System.Enum Class

```
[ILAsm]
.class public abstract serializable Enum extends System.ValueType
implements System.IComparable, System.IFormattable

[C#]
public abstract class Enum: ValueType, IComparable, IFormattable
```

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)

Implements:

- **System.IComparable**
- **System.IFormattable**

Summary

Provides support for all enumeration types. Serves as the base class for all enumeration types.

Inherits From: System.ValueType

Library: BCL

Description

A `System.Enum` is a distinct type with named constant members. Each enumeration type has a corresponding integral type called the *underlying type* of the enumeration type. This underlying type is required to be a system-supplied integer type that is large enough to represent all values defined in the enumeration; the field that holds the underlying type must be called `value__`. A `System.Enum` declaration is allowed to explicitly declare any integral type other than `System.Char` as an underlying type. This includes `System.Byte`, `System.SByte`, `System.Int16`, `System.Int32`, `System.Int64`, `System.UInt16`, `System.UInt32`, and `System.UInt64`. A `System.Enum` declaration that does not explicitly declare an underlying type has an underlying type of `System.Int32`.

`System.Enum` derives from `System.ValueType` but is not a value type. Programming languages typically provide syntax to declare sets of a specified enumeration type consisting of named constants and their values.

It is possible to treat instances of a `System.Enum` as bit fields containing multiple values. For more information, see `System.FlagsAttribute`.

[*Note:* `System.Enum` provides methods to compare instances of enumeration types, convert the value of an instance to its `System.String` representation, convert the `System.String` representation of a number to an instance of the enumeration type, and create an instance of a specified enumeration and value.]

Enum.CompareTo(System.Object) Method

```
[ILAsm]
.method public final hidebysig virtual int32 CompareTo(object target)

[C#]
public int CompareTo(object target)
```

Summary

Returns the sort order of the current instance compared to the specified `System.Object`.

Parameters

Parameter	Description
target	An object to compare the current instance to.

Return Value

The return value is a negative number, zero, or a positive number reflecting the sort order of the current instance as compared to *target*. For non-zero return values, the exact value returned by this method is unspecified. The following table defines the return value:

Return Value	Description
A negative integer	The value of the current instance is less than the value of *target*.
Zero	The value of the current instance is equal to the value of *target*.
Any positive integer	The value of the current instance is greater than the value of *target*, or *target* is `null`.

Description

[*Note:* This method is implemented to support the `System.IComparable` interface.]

Exceptions

Exception	Condition
System.ArgumentException	*target* and the current instance are not of the same enumeration type.

Enum.Equals(System.Object) Method

```
[ILAsm]
.method public hidebysig virtual bool Equals(object obj)

[C#]
public override bool Equals(object obj)
```

Summary

Determines whether the current instance and the specified `System.Object` represent the same type and value.

Parameters

Parameter	Description
obj	An object to compare the current instance to.

Return Value

`true` if *obj* is of the same enumeration type and represents the same value as the current instance; otherwise, `false`.

Description

[*Note:* This method overrides `System.Object.Equals`.]

Enum.Format(System.Type, System.Object, System.String) Method

```
[ILAsm]
.method public hidebysig static string Format(class System.Type enumType,
object value, string format)

[C#]
public static string Format(Type enumType, object value, string format)
```

Summary

Converts the specified element of the specified enumeration type to its `System.String` representation using the specified format.

Parameters

Parameter	Description
enumType	A `System.Type` that specifies the type of the enumeration of which *value* is a member.
value	The enumeration element to be converted.
format	A `System.String` that specifies the output format to use.

Return Value

The `System.String` representation of the value of the enumeration element.

