

System.Guid Structure

```
[ILAsm]
.class public sequential ansi serializable sealed beforefieldinit Guid
extends System.ValueType implements System.IFormattable,
System.IComparable, System.IComparable`1<valuetype System.Guid>,
System.IEquatable`1<valuetype System.Guid>

[C#]
public struct Guid: IComparable, IComparable<Guid>, IEquatable<Guid>,
IFormattable
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 4.0.0.0
- *Attributes:*
 - CLSCompliantAttribute(true)

Implements:

- **System.IComparable**
- **System.IComparable<System.Guid>**
- **System.IEquatable<System.Guid>**
- **System.IFormattable**

Summary

Represents a globally unique identifier (GUID).

Inherits From: System.ValueType

Library: BCL

Description

A GUID is a 128-bit integer (16 bytes) that can be used across all computers and networks wherever a unique identifier is required. Such an identifier has a very low probability of being duplicated.

Guid(System.Byte[]) Constructor

```
[ILAsm]
.method public hidebysig specialname rtspecialname instance void
.ctor(uint8[] b) cil managed

[C#]
public Guid (byte[] b)
```

Summary

Initializes a new instance of the `System.Guid` class using the specified array of bytes.

Parameters

Parameter	Description
<i>b</i>	A 16-element byte array containing values with which to initialize the GUID.

Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>b</i> is null.
<code>System.ArgumentException</code>	<i>b</i> is not 16 bytes long.

Guid(System.String) Constructor

```
[ILAsm]
.method public hidebysig specialname rtspecialname instance void
.ctor(string g) cil managed

[C#]
public Guid (string g)
```

Summary

Initializes a new instance of the `System.Guid` class using the value represented by the specified string.

Parameters

Parameter	Description
<i>g</i>	A string that contains a GUID in one of the following formats ("d" represents a hexadecimal digit whose case is ignored): 32 contiguous digits: ddddddddddddddddddddddddddddddd -or- Groups of 8, 4, 4, 4, and 12 digits with hyphens between the groups. The entire GUID can optionally be enclosed in matching braces or parentheses: dddddddd-dddd-dddd-dddd-dddddddddddd -or- { dddddddd-dddd-dddd-dddd-dddddddddddd } -or- (dddddddd-dddd-dddd-dddd-dddddddddddd) -or- Groups of 8, 4, and 4 digits, and a subset of eight groups of 2 digits, with each group prefixed by "0x" or "0X", and separated by commas. The entire GUID, as well as the subset, is enclosed in matching braces: { 0xdddddddd, 0xdddd, 0xdddd, { 0xdd, 0xdd, 0xdd, 0xdd, 0xdd, 0xdd, 0xdd, 0xdd } } All braces, commas, and "0x" prefixes are required. All embedded spaces are ignored. All leading zeros in a group are ignored. The digits shown in a group are the maximum number of meaningful digits that can appear in that group. You can specify from 1 to the number of digits shown for a group. The specified digits are assumed to be the low-order digits of the group.

Description

The alphabetic hexadecimal digits in the *g* parameter can be uppercase or lowercase. For example, the following strings represent the same GUID.

```
"ca761232ed4211cebacd00aa0057b223"
"CA761232-ED42-11CE-BACD-00AA0057B223"
"{ CA761232-ED42-11CE-BACD-00AA0057B223 }"
"(CA761232-ED42-11CE-BACD-00AA0057B223)"
```

1
2 "{0xCA761232, 0xED42, 0x11CE, {0xBA, 0xCD, 0x00, 0xAA, 0x00, 0x57, 0xB2,
3 0x23}}"

4 **Exceptions**

Exception	Condition
System.ArgumentNullException	g is null.
System.FormatException	The format of g is invalid.
System.OverflowException	The format of g is invalid.
System.Exception	An internal type conversion error occurred.

5

6

Guid(System.Int32, System.Int16, System.Int16, System.Byte[]) Constructor

```
[ILAsm]  
.method public hidebysig specialname rtspecialname instance void  
.ctor(int32 a, int16 b, int16 c, uint8[] d) cil managed  
  
