

# System.UIntPtr Structure

```
[ILAsm]
.class public sequential sealed serializable UIntPtr extends
System.ValueType

[C#]
public struct UIntPtr
```

## 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)

## Summary

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

## Inherits From: System.ValueType

**Library:** RuntimeInfrastructure

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

## Description

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

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

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

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

1  
2 ]  
3

# UIntPtr(System.UInt32) Constructor

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

## Summary

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

## Parameters

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

# UIntPtr(System.UInt64) Constructor

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

## Summary

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

## Parameters

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

## Exceptions

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

# UIntPtr.Zero Field

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

## Summary

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

## Description

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

]

# UIntPtr.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
<i>obj</i>	The <code>System.Object</code> to compare to the current instance.

## Return Value

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

## Description

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

# UIntPtr.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 the hash code for the current instance.

## Description

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

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

# UIntPtr.op\_Equality(System.UIntPtr, System.UIntPtr) Method

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

## Summary

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

## 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.

## Return Value

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

# UIntPtr.op\_Inequality(System.UIntPtr, System.UIntPtr) Method

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

## Summary

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

## 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.

## Return Value

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

# UIntPtr.ToPointer() Method

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

## Summary

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

## Type Attributes:

- CLSCompliantAttribute(false)

## Return Value

A pointer to void.

## Description

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

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

# UIntPtr.ToString() Method

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

## Summary

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

## Return Value

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

## Description

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

This method overrides `System.Object.ToString`.

]

# UIntPtr.ToInt32() Method

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

## Summary

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

## Return Value

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

## Exceptions

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

## UIntPtr.ToInt64() Method

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

### Summary

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

### Return Value

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

# UIntPtr.Size Property

```
[ILAsm]
.property int32 Size { public hidebysig static specialname int32
get_Size() }

[C#]
public static int Size { get; }
```

## Summary

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

## Property Value

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

## Description

This property is read-only.

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