Description

For cross-platform portability, the only valid values for *format* are:

Format	Description
"G" or "g"	If value is equal to a defined value of *enumType*, returns the element name defined for value. If the `System.FlagsAttribute` attribute is set on the `System.Enum` declaration and *value* is a built-in integer type and is equal to a summation of enumeration elements, the return value contains the element names in an unspecified order, separated by commas (e.g. "Red, Yellow"). Otherwise,

	value is returned in decimal format.
"X" or "x"	Returns *value* in hexadecimal format, without a leading 0x. The value is padded with leading zeroes to ensure the returned value is at least eight digits in length.
"F" or "f"	Behaves identically to "G", except the `FlagsAttribute` is not required to be present on the `System.Enum` declaration.
"D" or "d"	Returns *value* in decimal format with no leading zeroes.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType*, *value* or *format* is a null reference.
System.ArgumentException	*enumType* is not a `System.Enum`. -or- *value* is neither of type *enumType* nor does it have the same underlying type as *enumType*.
System.FormatException	*format* contains an invalid value.

Example

The following example demonstrates formatting enumeration values.

[C#]

```
using System;
public enum Signs {
  Stop = 1,
  Yield = 2,
  Merge = 4
};
[Flags]
public enum Lights {
```

```
  Red = 1,
  Yellow = 2,
  Green = 4
};
public class EnumCompTo {
  public static void Main() {
   Console.WriteLine(Signs.Format(typeof(Signs), Signs.Merge, "d"));
   Console.WriteLine(Signs.Format(typeof(Signs), 7, "g"));
   Console.WriteLine(Lights.Format(typeof(Lights), Lights.Yellow, "x"));
   Console.WriteLine(Lights.Format(typeof(Lights), 7, "g"));
  }
}
```

The output is

```
4


7


00000002


Red, Yellow, Green


```

Enum.GetHashCode() Method

```
[ILAsm]
.method public hidebysig virtual int32 GetHashCode()

[C#]
public override int GetHashCode()
```

Summary

Generates a hash code for the current instance.

Return Value

A `System.Int32` containing a hash code for the current instance.

Description

The algorithm used to generate the hash code is unspecified.

[*Note:* This method overrides `System.Object.GetHashCode`.]

Enum.GetName(System.Type, System.Object) Method

```
[ILAsm]
.method public hidebysig static string GetName(class System.Type enumType,
object value)

[C#]
public static string GetName(Type enumType, object value)
```

Summary

Retrieves the name of the constant of the specified enumeration type that has the specified value.

Parameters

Parameter	Description
enumType	A `System.Type` that specifies the type of the enumeration.
value	A `System.Object` that contains the integral value or the name of an enumeration constant.

Return Value

A `System.String` containing the name of the enumerated constant in *enumType* whose value is *value*, or a null reference if no such constant is found. If multiple constants have the same value, as to which name is returned for that value, is unspecified.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType* or *value* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`.

	-or- *value* is neither of type *enumType* nor does it have the same underlying type as *enumType*.

Example

The following example demonstrates the System.Enum.GetName method.

[C#]

```
using System;
public enum Signs {
  Stop = 1,
  Yield = 2,
  Merge = 4
};
public class EnumCompTo {
  public static void Main() {
   Console.Write( "The name of the constant with the value 4 is: " );
   Console.WriteLine( "{0}.", Signs.GetName(typeof(Signs), 4));
   Console.Write( "The name of the constant with the value Stop is: " );
   Console.WriteLine( "{0}.", Signs.GetName(typeof(Signs), Signs.Stop ));
  }
}
```

The output is

```
The name of the constant with the value 4 is: Merge.


The name of the constant with the value Stop is: Stop.
```

Enum.GetNames(System.Type) Method

```
[ILAsm]
.method public hidebysig static string[] GetNames(class System.Type
enumType)

[C#]
public static string[] GetNames(Type enumType)
```

Summary

Returns a zero-based, one-dimensional `System.String` array that contains the names of the constants of the specified enumeration type.

Parameters

Parameter	Description
enumType	A `System.Type` that specifies the type of an enumeration.

Return Value

A `System.String` vector of the names of the constants in *enumType*. The elements of the vector are sorted by the values of the enumerated constants. If multiple constants have the same value, the order in which their names appear in the vector, relative to each other, is unspecified.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`.

