required modifier (C# Reference)

Article • 01/31/2023

The `required` modifier indicates that the *field* or *property* it's applied to must be initialized by an object initializer. Any expression that initializes a new instance of the type must initialize all *required members*. The `required` modifier is available beginning with C# 11. The `required` modifier enables developers to create types where properties or fields must be properly initialized, yet still allow initialization using object initializers. Several rules ensure this behavior:

- The `required` modifier can be applied to *fields* and *properties* declared in `struct`, and `class` types, including `record` and `record struct` types. The `required` modifier can't be applied to members of an `interface`.
- Explicit interface implementations can't be marked as `required`. They can't be set in object initializers.
- Required members must be initialized, but they may be initialized to `null`. If the type is a non-nullable reference type, the compiler issues a warning if you initialize the member to `null`. The compiler issues an error if the member isn't initialized at all.
- Required members must be at least as visible as their containing type. For example, a `public` class can't contain a `required` field that's `protected`. Furthermore, required properties must have setters (`set` or `init` accessors) that are at least as visible as their containing types. Members that aren't accessible can't be set by code that creates an instance.
- Derived classes can't hide a `required` member declared in the base class. Hiding a required member prevents callers from using object initializers for it. Furthermore, derived types that override a required property must include the `required` modifier. The derived type can't remove the `required` state. Derived types can add the `required` modifier when overriding a property.
- A type with any `required` members may not be used as a type argument when the type parameter includes the `new()` constraint. The compiler can't enforce that all required members are initialized in the generic code.
- The `required` modifier isn't allowed on the declaration for positional parameters on a record. You can add an explicit declaration for a positional property that does include the `required` modifier.

Some types, such as positional records, use a primary constructor to initialize positional properties. If any of those properties include the `required` modifier, the primary constructor adds the SetsRequiredMembers attribute. This indicates that the primary constructor initializes all required members. You can write your own constructor with the System.Diagnostics.CodeAnalysis.SetsRequiredMembersAttribute attribute. However, the compiler doesn't verify that these constructors do initialize all required members. Rather, the attribute asserts to the compiler that the constructor does initialize all required members. The `SetsRequiredMembers` attribute adds these rules to constructors:

- A constructor that chains to another constructor annotated with the `SetsRequiredMembers` attribute, either `this()`, or `base()`, must also include the `SetsRequiredMembers` attribute. That ensures that callers can correctly use all appropriate constructors.
- Copy constructors generated for `record` types have the `SetsRequiredMembers` attribute applied if any of the members are `required`.

> ⚠ **Warning**
>
> The `SetsRequiredMembers` disables the compiler's checks that all `required` members are initialized when an object is created. Use it with caution.

The following code shows a class hierarchy that uses the `required` modifier for the `FirstName` and `LastName` properties:

C#

```csharp
public class Person
{
    public Person() { }

    [SetsRequiredMembers]
    public Person(string firstName, string lastName) =>
        (FirstName, LastName) = (firstName, lastName);

    public required string FirstName { get; init; }
    public required string LastName { get; init; }

    public int? Age { get; set; }
}

public class Student : Person
{
    public Student() : base()
    {
    }

    [SetsRequiredMembers]
    public Student(string firstName, string lastName) :
        base(firstName, lastName)
    {
    }

    public double GPA { get; set; }
}
```

For more information on required members, see the C#11 - Required members feature specification.