

Adding a Marker Attribute



Thomas Claudius Huber

Software Developer

@thomasclaudiush | www.thomasclaudiushuber.com



Module Outline

Understand the plan

Create a `GenerateToString` attribute

- Use the attribute on a class

**Check the attribute
in the source generator**





Understand the Plan



How the ToStringGenerator Works

Person.cs

```
public partial class Person
{
    public string? FirstName { get; set; }
    public string? LastName { get; set; }
}
```

Person.g.cs (generated)

```
public partial class Person
{
    public override string ToString()
    {
        return $"FirstName:{FirstName}; LastName:{LastName}";
    }
}
```



The Plan for the ToStringGenerator

Person.cs

```
[GenerateToString]
public partial class Person
{
    public string? FirstName { get; set; }
    public string? LastName { get; set; }
}
```

Person.g.cs (generated)

```
public partial class Person
{
    public override string ToString()
    {
        return $"FirstName:{FirstName}; LastName:{LastName}";
    }
}
```





Create the Attribute



Create the Attribute

Create the attribute
in the generator
project

Create the attribute
in a separate project

Generate the
attribute in the
target project



Register a post initialization output

Generate the attribute



Use the attribute on a class



Check the syntax target



Check the semantic target



Summary

Create a `GenerateToString` attribute

- Register a post initialization output
- Generate the attribute

Use the attribute on a class

Check the attribute in the `ToStringGenerator`



Up Next:

Using the Semantic Model

