

1 System.Xml.NameTable Class

```
2 [ILAsm]  
3 .class public NameTable extends System.Xml.XmlNameTable  
4 [C#]  
5 public class NameTable: XmlNameTable
```

6 Assembly Info:

- 7 • *Name:* System.Xml
- 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 Creates a table that stores unique instances of `System.String` objects.

14 Inherits From: System.Xml.XmlNameTable

15

16 **Library:** XML

17

18 **Thread Safety:** This class is multi-read threadsafe but not threadsafe for read/write.

19

20 Description

21 Only a single instance of any given string is stored even if the string is added multiple
22 times to the table.

23

24 Using this class provides an efficient means for an XML parser to use the same
25 `System.String` object for all repeated element and attribute names in an XML
26 document. If the same object is used for all repeated names, the efficiency of name
27 comparisons is increased by allowing the names to be compared using object
28 comparisons rather than string comparisons.

29

30 [*Note:* This class implements a single-threaded `System.Xml.XmlNameTable`.

31

32 This class is used internally by the `System.Xml.XmlNamespaceManager`,
33 `System.Xml.XmlParserContext`, and `System.Xml.XmlTextReader` classes to store
34 element and attribute names.

35

36]

37 Example

38 The following example demonstrates the difference between equal string values and
39 equal `System.String` objects using the `System.Xml.NameTable` class.

```

1
2     [C#]

3 using System;
4 using System.Text;
5 using System.Xml;
6
7 class Ntable {
8
9     public static void Main() {
10
11         NameTable nameTable = new NameTable();
12
13         string str1 = "sunny";
14         StringBuilder strBuilder = new StringBuilder();
15         string str2 =
16             strBuilder.Append("sun").Append("ny").ToString();
17         Console.WriteLine( "{0}: {1}",
18                             str1, str2 );
19         Console.WriteLine( "{0}: {1}",
20                             str1 == str2,
21                             (Object)str1==(Object)str2 );
22
23         string str3 = nameTable.Add(str1);
24         string str4 = nameTable.Add(str2);
25         Console.WriteLine( "{0}: {1}",
26                             str3, str4 );
27         Console.WriteLine( "{0}: {1}",
28                             str3 == str4,
29                             (Object)str3==(Object)str4 );
30     }
31 }

```

33 The output is

34
35 sunny: sunny

36
37 True: False

38
39 sunny: sunny

40
41 True: True

42

1 NameTable() Constructor

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor()  
4 [C#]  
5 public NameTable()
```

6 Summary

7 Constructs and initializes a new instance of the `System.Xml.NameTable` class.

8

1 NameTable.Add(System.String) Method

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

6 Summary

7 Adds the specified System.String to the table if a System.String instance with the
8 same value does not already exist in the table.

9 Parameters

Parameter	Description
<i>key</i>	The System.String to add.

10

11 Return Value

12 *key*, if it did not exist in the table at the time of the call, or the System.String instance
13 previously stored in the table with a value equal to *key*.

14 Description

15 Only a single instance of any given System.String is stored in the table. If the value of
16 *key* is already stored in the table, the System.String instance with that value is
17 returned.

18
19 [Note: This method overrides System.Xml.XmlNameTable.Add(String).
20
21]

22 Exceptions

Exception	Condition
System.ArgumentNullException	<i>key</i> is null.

23

24

1 NameTable.Add(System.Char[], 2 System.Int32, System.Int32) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual string Add(char[] key, int32 start, int32  
5 len)  
6 [C#]  
7 public override string Add(char[] key, int start, int len)
```

8 Summary

9 Adds the System.String equivalent of a specified subset of a System.Char array to the
10 table if the string equivalent does not already exist in the table.

11 Parameters

Parameter	Description
<i>key</i>	A System.Char array containing the string to add.
<i>start</i>	A System.Int32 specifying the zero-based index into the array of the first character of the string.
<i>len</i>	A System.Int32 containing the number of characters in the string.

12 13 Return Value

14 The System.String equivalent of the specified subset of the System.Char array that is
15 stored in the table, or System.String.Empty if *len* is zero.

16 Description

17 Only a single instance of any given System.String is stored in the table. Calling this
18 method with the same subset (containing the same characters) of any System.Char
19 array, returns the same instance of the System.String equivalent.
20

21 [Note: This method overrides System.Xml.XmlNameTable.Add(Char[], Int32, Int32).
22
23]

24 Exceptions

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

<p>System.IndexOutOfRangeException</p>	<p>$start < 0$.</p> <p>- or -</p> <p>$start \geq key.Length$.</p> <p>- or -</p> <p>$len > key.Length - start$.</p> <p>The above conditions do not cause an exception to be thrown if $len = 0$.</p>
<p>System.ArgumentOutOfRangeException</p>	<p>$len < 0$.</p>

1

2

1 NameTable.Get(System.String) Method

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

6 Summary

7 Looks up the value of the specified System.String in the table.

8 Parameters

Parameter	Description
<i>value</i>	The System.String to look up.

9

10 Return Value

11 The System.String instance previously stored in the table with a value equal to *value*,
12 or null if it does not exist.

13 Description

14 Only a single instance of any given System.String is stored in the table. If the value of
15 *value* is already stored in the table, the System.String instance with that value is
16 returned.

17
18 [*Note:* This method overrides System.Xml.XmlNameTable.Get(String).
19
20]

21 Exceptions

Exception	Condition
System.ArgumentNullException	<i>value</i> is null.

22

23

1 NameTable.Get(System.Char[], 2 System.Int32, System.Int32) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual string Get(char[] key, int32 start, int32  
5 len)  
6 [C#]  
7 public override string Get(char[] key, int start, int len)
```

8 Summary

9 Looks up the System.String equivalent to a specified subset of a System.Char array in
10 the table.

11 Parameters

Parameter	Description
<i>key</i>	A System.Char array containing the string to look up.
<i>start</i>	A System.Int32 specifying the zero-based index into the array of the first character of the string.
<i>len</i>	A System.Int32 containing the number of characters in the string.

12 13 Return Value

14 The System.String equivalent of the specified subset of the System.Char array that is
15 stored in the table, or null if the equivalent System.String is not in the table.

16 Description

17 Only a single instance of any given System.String is stored in the table. Calling this
18 method with the same subset (containing the same characters) of any System.Char
19 array, returns the same instance of the System.String equivalent, if it exists.

20
21 [Note: This method overrides System.Xml.XmlNameTable.Get(Char[], Int32, Int32).

22]
23

24 Exceptions

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

<p>System.IndexOutOfRangeException</p>	<p><i>start</i></p> <p>< 0.</p> <p>- or -</p> <p><i>start</i> >= <i>key.Length</i>.</p> <p>- or -</p> <p><i>len</i> > <i>key.Length</i> - <i>start</i>.</p> <p>The above conditions do not cause an exception to be thrown if <i>len</i> = 0.</p>
<p>System.ArgumentOutOfRangeException</p>	<p><i>len</i> < 0.</p>

1
2