

1 System.UIntPtr Structure

```
2 [ILAsm]  
3 .class public sequential sealed serializable UIntPtr extends  
4 System.ValueType  
5 [C#]  
6 public struct UIntPtr
```

7 Assembly Info:

- 8 • *Name:* mscorlib
- 9 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 10 • *Version:* 2.0.x.x
- 11 • *Attributes:*
 - 12 ○ CLSCompliantAttribute(true)

13 Type Attributes:

- 14 • CLSCompliantAttribute(false)

15 Summary

16 An implementation-specific type that is used to represent a pointer or a handle.

17 Inherits From: System.ValueType

18

19 **Library:** RuntimeInfrastructure

20

21 **Thread Safety:** This type is safe for multithreaded operations.

22

23 Description

24 The `System.UIntPtr` type is designed to be an implementation-sized pointer. An
25 instance of this type is expected to be the size of a native unsigned int for the
26 current implementation.

27

28 For more information on the native unsigned int type, see Partition II of the CLI
29 Specification.

30

31 [*Note:* `System.UIntPtr` instances can also be used to hold handles.

32

33 The `System.IntPtr` type is CLS-compliant while the `System.UIntPtr` type is not. The
34 `System.UIntPtr` type is provided mostly to maintain architectural symmetry with the
35 `System.IntPtr` type.

36

37]

38

1 UIntPtr(System.UInt32) Constructor

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor(unsigned int32 value)  
4 [C#]  
5 public UIntPtr(uint value)
```

6 Summary

7 Constructs a new `System.UIntPtr` structure using the specified `System.UInt32`
8 containing a pointer or a handle.

9 Parameters

Parameter	Description
<i>value</i>	A <code>System.UInt32</code> containing a pointer or handle.

10

11

1 UIntPtr(System.UInt64) Constructor

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor(unsigned int64 value)  
4 [C#]  
5 public UIntPtr(ulong value)
```

6 Summary

7 Constructs a new `System.UIntPtr` structure using the specified `System.UInt64`
8 containing a pointer or a handle.

9 Parameters

Parameter	Description
<i>value</i>	A <code>System.UInt64</code> containing a pointer or a handle.

10

11 Exceptions

Exception	Condition
System.OverflowException	The current platform is a 32-bit platform and the value of the current instance is greater than <code>System.UInt32.MaxValue</code> .

12

13

1 UIntPtr.Zero Field

```
2 [ILAsm]  
3 .field public static initOnly valuetype System.UIntPtr Zero  
4 [C#]  
5 public static readonly UIntPtr Zero
```

6 Summary

7 Represents a pointer or handle that has been initialized as zero.

8 Description

9 [Note: The value of this field is not null, but is instead a pointer which has been
10 assigned the value zero. Use this field to efficiently determine whether an instance of
11 System.UIntPtr has been set to a value other than zero. For example, if *uip* is a
12 System.UIntPtr instance, using *uip != UIntPtr.Zero* is more efficient than *uip != new*
13 *UIntPtr(0)* to test if *uip* has been set to a value other than zero.
14
15]

16

1 UIntPtr.Equals(System.Object) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual bool Equals(object obj)  
4 [C#]  
5 public override bool Equals(object obj)
```

6 Summary

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

9 Parameters

Parameter	Description
<i>obj</i>	The <code>System.Object</code> to compare to the current instance.

10

11 Return Value

12 true if *obj* is a `System.UIntPtr` instance and has the same value as the current
13 instance. If *obj* is a null reference or is not an instance of `System.UIntPtr`, returns
14 false.

15 Description

16 [Note: The method overrides `System.Object.Equals`.]
17
18

19

1 IntPtr.GetHashCode() Method

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

6 Summary

7 Generates a hash code for the current instance.

8 Return Value

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

10 Description

11 [*Note:* The algorithm used to generate the hash code is unspecified.]
12
13

14
15 [*Note:* This method overrides `System.Object.GetHashCode.`]
16
17

18

1 UIntPtr.op_Equality(System.UIntPtr, 2 System.UIntPtr) Method

