

Class Specifiers

Keywords used when declaring UClasses to specify how the class behaves with various aspects of the Engine and Editor.



When declaring classes, **Class Specifiers** can be added to the declaration to control how the class behaves with various aspects of the Engine and Editor.

Class Specifier	Effect
<code>Abstract</code>	The Abstract Specifier declares the class as an "abstract base class", preventing the user from adding Actors of this class to Levels. This is useful for classes which are not meaningful on their own. For example, the <code>ATriggerBase</code> base class is abstract, while the <code>ATriggerBox</code> subclass is not abstract and can be placed in a Level.
<code>AdvancedClassDisplay</code>	The AdvancedClassDisplay Specifier forces all properties of the class to show only in the advanced sections of any details panel where they appear. To override this on an individual property, use the <code>SimpleDisplay</code> specifier on that property.
<code>AutoCollapseCategories=(Category1, Category2, ...)</code>	The <code>AutoCollapseCategories</code> Specifier negates the effects, for the listed categories, of the AutoExpandCategories Specifier on a parent class.
<code>AutoExpandCategories=(Category1, Category2, ...)</code>	Specifies one or more categories that should be automatically expanded in the Unreal Editor Property window for Objects of this class. To auto-expand variables declared with no

Class Specifier

Effect

	category, use the name of the class which declares the variable.
Blueprintable	Exposes this class as an acceptable base class for creating Blueprints. The default is NotBlueprintable, unless inherited otherwise. This Specifier is inherited by subclasses.
BlueprintType	Exposes this class as a type that can be used for variables in Blueprints.
ClassGroup=GroupName	Indicates that Unreal Editor's Actor Browser should include this class and any subclass of this class within the specified GroupName when Group View is enabled in the Actor Browser.
CollapseCategories	Indicates that properties of this class should not be grouped in categories in Unreal Editor Property windows. This Specifier is propagated to child classes, and can be overridden by the DontCollapseCategories Specifier.
Config=ConfigName	Indicates that this class is allowed to store data in a configuration file (.ini). If there are any class properties declared with the config or globalconfig Specifiers, this Specifier causes those properties to be stored in the named configuration file. This Specifier is propagated to all child classes and cannot be negated, but child classes can change the config file by re-declaring the config Specifier and providing a different ConfigName. Common ConfigName values are "Engine", "Editor", "Input", and "Game".
Const	All properties and functions in this class are const and are exported as const. This Specifier is inherited by subclasses.
ConversionRoot	A root convert limits a subclass to only be able to convert to child classes of the first root class going up the hierarchy.
CustomConstructor	Prevents automatic generation of the constructor declaration.

Class Specifier

Effect


<div>DefaultToInstanced</div>	All instances of this class are considered "instanced". Instanced classes (components) are duplicated upon construction. This Specifier is inherited by subclasses.
<div>DependsOn=(ClassName1, ClassName2, ...)</div>	All classes listed will be compiled before this class. The class names provided must indicate classes in the same (or a previous) package. Multiple dependency classes can be identified using a single <div>DependsOn</div> line delimited by commas, or can be specified using a separate <div>DependsOn</div> line for each class. This is important when a class uses a struct or enum declared in another class, as the compiler only knows what is in the classes it has already compiled.
<div>Deprecated</div>	This class is deprecated and Objects of this class will not be saved when serializing. This Specifier is inherited by subclasses.
<div>DontAutoCollapseCategories=(Category, Category, ...)</div>	Negates the <div>AutoCollapseCategories</div> Specifier for the listed categories inherited from a parent class.
<div>DontCollapseCategories</div>	Negates a <div>CollapseCatogories</div> Specifier inherited from a base class.
<div>EditInlineNew</div>	Indicates that Objects of this class can be created from the Unreal Editor Property window, as opposed to being referenced from an existing Asset. The default behavior is that only references to existing Objects may be assigned through the Property window). This Specifier is propagated to all child classes; child classes can override this with the <div>NotEditInlineNew</div> Specifier.
<div>HideCategories=(Category1, Category2, ...)</div>	Lists one or more categories that should be hidden from the user entirely. To hide properties declared with no category, use the name of the class which declares the variable. This Specifier is propagated to child classes.
<div>HideDropdown</div>	Prevents this class from showing up in property window combo boxes.

Class Specifier	Effect
<code>HideFunctions=(Category1, Category2, ...)</code>	Hides all functions in the specified category from the user entirely.
<code>HideFunctions=FunctionName</code>	Hides the named functions from the user entirely.
<code>Intrinsic</code>	This indicates that the class was declared directly in C++, and has no boilerplate generated by Unreal Header Tool . Do not use this Specifier on new classes.
<code>MinimalAPI</code>	Causes only the class's type information to be exported for use by other modules. The class can be cast to, but the functions of the class cannot be called (with the exception of inline methods). This improves compile times by not exporting everything for classes that do not need all of their functions accessible in other modules.
<code>NoExport</code>	Indicates that this class's declaration should not be included in the automatically-generated C++ header file by the header generator. The C++ class declaration must be defined manually in a separate header file. Only valid for native classes. Do not use this for new classes.
<code>NonTransient</code>	Negates a <code>Transient</code> Specifier inherited from a base class.
<code>NotBlueprintable</code>	Specifies that this class is not an acceptable base class for creating Blueprints. This is the default and is inherited by subclasses.
<code>NotPlaceable</code>	Negates a <code>Placeable</code> Specifier inherited from a base class. Indicates that Objects of this class may not be placed into a Level, UI scene, or Blueprint in the Editor.
<code>PerObjectConfig</code>	Configuration information for this class will be stored per Object, where each object has a section in the <code>.ini</code> file named after the Object in the format <code>[ObjectName ClassName]</code> . This Specifier is propagated to child classes.
<code>Placeable</code>	Indicates that this class can be created in the Editor and placed into a level, UI scene, or Blueprint (depending on the class type). This

