

1 System.Guid Structure

```
2 [ILAsm]
3 .class public sequential ansi serializable sealed beforefieldinit Guid
4 extends System.ValueType implements System.IFormattable,
5 System.IComparable, System.IComparable`1<valuetype System.Guid>,
6 System.IEquatable`1<valuetype System.Guid>
7
8 [C#]
9 public struct Guid: IComparable, IComparable<Guid>, IEquatable<Guid>,
IFormattable
```

10 Assembly Info:

- 11 • *Name:* mscorlib
- 12 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 13 • *Version:* 4.0.0.0
- 14 • *Attributes:*
 - 15 ○ CLSCompliantAttribute(true)

16 Implements:

- 17 • **System.IComparable**
- 18 • **System.IComparable<System.Guid>**
- 19 • **System.IEquatable<System.Guid>**
- 20 • **System.IFormattable**

21 Summary

22 Represents a globally unique identifier (GUID).

23 Inherits From: System.ValueType

24

25 **Library:** BCL

26

27 Description

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

31

1 Guid(System.Byte[]) Constructor

```
2 [ILAsm]  
3 .method public hidebysig specialname rtspecialname instance void  
4 .ctor(uint8[] b) cil managed  
  
5 [C#]  
6 public Guid (byte[] b)
```

7 Summary

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

9 Parameters

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

10

11 Exceptions

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

12

13

1 Guid(System.String) Constructor

```
2 [ILAsm]  
3 .method public hidebysig specialname rtspecialname instance void  
4 .ctor(string g) cil managed  
  
5 [C#]  
6 public Guid (string g)
```

7 Summary

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

10 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: ddddddddddddddddddddddddddd -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- (ddddddd-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.

11 12 Description

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

- 15 "ca761232ed4211cebacd00aa0057b223"
- 16 "CA761232-ED42-11CE-BACD-00AA0057B223"
- 17 "{ CA761232-ED42-11CE-BACD-00AA0057B223 }"
- 18 "(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	<i>g</i> is null.
System.FormatException	The format of <i>g</i> is invalid.
System.OverflowException	The format of <i>g</i> is invalid.
System.Exception	An internal type conversion error occurred.

5

6

1 Guid(System.Int32, System.Int16, 2 System.Int16, System.Byte[]) Constructor

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

8 Summary

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

11 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.

12

13 Exceptions

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

14

15

1 **Guid(System.Int32, System.Int16,**
2 **System.Int16, System.Byte, System.Byte,**
3 **System.Byte, System.Byte, System.Byte,**
4 **System.Byte, System.Byte, System.Byte)**
5 **Constructor**

```
6 [ILAsm]  
7 .method public hidebysig specialname rtspecialname instance void  
8 .ctor(int32 a, int16 b, int16 c, uint8 d, uint8 e, uint8 f, uint8 g, uint8  
9 h, uint8 i, uint8 j, uint8 k) cil managed  
  
10 [C#]  
11 public Guid (int a, short b, short c, byte d, byte e, byte f, byte g, byte  
12 h, byte i, byte j, byte k)
```

13 **Summary**

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

16 **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

1 **Guid(System.UInt32, System.UInt16,**
2 **System.UInt16, System.Byte, System.Byte,**
3 **System.Byte, System.Byte, System.Byte,**
4 **System.Byte, System.Byte, System.Byte)**
5 **Constructor**

```
6 [ILAsm]  
7 .method public hidebysig specialname rtspecialname instance void  
8 .ctor(uint32 a, uint16 b, uint16 c, uint8 d, uint8 e, uint8 f, uint8 g,  
9 uint8 h, uint8 i, uint8 j, uint8 k) cil managed  
10 [C#]  
11 public Guid (uint a, ushort b, ushort c, byte d, byte e, byte f, byte g,  
12 byte h, byte i, byte j, byte k)
```

13 **Summary**

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

16 **Type Attributes:**

- 17 • `CLSCompliantAttribute(false)`

18 **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

1 Guid.Empty Field

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

6 Summary

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

9

1 Guid.CompareTo(System.Guid) Method

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

7 Summary

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

10 Parameters

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

11 12 Return Value

13 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> .

