

1 System.Collections.Specialized.NameValueCollection Class

```
3 [ILAsm]  
4 .class public serializable NameValueCollection extends System.Object  
5 implements System.Collections.ICollection, System.Collections.IEnumerable  
6 [C#]  
7 public class NameValueCollection: ICollection, IEnumerable
```

8 Assembly Info:

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

14 Type Attributes:

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

17 Implements:

- 18 • **System.Collections.ICollection**
- 19 • **System.Collections.IEnumerable**

20 Summary

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

22 Inherits From: System.Object

23

24 **Library:** Networking

25

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

28

29 Description

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

33

34 The capacity is the number of key-and-value pairs that the
35 System.Collections.Specialized.NameValueCollection can contain. The default
36 initial capacity is zero. The capacity is automatically increased as required.

- 1
- 2 The hash code provider dispenses hash codes for keys in the
- 3 `System.Collections.Specialized.NameValueCollection`.
- 4
- 5 The comparer determines whether two keys are equal.

6

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

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

11 Summary

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

15 Parameters

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.

16 17 Exceptions

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

18
19

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

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

9 Summary

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

15 Parameters

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.

16 17 Description

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

20 Exceptions

Exception	Condition
System.ArgumentNullException	<i>col</i> is null.
System.ArgumentOutOfRangeException	<i>capacity</i> is < 0.

21

22

1 NameValueCollection(System.Int32)

2 Constructor

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

7 Summary

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

11 Parameters

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

12 13 Description

14 The new instance is initialized with the default
15 System.Collections.IHashCodeProvider and System.Collections.IComparer.

16 Exceptions

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

17
18

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

NameValueCollection(System.Collections.IHashCodeProvider, System.Collections.IComparer) Constructor

```
[ILAsm]  
public rtspecialname specialname instance void .ctor(class  
System.Collections.IHashCodeProvider hashProvider, class  
System.Collections.IComparer comparer)  
  
[C#]  
public NameValueCollection(IHashCodeProvider hashProvider, IComparer  
comparer)
```

Summary

Constructs and initializes a new instance of the System.Collections.Specialized.NameValueCollection class with the specified System.Collections.IHashCodeProvider and the specified System.Collections.IComparer.

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.

Description

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

NameValueCollection(System.Collections.Specialized.NameValueCollection) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(class
System.Collections.Specialized.NameValueCollection col)

[C#]
public NameValueCollection(NameValueCollection col)
```

Summary

Constructs and initializes a new instance of the System.Collections.Specialized.NameValueCollection class using the values of the specified System.Collections.Specialized.NameValueCollection.

Parameters

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

Description

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

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

Exceptions

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

24

25

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 System.Collections.IComparer.

12

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

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

7 Summary

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

9 Parameters

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.

10

11 Behaviors

12 As described above.

13

14 Default

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

17

18 Attempting to assign the same value to an existing key adds a new value to that key,
19 thus providing two (or more) copies of the same value associated with the key.

20 How and When to Override

21 Override this method to customize the default behavior in a type derived from the
22 current type.

23

24 Usage

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

1

2 **Exceptions**

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

3

4

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

```
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 current instance.

12 **Parameters**

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

13
14 **Description**

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

20 **Exceptions**

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

21
22

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 instance.

8 Description

9 The value of each key and value in the current instance is set to null.

10

11 If the current instance is empty, it remains unchanged and no exception is thrown.

12 Exceptions

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

13

14

1 NameValueCollection.CopyTo(System.Array, 2 System.Int32) Method

```
3 [ILAsm]  
4 .method public hidebysig instance void CopyTo(class System.Array dest,  
5 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 `System.Array`, starting
10 at the specified index in that array.

11 Parameters

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

12

13 Description

14 This method uses `System.Array.Copy` to copy the elements.

15

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

18

19

20 Exceptions

Exception	Condition
System.ArgumentNullException	<i>dest</i> is null.
System.ArgumentOutOfRangeException	<i>index</i> < 0.
System.ArgumentException	<i>dest</i> has more than one dimension. -or-

	<p><i>index</i> >= <i>dest.Length</i>.</p> <p>-or-</p> <p>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>.</p>
System.InvalidCastException	<p>At least one element in the current instance is not assignment-compatible with the type of <i>dest</i>.</p>

1

2

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 instance combined
9 into one comma-separated list.

10 Parameters

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

11

12 Return Value

13 A `System.String` that contains a comma-separated list of the values associated with the
14 specified key from the current instance, if found; otherwise, `null`.

15 Behaviors

16 As described above.

17

18 Default

19 If *name* is `null`, the values associated with the null key, if any, are returned; otherwise,
20 `null` is returned.

21

22

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

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

10

11 Return Value

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

14 Behaviors

15 As described above.

16

17 Exceptions

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

18

19

1 NameValueCollection.GetEnumerator() 2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual class System.Collections.IEnumerator  
5 GetEnumerator()  
  
6 [C#]  
7 public virtual IEnumerator GetEnumerator()
```

8 Summary

9 Returns a System.Collections.IEnumerator for the current instance.

10 Return Value

11 A System.Collections.IEnumerator for the current instance.

12 Description

13 If the current instance is modified while an enumeration is in progress, a call to
14 System.Collections.IEnumerator.MoveNext or
15 System.Collections.IEnumerator.Reset throws
16 System.InvalidOperationException.
17

18 [*Note:* For detailed information regarding the use of an enumerator, see
19 System.Collections.IEnumerator. This property is implemented to support the
20 System.Collections.IEnumerable interface.]
21
22
23

1 NameValueCollection.GetKey(System.Int32)

2 Method

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

7 Summary

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

9 Parameters

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

10 11 Return Value

12 A `System.String` that contains the key at the specified index of the current instance, if
13 found; otherwise, `null`.

