

System.Collections.Specialized.NameValueCollection Class

```
[ILASM]
.class public serializable NameValueCollection extends
System.Object

[C#]
public class NameValueCollection: Object
```

Assembly Info:

- *Name:* System
- *Public Key:* [00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
 - CLSCompliantAttribute(true)

Type Attributes:

- DefaultMemberAttribute("Item") [*Note:* This attribute requires the RuntimeInfrastructure library.]

Implements:

- **System.Collections.ICollection**
- **System.Collections.IEnumerable**

Summary

Represents a collection of associated **System.String** keys and **System.String** values.

Inherits From: System.Object

Library: Networking

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

Description

This class can be used for headers, query strings and form data. Each key in the collection is associated with one or more values. Multiple values for a particular key are contained in a single **System.String** are stored as a **System.Collections.ArrayList**.

1 The capacity is the number of key-and-value pairs that the
2 **System.Collections.Specialized.NameValueCollection** can
3 contain. The default initial capacity is zero. The capacity is
4 automatically increased as required.
5
6 The hash code provider dispenses hash codes for keys in the
7 **System.Collections.Specialized.NameValueCollection**.
8
9 The comparer determines whether two keys are equal.
10

1 NameValueCollection() Constructor

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

6 Summary

7 Constructs and initializes a new instance of the
8 **System.Collections.Specialized.NameValueCollection** class.

9 Description

10 The new instance is initialized with the default initial capacity,
11 **System.Collections.IHashCodeProvider**, and
12 **System.Collections.IComparer**.

13

1 NameValueCollection(System.Collections. 2 Specialized.NameValueCollection) 3 Constructor

```
4 [ILASM]  
5 public rtspecialname specialname instance void .ctor(class  
6 System.Collections.Specialized.NameValueCollection col)  
  
7 [C#]  
8 public NameValueCollection(NameValueCollection col)
```

9 Summary

10 Constructs and initializes a new instance of the
11 **System.Collections.Specialized.NameValueCollection** class using
12 the values of the specified
13 **System.Collections.Specialized.NameValueCollection**.

14 Parameters

15
16

Parameter	Description
<i>col</i>	The System.Collections.Specialized.NameValueCollection used to initialize the new instance.

17
18

18 Description

19 The capacity, values, and order of values of the new instance are equal
20 to the capacity and values of *col*. The
21 **System.Collections.IHashCodeProvider** and
22 **System.Collections.IComparer** of the new instance are the default
23 instances.

24
25
26
27
28

The elements of the new
System.Collections.Specialized.NameValueCollection are sorted
in the same order as the source
System.Collections.Specialized.NameValueCollection.

29 Exceptions

30
31

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

32
33
34

1 **NameValueCollection(System.Collections.**
2 **IHashCodeProvider,**
3 **System.Collections.IComparer)**
4 **Constructor**

```
5 [ILASM]  
6 public rtspecialname specialname instance void .ctor(class  
7 System.Collections.IHashCodeProvider hashProvider, class  
8 System.Collections.IComparer comparer)
```

```
9 [C#]  
10 public NameValueCollection(IHashCodeProvider hashProvider,  
11 IComparer comparer)
```

12 **Summary**

13 Constructs and initializes a new instance of the
14 **System.Collections.Specialized.NameValueCollection** class with
15 the specified **System.Collections.IHashCodeProvider** and the
16 specified **System.Collections.IComparer**.

17 **Parameters**

Parameter	Description
<i>hashProvider</i>	The System.Collections.IHashCodeProvider that supplies the hash codes for all keys in the new instance; or, null to use the default hash code provider.
<i>comparer</i>	The System.Collections.IComparer to use to determine whether two keys are equal. Specify null to use the default comparer.