Example

The following example demonstrates the System.Enum.GetNames method.

[C#]

```
using System;

public enum Colors {
   Red,
   White,
   Blue
};

public class enumGetNames {

   public static void Main() {
      int i = 0;
      String[] strAry = Colors.GetNames( typeof(Colors) );
      foreach (String str in strAry) {
         Console.Write("The value indexed '{0}' ", i++ );
         Console.WriteLine("is {0}.", str);
      }
   }
}
```

The output is

```
The value indexed '0' is Red.


The value indexed '1' is White.


The value indexed '2' is Blue.
```

Enum.GetUnderlyingType(System.Type) Method

```
[ILAsm]
.method public hidebysig static class System.Type GetUnderlyingType(class
System.Type enumType)

[C#]
public static Type GetUnderlyingType(Type enumType)
```

Summary

Returns the underlying type of the specified enumeration type.

Parameters

Parameter	Description
enumType	The System.Type of an enumeration.

Return Value

A System.Type instance that describes the underlying type of *enumType*.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not an enumeration type.

Example

The following example demonstrates the System.Enum.GetUnderlyingType method.

[C#]

```
using System;
public enum Colors {
  Red,
  White,
  Blue
}
public class EnumUnderlyingTypeTest {
  public static void Main() {
     Type t = Colors.GetUnderlyingType( typeof(Colors) );
     Console.WriteLine("The underlying type of Colors is {0}.",
t.ToString());
  }
}
```

The output is

```
The underlying type of Colors is System.Int32.
```

Enum.GetValues(System.Type) Method

```
[ILAsm]
.method public hidebysig static class System.Array GetValues(class
System.Type enumType)

[C#]
public static Array GetValues(Type enumType)
```

Summary

Returns a zero-based, one-dimensional array of the values of the constants of the specified enumeration type.

Parameters

Parameter	Description
enumType	The `System.Type` of an enumeration.

Return Value

A vector of the enumeration type specified by *enumType* containing the values of the constants in *enumType*. The elements of the array are sorted by the values of the enumeration constants. If multiple constants have the same value, the value of each is included in the array.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not an enumeration type.

Example

The following example demonstrates the System.Enum.GetValues method.

[C#]

```
using System;
public enum Colors {
  Red = 1,
  White = 2,
  Blue = 4
}
public class enumGetValues {
   public static void Main() {
      Array valueAry = Enum.GetValues(typeof(Colors));
      foreach (int i in valueAry) {
        Console.WriteLine("The value of element {0} is {1}",
        Enum.Format(typeof(Colors), i, "g"),i);
      }
   }
}
```

The output is

```
The value of element Red is 1.


The value of element White is 2.


The value of element Blue is 4.
```

Enum.IsDefined(System.Type, System.Object) Method

```
[ILAsm]
.method public hidebysig static bool IsDefined(class System.Type enumType,
object value)

[C#]
public static bool IsDefined(Type enumType, object value)
```

Summary

Returns a System.Boolean indicating whether a constant with the specified value exists in the specified enumeration type.

Parameters

Parameter	Description
enumType	A System.Type that describes an enumeration.
value	The constant or value being searched for in *enumType*.

Return Value

true if a constant in the enumeration described by *enumType* has a value equal to *value*; otherwise, false.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType* or *value* is a null reference.
System.ArgumentException	*enumType* is not an enumeration type. -or- The type of *value* is not an *enumType* or an underlying

	type of *enumType*.

Example

The following example demonstrates the System.Enum.IsDefined method.

