System.Runtime.CompilerServices.DecimalConstantAttribute Class

```
[ILAsm]
.class public sealed serializable DecimalConstantAttribute extends
System.Attribute

[C#]
public sealed class DecimalConstantAttribute: Attribute
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

Type Attributes:

- CLSCompliantAttribute(false)
- AttributeUsageAttribute(AttributeTargets.Field | AttributeTargets.Parameter, AllowMultiple=false, Inherited=false)

Summary

Stores the value of a `System.Decimal` constant in metadata.

Inherits From: System.Attribute

Library: RuntimeInfrastructure

Thread Safety: All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

Description

[*Note:* This attribute can be applied to fields and parameters.

For more information on storing constants in metadata, see Partition II of the CLI Specification.

The types in `System.Runtime.CompilerServices` are intended primarily for use by compilers, not application programmers. They allow compilers to easily implement certain language features that are not directly visible to programmers.

]

DecimalConstantAttribute(System.Byte, System.Byte, System.UInt32, System.UInt32, System.UInt32) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(unsigned int8
scale, unsigned int8 sign, unsigned int32 hi, unsigned int32 mid,
unsigned int32 low)

[C#]
public DecimalConstantAttribute(byte scale, byte sign, uint hi, uint
mid, uint low)
```

Summary

Constructs a new instance of the `System.Runtime.CompilerServices.DecimalConstantAttribute` class with the specified value.

Parameters

Parameter	Description
scale	A `System.Byte` that specifies the number of digits to the right of the decimal point in the value of the new instance. Valid *scale* values are 0 through 28 inclusive.
sign	A `System.Byte` that specifies the sign of the value of the new instance. Zero indicates a positive value; any non-zero value indicates a negative value.
hi	A `System.UInt32` that specifies the high-order 32 bits of the value of the new instance.
mid	A `System.UInt32` that specifies the middle 32 bits of the value of the new instance.
low	A `System.UInt32` that specifies the low-order 32 bits of the value of the new instance.

Description

[*Note:* For representations of `System.Decimal` whose values can have scales exceeding 255, the constructor `System.Runtime.CompilerServices.DecimalConstantAttribute.System.DecimalConstantAttribute(System.String)` must be used instead.]

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*scale* > 28.

DecimalConstantAttribute(System.String) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(string s)

[C#]
public DecimalConstantAttribute(string s)
```

Summary

Constructs a new instance of the `System.Runtime.CompilerServices.DecimalConstantAttribute` class with the specified value.

Parameters

Parameter	Description
s	A `System.String` containing the value to convert.

Description

The format of *s* is specified by the grammar shown below, in which where the characters in the strings accepted for infinity and nan are not case-sensitive, trailing zero digits in decimal-part are significant, and he purpose of the sign of, and optional digits following, 'NaN' (quiet NaN) and 'sNaN' (signaling NaN) is unspecified. An implementation is free to ignore such signs and digits.

s shall not contain any white space characters.

If an implementation's representation of `System.Decimal` supports NaNs and infinities, that implementation shall correctly handle strings designating those values; otherwise, support for such strings is optional.

Results are rounded to the nearest representable value, and, when a result is equally close to two representable values, to the value that has an even number in the least significant digit position (banker's rounding).

```
numeric-string:: = [sign] numeric-value | [sign] nan
numeric-value:: = decimal-part [exponent-part] | infinity
decimal-part:: = digits '.' [digits] | ['.'] digits
exponent-part:: = indicator [sign] digits
sign:: = '+' | '-'
digits:: = digit [digit]+
digit:: = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
indicator:: = 'e' | 'E'
infinity:: = 'Infinity' | 'Inf'
nan:: = 'NaN' [digits] | 'sNaN' [digits]'
```

Exceptions

Exception	Condition
System.ArgumentNullException	*s* is a null reference.
System.FormatException	*s* is not in the correct format.
System.OverflowException	*s* represents a number greater than `System.Decimal.MaxValue` or less than `System.Decimal.MinValue`.