20
21 **Description**

22 The new instance is initialized with the default capacity of zero.
23

1 NameValueCollection(System.Int32)

2 Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(int32  
5 capacity)  
  
6 [C#]  
7 public NameValueCollection(int capacity)
```

8 Summary

9 Constructs and initializes a new instance of the
10 **System.Collections.Specialized.NameValueCollection** class with
11 the specified initial capacity.

12 Parameters

13
14

Parameter	Description
<i>capacity</i>	A System.Int32 containing the initial number of entries that the new instance can contain.

15
16

16 Description

17 The new instance is initialized with the default
18 **System.Collections.IHashCodeProvider** and
19 **System.Collections.IComparer**.

20 Exceptions

21
22

Exception	Condition
System.ArgumentOutOfRangeException	<i>capacity</i> < 0.

23
24
25

1 NameValueCollection(System.Int32, 2 System.Collections.Specialized.NameValueCollection) Constructor 3

```
4 [ILASM]  
5 public rtspecialname specialname instance void .ctor(int32  
6 capacity, class  
7 System.Collections.Specialized.NameValueCollection col)  
8  
9 [C#]  
10 public NameValueCollection(int capacity,  
NameValueCollection col)
```

11 Summary

12 Constructs and initializes new instance of the
13 **System.Collections.Specialized.NameValueCollection** class that
14 contains the same values as the specified
15 **System.Collections.Specialized.NameValueCollection** and either
16 the specified capacity or the capacity of the specified collection,
17 whichever is greater.

18 Parameters

19
20

Parameter	Description
<i>capacity</i>	A System.Int32 containing the initial number of entries that the new instance can contain.
<i>col</i>	The System.Collections.Specialized.NameValueCollection used to initialize the new instance.

21

22 Description

23 The new instance is initialized with the default
24 **System.Collections.IHashCodeProvider** and
25 **System.Collections.IComparer**.

26 Exceptions

27
28

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

29
30
31

1 **NameValueCollection(System.Int32,**
2 **System.Collections.IHashCodeProvider,**
3 **System.Collections.IComparer)**
4 **Constructor**

```
5 [ILASM]  
6 public rtspecialname specialname instance void .ctor(int32  
7 capacity, class System.Collections.IHashCodeProvider  
8 hashProvider, class System.Collections.IComparer comparer)  
9  
10 [C#]  
11 public NameValueCollection(int capacity, IHashCodeProvider  
hashProvider, IComparer comparer)
```

12 **Summary**

13 Constructs and initializes new instance of the
14 **System.Collections.Specialized.NameValueCollection** class with
15 the specified initial capacity, hash code provider, and comparer.

16 **Parameters**

17
18

Parameter	Description
<i>capacity</i>	A System.Int32 containing the initial number of entries that the System.Collections.Specialized.NameValueCollection can contain.
<i>hashProvider</i>	The System.Collections.IHashCodeProvider that will supply the hash codes for all keys in the new instance.
<i>comparer</i>	The System.Collections.IComparer to use to determine whether two keys in the new instance are equal.

19
20
21
22

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	<i>capacity</i> < 0.

23
24
25

1 NameValueCollection.Add(System.String, 2 System.String) Method

```
3 [ILASM]  
4 .method public hidebysig virtual void Add(string name,  
5 string value)  
6 [C#]  
7 public virtual void Add(string name, string value)
```

8 Summary

9 Adds an entry with the specified key and value to the current instance.

10 Parameters

11
12

Parameter	Description
<i>name</i>	A System.String that represents the key of the entry to add. Can be null .
<i>value</i>	A System.String that represents the value of the entry to add. Can be null .

13
14

15 Behaviors

16 As described above.

17 Default

18 If the specified key already exists in the current instance, the specified
19 value is added to the existing comma-separated list of values
20 associated with the same key.

21
22 If the same value already exists under the same key in the collection,
23 the new value overwrites the old value.

24 How and When to Override

25 Override this method to customize the default behavior in a type
26 derived from the current type.

27 Usage

1 Use this method to add an entry to the current instance.

2 **Exceptions**

3

4

Exception	Condition
System.NotSupportedException	The current instance is read-only.

5

6

7

1 NameValueCollection.Add(System.Collections.Specialized.NameValueCollection) 2 3 Method