Class Specifier	Effect
	flag is propagated to all child classes; child classes can override this flag using the <code>NotPlaceable</code> Specifier.
<code>ShowCategories=(Category1, Category2, ...)</code>	Negates a <code>HideCategories</code> Specifier (inherited from a base class) for the listed categories.
<code>ShowFunctions=(Category1, Category2, ...)</code>	Shows all functions within the listed categories in a property viewer.
<code>ShowFunctions=FunctionName</code>	Shows the named function in a property viewer.
<code>Transient</code>	Objects belonging to this class will never be saved to disk. This is useful in conjunction with certain kinds of native classes which are non-persistent by nature, such as players or windows. This Specifier is propagated to child classes, but can be overridden by the <code>NonTransient</code> Specifier.
<code>Within=OuterClassName</code>	Objects of this class cannot exist outside of an instance of an <code>OuterClassName</code> Object. This means that creating an Object of this class requires that an instance of <code>OuterClassName</code> is provided as its <code>Outer</code> Object.

Metadata Specifiers

When declaring classes, interfaces, structs, enums, enum values, functions, or properties, you can add **Metadata Specifiers** to control how they interact with various aspects of the engine and editor. Each type of data structure or member has its own list of Metadata Specifiers.

 Metadata only exists in the editor; do not write game logic that accesses metadata.

Classes can use the following Metatag Specifiers:

Class Meta Tag	Effect
<code>BlueprintSpawnableComponent</code>	If present, the component Class can be spawned by a Blueprint.
<code>BlueprintThreadSafe</code>	Only valid on Blueprint function libraries. This specifier marks the functions in this class as

Class Meta Tag

Effect

	callable on non-game threads in animation Blueprints.
ChildCannotTick	Used for Actor and Component classes. If the native class cannot tick, Blueprint-generated classes based on this Actor or Component can never tick, even if bCanBlueprintsTickByDefault is true.
ChildCanTick	Used for Actor and Component classes. If the native class cannot tick, Blueprint-generated classes based on this Actor or Component can have the bCanEverTick flag overridden, even if bCanBlueprintsTickByDefault is false.
DeprecatedNode	For behavior tree nodes, indicates that the class is deprecated and will display a warning when compiled.
DeprecationMessage="Message Text"	Deprecated classes with this metadata will include this text with the standard deprecation warning that Blueprint Scripts generate during compilation.
DisplayName="Blueprint Node Name"	The name of this node in a Blueprint Script will be replaced with the value provided here, instead of the code-generated name.
DontUseGenericSpawnObject	Do not spawn an Object of the class using Generic Create Object node in Blueprint Scripts; this specifier applies only to Blueprint-type classes that are neither Actors nor Actor Components.
ExposedAsyncProxy	Expose a proxy Object of this class in Async Task nodes.
IgnoreCategoryKeywordsInSubclasses	Used to make the first subclass of a class ignore all inherited ShowCategories and HideCategories Specifiers.
IsBlueprintBase="true/false"	States that this class is (or is not) an acceptable base class for creating Blueprints, similar to the Blueprintable or NotBlueprintable Specifiers.

Class Meta Tag	Effect
<code>KismetHideOverrides="Event1, Event2, .."</code>	List of Blueprint events that are not allowed to be overridden.
<code>ProhibitedInterfaces="Interface1, Interface2, .."</code>	Lists Interfaces that are not compatible with the class.
<code>RestrictedToClasses="Class1, Class2, .."</code>	Blueprint function library classes can use this to restrict usage to the classes named in the list.
<code>ShortToolTip="Short tooltip"</code>	A short tooltip that is used in some contexts where the full tooltip might be overwhelming, such as the Parent Class Picker dialog.
<code>ShowWorldContextPin</code>	Indicates that Blueprint nodes placed in graphs owned by this class must show their World context pins, even if they are normally hidden, because Objects of this class cannot be used as World context.
<code>UsesHierarchy</code>	Indicates the class uses hierarchical data. Used to instantiate hierarchical editing features in Details panels.
<code>ToolTip="Hand-written tooltip"</code>	Overrides the automatically generated tooltip from code comments.
<code>ScriptName="DisplayName"</code>	The name to use for this clas, property, or function when exporting it to a scripting language. You may include deprecated names as additional semi-colon-separated entries.