

C-Sharp Mapping for Constants

Here are the sample [constant definitions](#) once more:

Slice

```
const bool      AppendByDefault = true;
const byte      LowerNibble = 0x0f;
const string    Advice = "Don't Panic!";
const short     TheAnswer = 42;
const double    PI = 3.1416;

enum Fruit { Apple, Pear, Orange };
const Fruit     FavoriteFruit = Pear;
```

Here are the generated definitions for these constants:

C#

```
public abstract class AppendByDefault
{
    public const bool value = true;
}

public abstract class LowerNibble
{
    public const byte value = 15;
}

public abstract class Advice
{
    public const string value = "Don't Panic!";
}

public abstract class TheAnswer
{
    public const short value = 42;
}

public abstract class PI
{
    public const double value = 3.1416;
}

public enum Fruit { Apple, Pear, Orange }

public abstract class FavoriteFruit
{
    public const Fruit value = Fruit.Pear;
}
```

As you can see, each Slice constant is mapped to a class with the same name as the constant. The class contains a member named `value` that holds the value of the constant.



The mapping to classes instead of to plain constants is necessary because C# does not permit constant definitions at namespace scope.

See Also

- [Constants and Literals](#)
- [C-Sharp Mapping for Identifiers](#)
- [C-Sharp Mapping for Modules](#)

- [C-Sharp Mapping for Built-In Types](#)
- [C-Sharp Mapping for Enumerations](#)
- [C-Sharp Mapping for Structures](#)
- [C-Sharp Mapping for Sequences](#)
- [C-Sharp Mapping for Dictionaries](#)
- [C-Sharp Collection Comparison](#)
- [C-Sharp Mapping for Exceptions](#)