```
4 [ILASM]  
5 .method public hidebysig instance void Add(class  
6 System.Collections.Specialized.NameValueCollection c )  
7  
8 [C#]  
9 public void Add(NameValueCollection c)
```

9 Summary

10 Copies the entries from the specified
11 **System.Collections.Specialized.NameValueCollection** to the
12 current instance.

13 Parameters

Parameter	Description
<i>c</i>	The System.Collections.Specialized.NameValueCollection to copy to the current instance.

16 17 Description

18 If a key in *c* already exists in the target
19 **System.Collections.Specialized.NameValueCollection** instance,
20 the associated value in *c* is added to the existing comma-separated list
21 of values associated with the same key in the target
22 **System.Collections.Specialized.NameValueCollection** instance.

23 Exceptions

Exception	Condition
System.NotSupportedException	The current instance is read-only.
System.ArgumentNullException	<i>c</i> is null .

26
27
28

1 NameValueCollection.Clear() Method

```
2 [ILASM]  
3 .method public hidebysig instance void Clear()  
4 [C#]  
5 public void Clear()
```

6 Summary

7 Invalidates the cached arrays and removes all entries from the current
8 instance.

9 Description

10 The value of each key and value in the current instance is set to **null**.
11
12 If the current instance is empty, it remains unchanged and no
13 exception is thrown.

14 Exceptions

15
16

Exception	Condition
System.NotSupportedException	The current instance is read-only.

17
18
19

1 NameValueCollection.CopyTo(System.Array, System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig instance void CopyTo(class  
5 System.Array dest, int32 index)  
  
6 [C#]  
7 public void CopyTo(Array dest, int index)
```

8 Summary

9 Copies the elements from the current instance to the specified
10 **System.Array**, starting at the specified index in that array.

11 Parameters

12
13

Parameter	Description
<i>dest</i>	A one-dimensional, zero-based System.Array that is the destination of the elements copied from the current instance.
<i>index</i>	A System.Int32 containing the zero-based index in <i>dest</i> at which copying begins.

14
15

15 Description

16 This method uses **System.Array.Copy** to copy the elements.

17
18
19

[Note: This method is implemented to support the **System.Collections.ICollection** interface.]

20 Exceptions

21
22

Exception	Condition
System.ArgumentNullException	<i>dest</i> is null .
System.ArgumentOutOfRangeException	<i>index</i> < 0. <i>dest</i> has more than one dimension. <i>index</i> >= <i>dest.Length</i> .
System.ArgumentException	The number of elements in the current instance is greater than the available space from <i>index</i> to the end of the destination <i>dest</i> .

1
2
3

System.InvalidCastException

At least one element in the current instance is not assignment-compatible with the type of *dest*.

1 NameValueCollection.Get(System.Int32)

2 Method

```
3 [ILASM]  
4 .method public hidebysig virtual string Get(int32 index)  
5 [C#]  
6 public virtual string Get(int index)
```

7 Summary

8 Returns the values at the specified index of the current instance.

9 Parameters

10
11

Parameter	Description
<i>index</i>	A System.Int32 that specifies the zero-based index of the entry that contains the values to get from the current instance.

12
13
14

Return Value

15 A **System.String** that contains a comma-separated list of the values
16 at the specified index of the current instance, if found; otherwise, **null**.

17 Behaviors

18 As described above.

19 Exceptions

20
21

Exception	Condition
System.ArgumentOutOfRangeException	<i>index</i> is outside the valid range of indices for the current instance.

22
23
24

1 NameValueCollection.Get(System.String)

2 Method

```
3 [ILASM]  
4 .method public hidebysig virtual string Get(string name)  
5 [C#]  
6 public virtual string Get(string name)
```

7 Summary

8 Gets the values associated with the specified key from the current
9 instance combined into one comma-separated list.

10 Parameters

11
12

Parameter	Description
<i>name</i>	A System.String that specified the key of the entry that contains the values to get.

13
14
15

Return Value

16 A **System.String** that contains a comma-separated list of the values
17 associated with the specified key from the current instance, if found;
18 otherwise, **null**.

