System.IntPtr Structure

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

[C#]
public struct IntPtr
```

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)

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

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

[*Note:* The `System.IntPtr` type provides CLS-compliant pointer functionality.

`System.IntPtr` 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.

]

IntPtr(System.Int32) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(int32 value)


[C#]
public IntPtr(int value)
```

Summary

Constructs a new System.IntPtr structure using the specified System.Int32 containing a pointer or a handle.

Parameters

Parameter	Description
value	A System.Int32 containing a pointer or a handle.

IntPtr(System.Int64) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(int64 value)


[C#]
public IntPtr(long value)
```

Summary

Constructs a new `System.IntPtr` structure using the specified `System.Int64` containing a pointer or a handle.

Parameters

Parameter	Description
value	A `System.Int64` containing a pointer or a handle.

Exceptions

Exception	Condition
System.OverflowException	The current platform is a 32-bit platform and the value of the current instance is greater than `System.Int32.MaxValue` or less than `System.Int32.MinValue`.

IntPtr.Zero Field

```
[ILAsm]
.field public static initOnly valuetype System.IntPtr Zero

[C#]
public static readonly IntPtr 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.IntPtr` has been set to a value other than zero. For example, if *ip* is a `System.IntPtr` instance, using *ip != IntPtr.Zero* is more efficient than *ip != new IntPtr(0)* to test if *ip* has been set to a value other than zero.

]

IntPtr.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	The `System.Object` to compare to the current instance.

Return Value

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

Description

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

IntPtr.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`.]

IntPtr.op_Equality(System.IntPtr, System.IntPtr) Method

```
[ILAsm]
.method public hidebysig static specialname bool
op_Equality(valuetype System.IntPtr value1, valuetype System.IntPtr
value2)


[C#]
public static bool operator ==(IntPtr value1, IntPtr value2)
```

Summary

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

Parameters

Parameter	Description
value1	The first System.IntPtr to compare for equality.
value2	The second System.IntPtr to compare for equality.

Return Value

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

IntPtr.op_Inequality(System.IntPtr, System.IntPtr) Method

```
[ILAsm]
.method public hidebysig static specialname bool
op_Inequality(valuetype System.IntPtr value1, valuetype
System.IntPtr value2)


[C#]
public static bool operator !=(IntPtr value1, IntPtr value2)
```

Summary

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

Parameters

Parameter	Description
value1	The first `System.IntPtr` to compare for inequality.
value2	The second `System.IntPtr` to compare for inequality.

Return Value

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

IntPtr.ToInt32() Method

```
[ILAsm]
.method public hidebysig instance int32 ToInt32()


[C#]
public int ToInt32()
```

Summary

Converts the value of the current instance to a `System.Int32`.

Return Value

A `System.Int32` containing the same value as the current instance.

Exceptions

Exception	Condition
System.OverflowException	The current platform is not a 32-bit platform and the value of the current instance is greater than `System.Int32.MaxValue` or less than `System.Int32.MinValue`.

IntPtr.ToInt64() Method

```
[ILAsm]
.method public hidebysig instance int64 ToInt64()


[C#]
public long ToInt64()
```

Summary

Converts the value of the current instance to a `System.Int64`.

Return Value

A `System.Int64` containing the same value as the current instance.

IntPtr.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

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.IntPtr.ToInt32.`

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

IntPtr.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.IntPtr.Size` for the current instance is 4,
`System.IntPtr.ToString` is equivalent to `System.Int32.ToString()`; otherwise,
this method is equivalent to `System.Int64.ToString()`.

This method overrides `System.Object.ToString`.

]

IntPtr.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 int` type in the current implementation.

Description

This property is read-only.

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