[C#]  
public Guid (int a, short b, short c, byte[] d)
```

Summary

Initializes a new instance of the `System.Guid` class using the specified integers and byte array.

Parameters

Parameter	Description
<i>a</i>	The first 4 bytes of the GUID.
<i>b</i>	The next 2 bytes of the GUID.
<i>c</i>	The next 2 bytes of the GUID.
<i>d</i>	The remaining 8 bytes of the GUID.

Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>d</i> is null.
<code>System.ArgumentException</code>	<i>d</i> is not 8 bytes long.

Guid(System.Int32, System.Int16, System.Int16, System.Byte, System.Byte, System.Byte, System.Byte, System.Byte, System.Byte) Constructor

```
[ILAsm]
.method public hidebysig specialname rtspecialname instance void
.ctor(int32 a, int16 b, int16 c, uint8 d, uint8 e, uint8 f, uint8 g, uint8
h, uint8 i, uint8 j, uint8 k) cil managed

[C#]
public Guid (int a, short b, short c, byte d, byte e, byte f, byte g, byte
h, byte i, byte j, byte k)
```

Summary

Initializes a new instance of the `System.Guid` class using the specified integers and bytes.

Parameters

Parameter	Description
<i>a</i>	The first 4 bytes of the GUID.
<i>b</i>	The next 2 bytes of the GUID.
<i>c</i>	The next 2 bytes of the GUID.
<i>d</i>	The next byte of the GUID.
<i>e</i>	The next byte of the GUID.
<i>f</i>	The next byte of the GUID.
<i>g</i>	The next byte of the GUID.
<i>h</i>	The next byte of the GUID.
<i>i</i>	The next byte of the GUID.
<i>j</i>	The next byte of the GUID.

<i>k</i>	The next byte of the GUID.
----------	----------------------------

1

2 **Description**

3 Specifying individual bytes in this manner can be used to circumvent byte order
 4 restrictions (big-endian or little-endian byte order) on particular types of computers.

5

Guid(System.UInt32, System.UInt16, System.UInt16, System.Byte, System.Byte, System.Byte, System.Byte, System.Byte, System.Byte, System.Byte)

Constructor

```
[ILAsm]
.method public hidebysig specialname rtspecialname instance void
.ctor(uint32 a, uint16 b, uint16 c, uint8 d, uint8 e, uint8 f, uint8 g,
uint8 h, uint8 i, uint8 j, uint8 k) cil managed

[C#]
public Guid (uint a, ushort b, ushort c, byte d, byte e, byte f, byte g,
byte h, byte i, byte j, byte k)
```

Summary

Initializes a new instance of the `System.Guid` class using the specified unsigned integers and bytes.

Type Attributes:

- `CLSCompliantAttribute(false)`

Parameters

Parameter	Description
<i>a</i>	The first 4 bytes of the GUID.
<i>b</i>	The next 2 bytes of the GUID.
<i>c</i>	The next 2 bytes of the GUID.
<i>d</i>	The next byte of the GUID.
<i>e</i>	The next byte of the GUID.
<i>f</i>	The next byte of the GUID.
<i>g</i>	The next byte of the GUID.
<i>h</i>	The next byte of the GUID.

<i>i</i>	The next byte of the GUID.
<i>j</i>	The next byte of the GUID.
<i>k</i>	The next byte of the GUID.

1

2 **Description**

3 Specifying the bytes in this manner avoids endianness issues.

4

Guid.Empty Field

```
[ILAsm]  
.field public static initonly valuetype System.Guid Empty  
  
[C#]  
public static readonly Guid Empty
```

Summary

A read-only instance of the `System.Guid` class whose value is guaranteed to be all zeros.

Guid.CompareTo(System.Guid) Method

```
[ILAsm]  
.method public hidebysig newslot virtual final instance int32  
CompareTo(valuetype System.Guid 'value') cil managed  
  
[C#]  
public int CompareTo (Guid value)
```

Summary

Compares this instance to a specified `System.Guid` object and returns an indication of their relative values.

Parameters

Parameter	Description
<i>value</i>	An object to compare to this instance.

Return Value

A signed number indicating the relative values of this instance and *value*.

Return value	Description
A negative integer	This instance is less than <i>value</i> .
Zero	This instance is equal to <i>value</i> .
A positive integer	This instance is greater than <i>value</i> .

Guid.CompareTo(System.Object) Method