19 Behaviors

20 As described above.

21 Default

22 If *name* is **null**, no exception is thrown and **null** is returned.

23

1 NameValueCollection.GetKey(System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig virtual string GetKey(int32 index)  
5  
6 [C#]  
7 public virtual string GetKey(int index)
```

7 Summary

8 Returns the key at the specified index of the current instance.

9 Parameters

10
11

Parameter	Description
<i>index</i>	A System.Int32 that specifies the zero-based index of the key to get from the current instance.

12
13
14

13 Return Value

15 A **System.String** that contains the key at the specified index of the
16 current instance, if found; otherwise, **null**.

17 Behaviors

18 As described above.

19 Exceptions

20
21

Exception	Condition
System.ArgumentOutOfRangeException	<i>index</i> is outside the valid range of indices for the current instance.

22
23
24

1 NameValueCollection.GetValues(System.Int32) Method

```
3 [ILASM]
4 .method public hidebysig virtual class System.String[]
5 GetValues(int32 index)
6
7 [C#]
8 public virtual string[] GetValues(int index)
```

8 Summary

9 Returns an array that contains the values at the specified index of the
10 current instance.

11 Parameters

12
13

Parameter	Description
<i>index</i>	A System.Int32 that specifies the zero-based index of the entry that contains the values to get from the current instance.

14
15
16

15 Return Value

17 A **System.String** array containing the values at the specified index of
18 the current instance, if found; otherwise, **null**.

19 Behaviors

20 As described above.

21 Exceptions

22
23

Exception	Condition
System.ArgumentOutOfRangeException	<i>index</i> is outside the valid range of indices for the current instance.

24
25
26

1 NameValueCollection.GetValues(System.String) Method

```
3 [ILASM]  
4 .method public hidebysig virtual class System.String[]  
5 GetValues(string name)  
  
6 [C#]  
7 public virtual string[] GetValues(string name)
```

8 Summary

9 Gets the values associated with the specified key from the current
10 instance.

11 Parameters

12
13

Parameter	Description
<i>name</i>	A System.String that specifies the key of the entry that contains the values to get.

14
15
16

Return Value

17 A **System.String** array containing the values associated with *name*
18 from the current instance, if found; otherwise, **null**.

19 Behaviors

20 As described above.

21 Default

22 If *name* is **null**, no exception is thrown and **null** is returned.

23

1 NameValueCollection.HasKeys() Method

```
2 [ILASM]  
3 .method public hidebysig instance bool HasKeys()  
4 [C#]  
5 public bool HasKeys()
```

6 Summary

7 Gets a **System.Boolean** value indicating whether the current instance
8 contains keys that are not **null**.

9 Return Value 10

11 **true** if the current instance contains keys that are not **null**; otherwise,
12 **false**.

13

1 NameValueCollection.InvalidateCachedArrays() Method

```
3 [ILASM]
4 .method family hidebysig instance void
5 InvalidateCachedArrays()
6
7 [C#]
8 protected void InvalidateCachedArrays()
```

8 Summary

9 Resets the cached arrays of the current instance to **null**.

10 Description

11 [Note: The array returned by
12 **System.Collections.Specialized.NameValueCollection.AllKeys** is
13 cached for better performance and is automatically refreshed when the
14 collection changes. A derived class can invalidate the cached version
15 by calling
16 **System.Collections.Specialized.NameValueCollection.Invalidate**
17 **CachedArrays**, thereby forcing the arrays to be recreated.]

18

1 NameValueCollection.Remove(System.String) Method

```
3 [ILASM]  
4 .method public hidebysig virtual void Remove(string name)  
5 [C#]  
6 public virtual void Remove(string name)
```

7 Summary

8 Removes the entry with the specified key from the current instance.

9 Parameters

10
11

Parameter	Description
<i>name</i>	A System.String containing the key of the entry to remove from the current instance.

12
13

14 Behaviors

15 If *name* is found, the key *name* and its associated value are set to
16 **null**. Removing an element does not alter the capacity of a
17 **System.Collections.Specialized.NameValueCollection**.

18 Default

19 This method uses the **System.Object.Equals** implementation of
20 *name* to locate *name* in the current instance. If *name* is not found in
21 the current instance or is **null**, no exception is thrown and the current
22 instance is unchanged.

23 Exceptions

24
25

Exception	Condition
System.NotSupportedException	The current instance is read-only or has a fixed size.

26
27
28

1 NameValueCollection.Set(System.String, 2 System.String) Method