[C#]

```
using System;
public enum Colors {
  Red = 1,
  White = 2,
  Blue = 4
}
public class enumIsDefined {
  public static void Main() {
     Console.Write("It is {0} ", Enum.IsDefined(typeof(Colors), 1));
     Console.WriteLine("that '1' is defined in 'Colors'.");
     Console.Write("It is {0} ", Enum.IsDefined(typeof(Colors), 3));
     Console.WriteLine("that '3' is defined in 'Colors'.");
  }
}
```

The output is

```
It is True that '1' is defined in 'Colors'.


It is False that '3' is defined in 'Colors'.
```

Enum.Parse(System.Type, System.String) Method

```
[ILAsm]
.method public hidebysig static object Parse(class System.Type enumType,
string value)

[C#]
public static object Parse(Type enumType, string value)
```

Summary

Converts the specified System.String representation of one or more enumerated constants of a specified enumeration type to an equivalent enumerated object.

Parameters

Parameter	Description
enumType	The System.Type of an enumeration.
value	A System.String containing one or more names or a single numeric value to convert. If the string contains more than one name, each name is required to be separated by a comma (','). The names are parsed in a case-sensitive manner. The names or number can be surrounded by any amount of white space.

Return Value

A System.Object of type *enumType* whose values are represented by *value*.

Description

This version of System.Enum.Parse is equivalent to System.Enum.Parse (*enumType*, *value*, false).

Exceptions

Exception	Condition

System.ArgumentNullException	*enumType* or *value* is a null reference.
System.ArgumentException	*enumType* is not a System.Type that describes a System.Enum. -or- *value* is either equal to System.String.Empty or contains only white space. -or- *value* represents one or more names, and at least one name represented by *value* is not of type *enumType*.

Example

The following example demonstrates the System.Enum.Parse method.

[C#]

```
using System;
public enum Colors {
  Red = 1,
  Blue = 2
}
public class enumTest {
  public static void Main() {
    object o = Enum.Parse( typeof (Colors), "Red, Blue");
    Type oType = o.GetType();
    Console.WriteLine("The type of the object returned is
{0}",oType.ToString());
    Array values = Enum.GetValues(oType);
    foreach (Colors c in values) {
      Console.WriteLine(Enum.Format(oType,c,"D"));
    }
  }
}
```

The output is

```
The type of the object returned is Colors


1
```

1 2
2

3

Enum.Parse(System.Type, System.String, System.Boolean) Method

```
[ILAsm]
.method public hidebysig static object Parse(class System.Type enumType,
string value, bool ignoreCase)

[C#]
public static object Parse(Type enumType, string value, bool ignoreCase)
```

Summary

Converts the specified `System.String` representation of one or more enumerated constants of a specified enumeration type to an equivalent enumerated object. This method behaves in a case-sensitive or insensitive manner according to the specified `System.Boolean`.

Parameters

Parameter	Description
enumType	The `System.Type` of an enumeration.
value	A `System.String` containing one or more names or a single numeric value to convert. If the string contains more than one name, each name is required to be separated by a comma (','). The names or number can be surrounded by any amount of white space.
ignoreCase	A `System.Boolean` value. Specify `true` to have names in *value* parsed in a case-insensitive manner. Specify `false` to have names parsed in a case-sensitive manner.

Return Value

A `System.Object` of type *enumType* whose values are represented by *value*.

Exceptions

Exception	Condition

System.ArgumentNullException	*enumType* or *value* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`. -or- *value* is either equal to `System.String.Empty` or contains only white space. -or- *value* represents one or more names, and at least one name represented by *value* is not of type *enumType*.

Example

For an example that demonstrates parsing strings containing enumeration values, see `System.Enum.Parse(System.Type, System.String)`.

Enum.ToObject(System.Type, System.Object) Method

```
[ILAsm]
.method public hidebysig static object ToObject(class System.Type
enumType, object value)

[C#]
public static object ToObject(Type enumType, object value)
```

Summary

Returns an instance of the specified enumeration type set to the specified value.

Parameters

Parameter	Description
enumType	The `System.Type` of the enumeration.
value	The value to set.

Return Value

An enumeration object of type *enumType* whose value is *value*.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`.

