
- interface_access interface class name : inherit_access base_interface {};



Code:

```
interface class MyInterfaceClass
{
};
```

UML model with applied Stereotype «C++CLInterface» and Tag value **declarationKeyword** = class



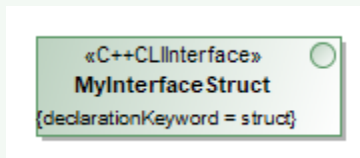
- interface_access interface struct name : inherit_access base_interface {};



Code:

```
interface struct MyInterfaceStruct
{
};
```

UML model with applied Stereotype «C++CLInterface» and Tag value **declarationKeyword** = struct



enum class, enum struct

- access enum class name [: type] { enumerator-list } var;

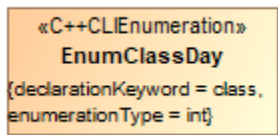




Code:

```
enum class EnumClassDay:int
{
    sun,
    mon
};
```

UML model with applied Stereotype «C++CLIEnumeration» and Tag value **declarationKeyword** = class, **enumerationType** = int



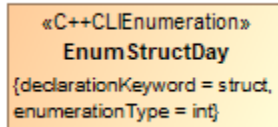
- access enum struct name [:type] { enumerator-list } var;



Code:

```
enum class EnumClassDay:int
{
    sun,
    mon
};
```

UML model with applied Stereotype «C++CLIEnumeration» and Tag value **declarationKeyword** = struct, **enumerationType** = int



property





Code:

```
public ref class MyClass
{
    // property data memberproperty
    property String ^ propertyX;
    // property block
    property int propertyY
    {
        int get();
        void set(int value);
    }
    //property block with index
    property int propertyZ[int, long]
    {
        int get(int index1,long index2);
        void set(int index1,long index2, int value);
    }
};
```

UML model:

MyClass
«C++CLIProperty»-propertyX : String""\$^""{isVirtualProperty = false, propertyType = property data member}
«C++CLIProperty»-propertyY : int{isVirtualProperty = false, propertyType = property block with get and set}
«C++CLIProperty»-propertyZ : int{isVirtualProperty = false, propertyIndexType = "int", "long", propertyType = property block with get and set}

Specification of Property propertyZ

Relations

Connectors

Tags

Constraints

C++ Language Properties

Microsoft Visual C++ Language

C++/CLI Language Properties

Traceability

Language Properties

C++/CLI Language Properties

Access Specifier

Field

Is CLI Virtual Event

Is Virtual Property

Property Index Type

Property Type

<undefined>

false

int

long

property block with get and set

Close

Help

Example of C++/CLI Language Properties for describing Property *propertyZ*

delegate

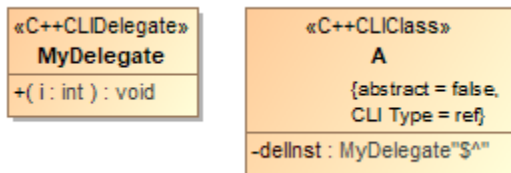




Code:

```
public delegate void MyDelegate(int i);  
ref class A  
{  
    MyDelegate^ delInst;  
};
```

UML model:



Specification of Operation <>

Name	Type	Default Value	Direction
i	int [UML Stand...		inout
	void [UML Stand...		return

event



Code:

```
public delegate void  
MyDelegate(int);  
ref class B {  
    event MyDelegate^ MyEvent;  
};
```

UML model:

