## **Always Visible Properties**

Generally, stereotype properties become visible in an element's Specification window only after the stereotype is applied on the element. However, you may customize stereotype properties to be visible in the element Specification window even if the stereotype is not yet applied to the element.

This feature allows specifying domain-specific properties for standard UML elements. It is also used to make stereotype properties (**Type Modifier**, **Active Hyperlink**, and **To Do**, etc.) visible even if appropriate stereotypes are not yet applied to an element.

Property		
Name	country	
Туре	Character [java::lang]	
Type Modifier	0	
Visibility	private	
Default Value		
Applied Stereotype	<ul> <li>TODO_Owner [Element] [UML Standard Profile::MagicDraw Profile]</li> <li>typeModifier [Element] [UML Standard Profile::MagicDraw Profile]</li> <li>HyperlinkOwner [Element] [UML Standard Profile::MagicDraw Profile]</li> </ul>	
Multiplicity	1	
Is Read Only	false	
Is Static	false	
Aggregation	none	
Is Derived	false	
Active Hyperlink	📷 magiclibrary::domain::User registration::User registration	
To Do	Please review if the multiplicity is correct.	

Always visible properties in UML property Specification Window.

To make stereotype properties visible when the stereotype is not yet applied to an element

- 1. Open the customization element Specification window.
- Click to change the Show Properties When Not Applied property to true. All properties of the customized stereotype will become visible in the DSL element Specification window even if the stereotype is not yet applied to the element.

## **Case Study**

Use the *C# Profile* (*C#\_Profile.mdzip*) in your project and you will see the **C# Language Properties** appear in a class Specification window. C# properties will appear in the Specification window of a class even though no profile-specific stereotype is applied to the class.

C# properties in class Specification window .

Specify a desired C# property value in the class Specification window and you will see the «C#LanguageProperty» stereotype automatically applied to the class.

## **Related Pages**

- Creating Customization
   Data
- Using Customization DataCustomizing Specification
- Window • Advanced cases of
- properties visibility

	Address (Internal) V: Character'II' (1) Class - Address Specification of C# Language I	omatically applied stereotype Properties ains a list of specific C≢ Language Properties
1 prof prile ) edWorkin te	Ports     Constraints     Constraints	C# Language Properties E # Language Properties Extern Alas Internal
	Close	Hep

Class with automatically applied «C#LanguageProperty» stereotype.