```
3 [ILASM]  
4 .method public hidebysig virtual void Set(string name,  
5 string value)  
6  
7 [C#]  
8 public virtual void Set(string name, string value)
```

8 Summary

9 Sets the value of the specified entry in the current instance to the
10 specified value.

11 Parameters

12
13

Parameter	Description
<i>name</i>	A System.String containing the key of the entry to add the new value to.
<i>value</i>	A System.String containing the new value to add to the specified entry.

14
15

16 Behaviors

17 If the specified key already exists in the current instance, this method
18 overwrites the existing values with the specified value. If the specified
19 key does not exist in the current instance, this method creates a new
20 entry using the specified key and the specified value.

21 Usage

22 Use the
23 **System.Collections.Specialized.NameValueCollection.Add**
24 method to add the new value to the existing list of values.

25 Exceptions

26
27

Exception	Condition
System.NotSupportedException	The current instance is read-only.

1
2
3

1 NameValueCollection.AllKeys Property

```
2 [ILASM]
3 .property class System.String[] AllKeys { public hidebySig
4 virtual specialname class System.String[] get_AllKeys() }
5 [C#]
6 public virtual string[] AllKeys { get; }
```

7 Summary

8 Gets all the keys in the current instance.

9 Property Value

10

11 A **System.String** array containing all the keys of the current instance.
12 If the current instance is empty, the value of this property is an empty
13 array.

14 Behaviors

15 This property is read-only.

16 Usage

17 The array returned by
18 **System.Collections.Specialized.NameValueCollection.AllKeys** is
19 cached for better performance and is automatically refreshed when the
20 collection changes. A derived class can invalidate the cached version
21 by calling
22 **System.Collections.Specialized.NameValueCollection.Invalidate**
23 **CachedArrays**, thereby forcing the array to be refreshed.

24

NameValueCollection.Item Property

```
[ILASM]
.property string Item[string name] { public hidebysig
specialname instance string get_Item(string name) public
hidebysig specialname instance void set_Item(string name,
string value) }

[C#]
public string this[string name] { get; set; }
```

Summary

Gets or sets the value in the current instance that is associated with the specified key.

Parameters

Parameter	Description
<i>name</i>	A System.String containing the key of the entry to locate.

Property Value

A **System.String** that contains the comma-separated list of values associated with the specified key. If *key* is not contained in the current instance, attempting to get it returns **null**, and attempting to set it creates a new entry using *key*.

Description

If the specified key already exists in the collection, setting this property overwrites the existing values with the specified value. If the specified key does not exist in the collection, setting this property creates a new entry using the specified key and the specified value.

[*Note:* This property provides the ability to access a specific element in the current instance using the following notation: `myCollection[key]`.

To add the new value to the existing list of values, use the **System.Collections.Specialized.NameValueCollection.Add** method.]

Exceptions

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

System.ArgumentOutOfRangeException	<i>index</i> is outside the valid range of indexes for the collection.
System.NotSupportedException	The property is being set and the current instance is read-only. The property is being set, <i>key</i> is not contained in the current instance, and the current instance has a fixed size.

1
2
3

1 NameValueCollection.Item Property

```
2 [ILASM]
3 .property string Item[int32 index] { public hideby sig
4 specialname instance string get_Item(int32 index) }
5
6 [C#]
7 public string this[int index] { get; }
```

7 Summary

8 Gets the value in the current instance that is associated with the
9 specified index.

10 Parameters

11
12

Parameter	Description
<i>index</i>	A System.Int32 that specifies the zero-based index of the entry to locate in the current instance.

13
14
15

14 Property Value

16 A **System.String** that contains the comma-separated list of values at
17 the specified index of the current instance.

18 Description

19 This property is read-only.

20
21
22
23

[*Note*: This property provides the ability to access a specific element in the collection by using the following syntax: `myCollection[index]`.]

24 This property cannot be set. To set the value at a specified index, use
25 `Item[GetKey(index)].`

26 Exceptions

27
28

Exception	Condition
System.ArgumentOutOfRangeException	<i>index</i> is outside the valid range of indices for the current instance.

29
30