```
[ILAsm]  
.method public hidebysig newslot virtual final instance int32  
CompareTo(object 'value') cil managed  
  
[C#]  
public int CompareTo (object value)
```

Summary

Compares this instance to a specified object and returns an indication of their relative values.

Parameters

Parameter	Description
<i>value</i>	An object to compare, or null.

Return Value

A signed number indicating the relative values of this instance and *value*.

Return value	Description
A negative integer	This instance is less than <i>value</i> .
Zero	This instance is equal to <i>value</i> .
A positive integer	This instance is greater than <i>value</i> , or <i>value</i> is null.

Description

Any instance of `System.Guid`, regardless of its value, is considered greater than null.

The *value* parameter must be null or an instance of `System.Guid`; otherwise, an exception is thrown.

Exceptions

Exception	Condition
-----------	-----------

System.ArgumentException

value is not a System.Guid.

1

2

Guid.Equals(System.Guid) Method

```
[ILAsm]  
.method public hidebysig newslot virtual final instance bool  
Equals(valuetype System.Guid g) cil managed  
  
[C#]  
public bool Equals (Guid g)
```

Summary

Returns a value indicating whether this instance and a specified `System.Guid` object represent the same value.

Parameters

Parameter	Description
<i>g</i>	An object to compare to this instance.

Return Value

true if *g* is equal to this instance; otherwise, false.

Guid.Equals(System.Object) Method

```
[ILAsm]  
.method public hidebysig virtual instance bool Equals(object o) cil  
managed  
  
[C#]  
public override bool Equals (object o)
```

Summary

Returns a value that indicates whether this instance is equal to a specified object.

Parameters

Parameter	Description
<i>o</i>	The object to compare with this instance.

Return Value

true if *o* is a System.Guid that has the same value as this instance; otherwise, false.

Guid.GetHashCode() Method

```
[ILAsm]  
.method public hidebysig virtual instance int32 GetHashCode() cil managed  
  
[C#]  
public override int GetHashCode ()
```

Summary

Returns the hash code for this instance.

Return Value

The hash code for this instance.

Guid.NewGuid() Method

```
[ILAsm]  
.method public hidebysig static valuetype System.Guid NewGuid() cil  
managed  
  
[C#]  
public static Guid NewGuid ()
```

Summary

Initializes a new instance of the `System.Guid` class.

Return Value

A new GUID object.

Description

This is a convenient `static` method that you can call to get a new `System.Guid`.

The chance that the value of the new `System.Guid` will be all zeros or equal to any other `System.Guid` is very low.

Guid.op_Equality(System.Guid, System.Guid)

Method

```
[ILAsm]  
.method public hidebysig specialname static bool op_Equality(valuetype  
System.Guid a, valuetype System.Guid b) cil managed  
  
[C#]  
public static bool op_Equality (Guid a, Guid b)
```

Summary

Indicates whether the values of two specified `System.Guid` objects are equal.

Parameters

Parameter	Description
<i>a</i>	The first object to compare.
<i>b</i>	The second object to compare.

Return Value

true if *a* and *b* are equal; otherwise, false.

Guid.op_Inequality(System.Guid, System.Guid) Method

```
[ILAsm]  
.method public hidebysig specialname static bool op_Inequality(valuetype  
System.Guid a, valuetype System.Guid b) cil managed  
  
[C#]  
public static bool op_Inequality (Guid a, Guid b)
```

Summary

Indicates whether the values of two specified `System.Guid` objects are not equal.

Parameters

Parameter	Description
<i>a</i>	The first object to compare.
<i>b</i>	The second object to compare.

Return Value

true if *a* and *b* are not equal; otherwise, false.

Guid.Parse(System.String) Method

```
[ILAsm]  
.method public hidebysig static valuetype System.Guid Parse(string input)  
cil managed  
  