Enum.ToObject(System.Type, System.SByte) Method

```
[ILAsm]
.method public hidebysig static object ToObject(class System.Type
enumType, int8 value)

[C#]
public static object ToObject(Type enumType, sbyte value)
```

Summary

Returns an instance of the specified enumeration type set to the specified System.SByte value.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
enumType	The System.Type of an enumeration.
value	The System.SByte value to set.

Return Value

An enumeration object of type *enumType* whose value is *value*.

Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Enum.ToObject(System.Type, System.Int16).

Exceptions

Exception	Condition

System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`.

1
2
3

Enum.ToObject(System.Type, System.Int16) Method

```
[ILAsm]
.method public hidebysig static object ToObject(class System.Type
enumType, int16 value)

[C#]
public static object ToObject(Type enumType, short value)
```

Summary

Returns an instance of the specified enumeration type set to the specified `System.Int16` value.

Parameters

Parameter	Description
enumType	The `System.Type` of an enumeration.
value	The `System.Int16` value to set.

Return Value

An enumeration object of type *enumType* whose value is *value*.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`.

Enum.ToObject(System.Type, System.UInt64) Method

```
[ILAsm]
.method public hidebysig static object ToObject(class System.Type
enumType, unsigned int64 value)

[C#]
public static object ToObject(Type enumType, ulong value)
```

Summary

Returns an instance of the specified enumeration type set to the specified `System.UInt64` value.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
enumType	The `System.Type` of an enumeration.
value	The `System.UInt64` value to set.

Return Value

An enumeration object of type *enumType* whose value is *value*.

Description

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Enum.ToObject(System.Type, System.Int64 )`.

Exceptions

Exception	Condition

System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`.

1
2
3

Enum.ToObject(System.Type, System.Int64) Method

```
[ILAsm]
.method public hidebysig static object ToObject(class System.Type
enumType, int64 value)

[C#]
public static object ToObject(Type enumType, long value)
```

Summary

Returns an instance of the specified enumeration type set to the specified `System.Int64` value.

Parameters

Parameter	Description
enumType	The `System.Type` of an enumeration.
value	The `System.Int64` value to set.

Return Value

An enumeration object of type *enumType* whose value is *value*.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`.

Enum.ToObject(System.Type, System.UInt32) Method

```
[ILAsm]
.method public hidebysig static object ToObject(class System.Type
enumType, unsigned int32 value)

[C#]
public static object ToObject(Type enumType, uint value)
```

Summary

Returns an instance of the specified enumeration type set to the specified `System.UInt32` value.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
enumType	The `System.Type` of an enumeration.
value	The `System.UInt32` value to set.

Return Value

An enumeration object of type *enumType* whose value is *value*.

Description

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Enum.ToObject(System.Type, System.Int64)`.

Exceptions

Exception	Condition

System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`.

1
2
3

Enum.ToObject(System.Type, System.UInt16) Method

```
[ILAsm]
.method public hidebysig static object ToObject(class System.Type
enumType, unsigned int16 value)

[C#]
public static object ToObject(Type enumType, ushort value)
```

Summary

Returns an instance of the specified enumeration type set to the specified `System.UInt16` value.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
enumType	The `System.Type` of an enumeration.
value	The `System.UInt16` value to set.

Return Value

An enumeration object of type *enumType* whose value is *value*.

Description

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Enum.ToObject(System.Type, System.Int32)`.

Exceptions

Exception	Condition

System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`.

1
2
3

Enum.ToObject(System.Type, System.Byte) Method

```
[ILAsm]
.method public hidebysig static object ToObject(class System.Type
enumType, unsigned int8 value)

[C#]
public static object ToObject(Type enumType, byte value)
```

Summary

Returns an instance of the specified enumeration type set to the specified System.Byte value.

Parameters

Parameter	Description
enumType	The System.Type of an enumeration.
value	The System.Byte value to set.

Return Value

An enumeration object of type *enumType* whose value is *value*.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not a System.Type that describes a System.Enum.