```
3 [ILAsm]  
4 .method public hidebysig static specialname bool op_Equality(valuetype  
5 System.UIntPtr value1, valuetype System.UIntPtr value2)  
  
6 [C#]  
7 public static bool operator ==(UIntPtr value1, UIntPtr value2)
```

8 Summary

9 Determines whether the two specified instances of `System.UIntPtr` represent the same
10 value.

11 Parameters

Parameter	Description
<i>value1</i>	The first <code>System.UIntPtr</code> to compare for equality.
<i>value2</i>	The second <code>System.UIntPtr</code> to compare for equality.

12 13 Return Value

14 true if *value1* represents the same value as *value2*; otherwise, false.

15

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

```
3 [ILAsm]  
4 .method public hidebysig static specialname bool op_Inequality(valuetype  
5 System.UIntPtr value1, valuetype System.UIntPtr value2)  
  
6 [C#]  
7 public static bool operator !=(UIntPtr value1, UIntPtr value2)
```

8 Summary

9 Determines whether two specified instances of `System.UIntPtr` represent different
10 values.

11 Parameters

Parameter	Description
<i>value1</i>	The first <code>System.UIntPtr</code> to compare for inequality.
<i>value2</i>	The second <code>System.UIntPtr</code> to compare for inequality.

12 13 Return Value

14 true if *value1* represents a different value than *value2*; otherwise, false.
15

1 UIntPtr.ToPointer() Method

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Void* ToPointer()  
4 [C#]  
5 unsafe public void* ToPointer()
```

6 Summary

7 Converts the value of the current instance to a pointer to `void`.

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Return Value

11 A pointer to `void`.

12 Description

13 [*Note:* A pointer to `void` points to memory containing data of an unspecified type.]
14
15

16 This method is not CLS-compliant. For a CLS-compliant alternative use
17 `System.IntPtr.ToPointer`.
18

19

1 UIntPtr.ToString() Method

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

6 Summary

7 Returns a `System.String` representation of the value of the current instance.

8 Return Value

9 A `System.String` representation of the current instance.

10 Description

11 [*Note:* If `System.UIntPtr.Size` for the current instance is 4,
12 `System.UIntPtr.ToString` is equivalent to `System.UInt32.ToString()`; otherwise, this
13 method is equivalent to `System.UInt64.ToString()`.

14 This method overrides `System.Object.ToString`.

15]
16]
17]

18

1 UIntPtr.ToInt32() Method

```
2 [ILAsm]  
3 .method public hidebysig instance unsigned int32 ToUInt32()  
4 [C#]  
5 public uint ToUInt32()
```

6 Summary

7 Converts the value of the current instance to a `System.UInt32`.

8 Return Value

9 A `System.UInt32` containing the same value as the current instance.

10 Exceptions

Exception	Condition
System.OverflowException	The current platform is not a 32-bit platform and the value of the current instance is greater than <code>System.UInt32.MaxValue</code> .

11

12

1 UIntPtr.ToInt64() Method

```
2 [ILAsm]  
3 .method public hidebysig instance unsigned int64 ToUInt64()  
4 [C#]  
5 public ulong ToUInt64()
```

6 Summary

7 Converts the value of the current instance to a System.UInt64.

8 Return Value

9 A System.UInt64 containing the same value as the current instance.

10

1 UIntPtr.Size Property

```
2 [ILAsm]  
3 .property int32 Size { public hidebysig static specialname int32  
4 get_Size() }  
5 [C#]  
6 public static int Size { get; }
```

7 Summary

8 Gets the size in bytes of a pointer or a handle for the current implementation.

9 Property Value

10 A `System.Int32` containing the number of bytes of a pointer or handle for the current
11 implementation. The value of this property is equal to the number of bytes contained by
12 the `native unsigned int` type in the current implementation.

13 Description

14 This property is read-only.

15
16 For more information on the `native unsigned int` type, see Partition II of the CLI
17 Specification.

18