[C#]  
public static Guid Parse (string input)
```

Summary

Converts the string representation of a GUID to the equivalent `System.Guid` structure.

Parameters

Parameter	Description
<i>input</i>	The GUID to convert.

Return Value

A structure that contains the value that was parsed.

Description

Use `System.Guid.TryParse` method to catch any unsuccessful parse operations.

Exceptions

Exception	Condition
System.ArgumentNullException	<i>input</i> is null.
System.FormatException	<i>input</i> is not in a recognized format.
System.Exception	An internal type conversion error occurred.

Guid.ParseExact(System.String, System.String) Method

```
[ILAsm]  
.method public hidebysig static valuetype System.Guid ParseExact(string  
input, string format) cil managed  
  
[C#]  
public static Guid ParseExact (string input, string format)
```

Summary

Converts the string representation of a GUID to the equivalent `System.Guid` structure, provided that the string is in the specified format.

Parameters

Parameter	Description
<i>input</i>	The GUID to convert.
<i>format</i>	One of the following specifiers that indicates the exact format to use when interpreting <i>input</i> : "N", "D", "B", "P", or "X".

Return Value

A structure that contains the value that was parsed.

Description

The following table shows the accepted format specifiers for the *format* parameter. "0" represents a digit; hyphens ("-"), braces ("{" , "}"), and parentheses ("(" , ")") appear as shown.

Specifier	Format of the <i>input</i> parameter
N	32 digits: 00000000000000000000000000000000
D	32 digits separated by hyphens: 00000000-0000-0000-0000-000000000000

B	32 digits separated by hyphens, enclosed in braces: {00000000-0000-0000-0000-000000000000}
P	32 digits separated by hyphens, enclosed in parentheses: (00000000-0000-0000-0000-000000000000)
X	Four hexadecimal values enclosed in braces, where the fourth value is a subset of eight hexadecimal values that is also enclosed in braces: {0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}}

1

2 Exceptions

Exception	Condition
System.ArgumentNullException	<i>input</i> or <i>format</i> is null.
System.FormatException	<i>input</i> is not in a recognized format.
System.Exception	An internal type conversion error occurred.

3

4

Guid.ToArray() Method

```
[ILAsm]  
.method public hidebysig instance uint8[] ToByteArray() cil managed  
  
[C#]  
public byte[] ToByteArray ()
```

Summary

Returns a 16-element byte array that contains the value of this instance.

Return Value

A 16-element byte array.

Guid.ToString() Method

```
[ILAsm]  
.method public hidebysig virtual instance string ToString() cil managed  
  
[C#]  
public override string ToString ()
```

Summary

Returns a string representation of the value of this instance in registry format.

Return Value

The value of this `System.Guid`, formatted as follows:

xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

where the value of the GUID is represented as a series of lowercase hexadecimal digits in groups of 8, 4, 4, 4, and 12 digits and separated by hyphens. An example of a return value is "382c74c3-721d-4f34-80e5-57657b6cbc27".

Description

This method provides a default GUID format that is sufficient for typical use; however, other versions of this method that take a *format* parameter provide a few common format variations.

Guid.ToString(System.String) Method

```
[ILAsm]  
.method public hidebysig instance string ToString(string format) cil  
managed  
  
[C#]  
public string ToString (string format)
```

Summary

Returns a string representation of the value of this `System.Guid` instance, according to the provided format specifier.

Parameters

Parameter	Description
<i>format</i>	A single format specifier that indicates how to format the value of this <code>System.Guid</code> . The <i>format</i> parameter can be "N", "D", "B", "P", or "X". If <i>format</i> is null or an empty string (""), "D" is used.

Return Value

The value of this `System.Guid`, represented as a series of lowercase hexadecimal digits in the specified format.

Description

The following table shows the accepted format specifiers for the *format* parameter. "0" represents a digit; hyphens ("-"), braces ("{" , "}"), and parentheses ("(" , ")") appear as shown.

Specifier	Format of return value
N	32 digits: 00000000000000000000000000000000
D	32 digits separated by hyphens: 00000000-0000-0000-0000-000000000000
B	32 digits separated by hyphens, enclosed in braces:

	{00000000-0000-0000-0000-000000000000}
P	32 digits separated by hyphens, enclosed in parentheses: (00000000-0000-0000-0000-000000000000)
X	Four hexadecimal values enclosed in braces, where the fourth value is a subset of eight hexadecimal values that is also enclosed in braces: {0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}}

1

2 Exceptions

Exception	Condition
System.FormatException	The value of <i>format</i> is not <code>null</code> , an empty string (<code>""</code>), "N", "D", "B", "P", or "X".

3

4

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