Enum.ToObject(System.Type, System.Int32) Method

```
[ILAsm]
.method public hidebysig static object ToObject(class System.Type
enumType, int32 value)

[C#]
public static object ToObject(Type enumType, int value)
```

Summary

Returns an instance of the specified enumeration type set to the specified `System.Int32` value.

Parameters

Parameter	Description
enumType	The `System.Type` of an enumeration.
value	The `System.Int32` value to set.

Return Value

An enumeration object of type *enumType* whose value is *value*.

Exceptions

Exception	Condition
System.ArgumentNullException	*enumType* is a null reference.
System.ArgumentException	*enumType* is not a `System.Type` that describes a `System.Enum`.

Enum.ToString(System.String) Method

```
[ILAsm]
.method public hidebysig instance string ToString(string format)

[C#]
public string ToString(string format)
```

Summary

Converts the value of the current instance to its equivalent `System.String` representation using the specified format.

Parameters

Parameter	Description
format	A `System.String` that specifies the format to use when converting the current instance to a `System.String`. Specify one of the following values in upper or lowercase: "G", "D", "X", or "F".

Return Value

The `System.String` representation of the value of the current instance as specified by *format*.

Description

If *format* is a null reference or an empty string, the return value is formatted using the general format specifier ("G").

[*Note:* For details on the format specifiers used with an enumeration object, see `System.Enum.Format`.]

Exceptions

Exception	Condition

System.FormatException	*format* contains an invalid value.

Enum.ToString() Method

```
[ILAsm]
.method public hidebysig virtual string ToString()

[C#]
public override string ToString()
```

Summary

Converts the name of the value of the current instance to its equivalent `System.String` representation.

Return Value

The `System.String` representation of the named constant specified by the current instance.

Description

This version of `System.Enum.ToString` is equivalent to `System.Enum.ToString` ("G", `null`).

This method returns the same value as `System.Enum.Format` with the "g" or "G" format option.

An instance of an enumeration is set to a named constant value. This method returns the name of the constant an instance is set to, not the value itself.

Enum.ToString(System.String, System.IFormatProvider) Method

```
[ILAsm]
.method public final hidebysig virtual string ToString(string format,
class System.IFormatProvider provider)

[C#]
public string ToString(string format, IFormatProvider provider)
```

Summary

Converts the numeric value of the current instance to its equivalent System.String representation using the specified format.

Parameters

Parameter	Description
format	A System.String that specifies the format to use when converting the current instance to a System.String. Specify one of the following values in upper or lowercase: "G", "D", "X", or "F".
provider	An object that implements the System.IFormatProvider interface or a null reference. The System.IFormatProvider is not used in the implementation of this method. [*Note:* There is no System.IFormatProvider that corresponds to a System.Enum object; therefore, *provider* is not utilized by this method, and any System.IFormatProvider can be passed as a parameter.]

Return Value

The System.String representation of the value of the current instance as specified by *format*.

Description

If *format* is a null reference or an empty string, the return value is formatted using the general format specifier ("G").

[*Note:* For details on the format specifiers used with an enumeration object, see System.Enum.Format.

This method is implemented to support the `System.IFormattable` interface.

]

Exceptions

Exception	Condition
System.FormatException	*format* does not contain a valid format specifier.

Enum.ToString(System.IFormatProvider) Method

```
[ILAsm]
.method public final hidebysig virtual string ToString(class
System.IFormatProvider provider)

[C#]
public string ToString(IFormatProvider provider)
```

Summary

Converts the name of the value of the current instance to its equivalent `System.String` representation.

Parameters

Parameter	Description
provider	An object that implements the `System.IFormatProvider` interface or a null reference. The `System.IFormatProvider` is not used in the implementation of this method. [*Note:* There is no `System.IFormatProvider` that corresponds to a `System.Enum` object; therefore, *provider* is not utilized by this method, and any `System.IFormatProvider` can be passed as a parameter.]

Return Value

The `System.String` representation of the name of the value of the current instance.

Description

This method is equivalent to the version of `System.Enum.ToString` that takes no arguments.