14
15

1 Guid.CompareTo(System.Object) Method

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

7 Summary

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

10 Parameters

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

11 12 Return Value

13 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.

14 15 Description

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

17
18 The *value* parameter must be null or an instance of `System.Guid`; otherwise, an
19 exception is thrown.

20 Exceptions

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

System.ArgumentException

value is not a System.Guid.

1

2

1 **Guid.Equals(System.Guid) Method**

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

7 **Summary**

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

10 **Parameters**

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

11

12 **Return Value**

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

14

1 Guid.Equals(System.Object) Method

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

7 Summary

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

9 Parameters

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

10

11 Return Value

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

13

1 Guid.GetHashCode() Method

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

6 Summary

7 Returns the hash code for this instance.

8 Return Value

9 The hash code for this instance.

10

1 Guid.NewGuid() Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Guid NewGuid() cil  
4 managed  
  
5 [C#]  
6 public static Guid NewGuid ()
```

7 Summary

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

9 Return Value

10 A new GUID object.

11 Description

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

13
14 The chance that the value of the new `System.Guid` will be all zeros or equal to any other
15 `System.Guid` is very low.

16

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

2 Method

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

8 Summary

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

10 Parameters

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

11

12 Return Value

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

14

1 Guid.op_Inequality(System.Guid, 2 System.Guid) Method

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

8 Summary

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

10 Parameters

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

11

12 Return Value

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

14

1 Guid.Parse(System.String) Method

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

7 Summary

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

9 Parameters

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

10 11 Return Value

12 A structure that contains the value that was parsed.

13 Description

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

15 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.

16
17

1 Guid.ParseExact(System.String, 2 System.String) Method

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

8 Summary

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

11 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".

12

13 Return Value

14 A structure that contains the value that was parsed.

15 Description

16 The following table shows the accepted format specifiers for the *format* parameter. "0"
17 represents a digit; hyphens ("-"), braces ("{", "}"), and parentheses ("(", ")") appear as
18 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

1 Guid.ToByteArray() Method

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

6 Summary

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

8 Return Value

9 A 16-element byte array.

10

1 Guid.ToString() Method

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

6 Summary

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

8 Return Value

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

10
11 xxxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx

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

16 Description

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

20

1 Guid.ToString(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig instance string ToString(string format) cil  
4 managed  
  
5 [C#]  
6 public string ToString (string format)
```

7 Summary

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

10 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.

11 12 Return Value

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

15 Description

16 The following table shows the accepted format specifiers for the *format* parameter. "0"
17 represents a digit; hyphens ("-"), braces ("{" , "}") , and parentheses ("(" , ")") appear as
18 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

1 Guid.TryParse(System.String, System.Guid&) 2 Method

```
3 [ILAsm]  
4 .method public hidebysig static bool TryParse(string input, [out]  
5 valuetype System.Guid& result) cil managed  
  
6 [C#]  
7 public static bool TryParse (string input, out Guid result)
```

8 Summary

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

10 Parameters

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

11

12 Return Value

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

14 Description

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

17

1 Guid.TryParseExact(System.String, 2 System.String, System.Guid&) Method

```
3 [ILAsm]  
4 .method public hidebysig static bool TryParseExact(string input, string  
5 format, [out] valuetype System.Guid& result) cil managed  
  
6 [C#]  
7 public static bool TryParseExact (string input, string format, out Guid  
8 result)
```

9 Summary

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

12 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.

13 14 Return Value

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

16 Description

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

19
20 The following table shows the accepted format specifiers for the *format* parameter. "0"
21 represents a digit; hyphens ("-"), braces ("{" , "}"), and parentheses ("(", ")") appear as
22 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