

1 System.ArgIterator Structure

```
2 [ILAsm]  
3 .class public sealed ArgIterator extends System.ValueType  
4 [C#]  
5 public struct ArgIterator
```

6 Assembly Info:

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

12 Summary

13 Represents an iterator for a variable-length argument list.

14 Inherits From: System.ValueType

15

16 **Library:** Vararg

17

18 **Thread Safety:** All public static members of this type are safe for multithreaded operations.
19 No instance members are guaranteed to be thread safe.

20

21 Description

22 This class is used to enumerate the optional arguments in an argument list.

23

24 [*Note:* The functionality in the `System.ArgIterator` structure is typically hidden in the
25 syntax of a specific programming language. For example, in the C++ programming
26 language you declare a variable-length argument list by specifying an ellipsis ("...") at
27 the end of the argument list.]

28

29

30

1

2 ArgIterator(System.RuntimeArgumentHandle

3) Constructor

```
4 [ILAsm]  
5 public rtspecialname specialname instance void .ctor(valuetype  
6 System.RuntimeArgumentHandle rth)  
7  
8 [C#]  
9 public ArgIterator(RuntimeArgumentHandle rth)
```

9 Summary

10 Constructs and initializes a new instance of System.ArgIterator.

11 Parameters

Parameter	Description
<i>rth</i>	An argument list consisting of mandatory and optional arguments.

12

13 Description

14 The new System.ArgIterator object enumerates the argument list starting from the
15 first optional argument.

16

1 ArgIterator.End() Method

```
2 [ILAsm]  
3 .method public hidebysig void End()  
4 [C#]  
5 public void End()
```

6 Summary

7 Concludes processing of the variable-length argument list represented by this instance.

8 Description

9 [Note: This method corresponds, for example, to the va_end method in the C standard
10 library.]
11
12

13

1 ArgIterator.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 This method is not supported, and always throws `System.NotSupportedException`.

8 Parameters

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

9 Return Value

11 This comparison is not supported. No value is returned.

12 Description

13 [*Note:* This method overrides `System.Object.Equals`.]
14
15

16

1 ArgIterator.GetNextArg() Method

```
2 [ILAsm]  
3 .method public hidebysig TypedReference GetNextArg()  
4 [C#]  
5 public TypedReference GetNextArg()
```

6 Summary

7 Returns the next argument in a variable-length argument list.

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Return Value

11 The next argument as a *System.TypedReference* object.

12 Description

13 The iterator is automatically advanced to the next argument.

14 Exceptions

Exception	Condition
System.InvalidOperationException	An attempt was made to read beyond the end of the list.

15

16

1 ArgIterator.GetNextArgType() Method

```
2 [ILAsm]  
3 .method public hidebysig RuntimeTypeHandle GetNextArgType()  
4 [C#]  
5 public RuntimeTypeHandle GetNextArgType()
```

6 Summary

7 Returns the type of the next argument.

8 Return Value

9 The next argument as a `System.RuntimeTypeHandle` object.

10 Description

11 This method does not advance the iterator to the next argument.

12

13 Returns the type of the argument as specified in the call. For example, if an argument
14 has type `System.String`, but the call specifies the argument as type `System.Object`,
15 then this function returns `System.Object`.

16

1 ArgIterator.GetRemainingCount() Method

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

6 Summary

7 Returns the number of arguments remaining in the argument list.

8 Return Value

9 The number of remaining arguments as a `System.Int32`.

10