```
[ILAsm]  
.method public hidebysig newslot virtual final instance string  
ToString(string format, class System.IFormatProvider provider) cil managed  
  
[C#]  
public string ToString (string format, IFormatProvider provider)
```

Summary

Returns a string representation of the value of this instance of the `System.Guid` class, according to the provided format specifier and culture-specific format information.

Parameters

Parameter	Description
<i>format</i>	A single format specifier that indicates how to format the value of this <code>System.Guid</code> . The <i>format</i> parameter can be "N", "D", "B", "P", or "X". If <i>format</i> is null or an empty string (""), "D" is used.
<i>provider</i>	(Reserved) An object that supplies culture-specific formatting services.

Return Value

The value of this `System.Guid`, represented as a series of lowercase hexadecimal digits in the specified format.

Description

The *provider* parameter is reserved for future use and does not contribute to the execution of this method. A null reference can be coded for this parameter.

The following table shows the accepted format specifiers for the *format* parameter. "0" represents a digit; hyphens ("-"), braces ("{" , "}"), and parentheses "(" , ")" appear as shown.

Specifier	Format of return value
N	32 digits: 00000000000000000000000000000000

D	32 digits separated by hyphens: 00000000-0000-0000-0000-000000000000
B	32 digits separated by hyphens, enclosed in braces: {00000000-0000-0000-0000-000000000000}
P	32 digits separated by hyphens, enclosed in parentheses: (00000000-0000-0000-0000-000000000000)
X	Four hexadecimal values enclosed in braces, where the fourth value is a subset of eight hexadecimal values that is also enclosed in braces: {0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}}

1

2 Exceptions

Exception	Condition
System.FormatException	The value of <i>format</i> is not <code>null</code> , an empty string (<code>""</code>), "N", "D", "B", "P", or "X".

3

4

Guid.TryParse(System.String, System.Guid&) Method

```
[ILAsm]
.method public hidebysig static bool TryParse(string input, [out]
valuetype System.Guid& result) cil managed

[C#]
public static bool TryParse (string input, out Guid result)
```

Summary

Converts the string representation of a GUID to the equivalent `System.Guid` structure.

Parameters

Parameter	Description
<i>input</i>	The GUID to convert.
<i>result</i>	The structure that will contain the parsed value.

Return Value

`true` if the parse operation was successful; otherwise, `false`.

Description

This method returns `false` if *input* is `null` or not in a recognized format, and does not return an exception.

Guid.TryParseExact(System.String, System.String, System.Guid&) Method

```
[ILAsm]
.method public hidebysig static bool TryParseExact(string input, string
format, [out] valuetype System.Guid& result) cil managed

[C#]
public static bool TryParseExact (string input, string format, out Guid
result)
```

Summary

Converts the string representation of a GUID to the equivalent `System.Guid` structure, provided that the string is in the specified format.

Parameters

Parameter	Description
<i>input</i>	The GUID to convert.
<i>format</i>	One of the following specifiers that indicates the exact format to use when interpreting <i>input</i> : "N", "D", "B", "P", or "X".
<i>result</i>	The structure that will contain the parsed value.

Return Value

`true` if the parse operation was successful; otherwise, `false`.

Description

This method returns `false` if *input* is null or not in a recognized format, and does not return an exception.

The following table shows the accepted format specifiers for the *format* parameter. "0" represents a digit; hyphens ("-"), braces ("{" , "}"), and parentheses "(" , ")" appear as shown.

Specifier	Format of the <i>input</i> parameter
N	32 digits: 00000000000000000000000000000000

D	32 digits separated by hyphens: 00000000-0000-0000-0000-000000000000
B	32 digits separated by hyphens, enclosed in braces: {00000000-0000-0000-0000-000000000000}
P	32 digits separated by hyphens, enclosed in parentheses: (00000000-0000-0000-0000-000000000000)
X	Four hexadecimal values enclosed in braces, where the fourth value is a subset of eight hexadecimal values that is also enclosed in braces: {0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} }

1

2