14 Behaviors

15 As described above.

16 17 Exceptions

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

18
19

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

NameValueCollection.GetValues(System.String) Method

```
[ILAsm]  
.method public hidebysig virtual string[] GetValues(string name)  
  
[C#]  
public virtual string[] GetValues(string name)
```

Summary

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

Parameters

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

Return Value

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

Behaviors

As described above.

Default

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

1
2 **NameValueCollection.GetValues(System.Int32)**
3 **Method**

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

8 **Summary**

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

10 **Parameters**

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

11
12 **Return Value**

13 A `System.String` array containing the values at the specified index of the current
14 instance, if found; otherwise, `null`.

15 **Behaviors**

16 As described above.

17

18 **Exceptions**

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

19

20

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 contains keys that
8 are not null.

9 Return Value

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

11

1
2 **NameValueCollection.InvalidateCachedArrays**
3 **() Method**

```
4 [ILAsm]  
5 .method family hidebysig instance void InvalidateCachedArrays()  
6 [C#]  
7 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 cached for better
13 performance and is automatically refreshed when the collection changes. A derived class
14 can invalidate the cached version by calling
15 System.Collections.Specialized.NameValueCollection.InvalidateCachedArrays,
16 thereby forcing the arrays to be recreated.]
17
18

19

1 NameValueCollection.Remove(System.String)

2 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

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

10

11 Behaviors

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

15

16 Default

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

20

21 Exceptions

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

22

23

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

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

7 Summary

8 Sets the value of the specified entry in the current instance to the specified value.

9 Parameters

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

10

11 Behaviors

12 If the specified key already exists in the current instance, this method overwrites the
13 existing values with the specified value. (If the existing value contains a string of
14 multiple comma-delimited values, the complete string is replaced with a single instance
15 of value.) If the specified key does not exist in the current instance, this method creates
16 a new entry using the specified key and the specified value.

17

18 Usage

19 Use the `System.Collections.Specialized.NameValueCollection.Add` method to add
20 the new value to the existing list of values.

21

22 Exceptions

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

1

2

1 NameValueCollection.AllKeys Property

```
2 [ILAsm]  
3 .property string[] AllKeys { public hidebysig virtual specialname string[]  
4 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 A `System.String` array containing all the keys of the current instance. If the current
11 instance is empty, the value of this property is an empty array.

12 Behaviors

13 This property is read-only.

14

15 Usage

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

22

23

1 NameValueCollection.Count Property

```
2 [ILAsm]
3 .property int32 Count { public hidebysig virtual specialname int32
4 get_Count() }
5 [C#]
6 public virtual int Count { get; }
```

7 Summary

8 Gets the number of elements contained in the current instance.

9 Property Value

10 A `System.Int32` that indicates the number of elements contained in the current
11 instance.

12 Description

13 This property is read-only.

14

15 [*Note:* This property is implemented to support the `System.Collections.ICollection`
16 interface.

17

18]

19

1 NameValueCollection.IsReadOnly Property

```
2 [ILAsm]  
3 .property bool IsReadOnly { family hidebysig specialname bool  
4 get_IsReadOnly() family hidebysig specialname void set_IsReadOnly(bool  
5 value) }  
6 [C#]  
7 protected bool IsReadOnly { get; set; }
```

8 Summary

9 Gets or sets a value indicating whether the current instance is read-only.

10 Property Value

11 true if the current instance is read-only; otherwise, false.

12 Description

13 This property is read-write.

14

1 NameValueCollection.IsSynchronized

2 Property

```
3 [ILAsm]  
4 .property bool ICollection.IsSynchronized { public hidebysig virtual  
5 abstract specialname bool get_ICollection.IsSynchronized() }  
  
6 [C#]  
7 bool ICollection.IsSynchronized { get; }
```

8 Summary

9 Implemented to support System.Collections.ICollection.

10

11 [*Note:* For more information, see System.Collections.ICollection.IsSynchronized.]

12

13

14

1 NameValueCollection.Item Property

```
2 [ILAsm]
3 .property string Item[string name] { public hidebysig specialname instance
4 string get_Item(string name) public hidebysig specialname instance void
5 set_Item(string name, string value) }
6
7 [C#]
8 public string this[string name] { get; set; }
```

8 Summary

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

10 Parameters

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

11 12 Property Value

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

16 Description

17 If the specified key already exists in the collection, setting this property overwrites the
18 existing values with the specified value. (If the existing value contains a string of
19 multiple comma-delimited values, the complete string is replaced with a single instance
20 of the specified value.) If the specified key does not exist in the collection, setting this
21 property creates a new entry using the specified key and the specified value.

22
23 [Note: This property provides the ability to access a specific element in the current
24 instance using the following notation: myCollection[key].

25
26 To add the new value to the existing list of values, use the
27 System.Collections.Specialized.NameValueCollection.Add method.

28]
29

30 Exceptions

Exception	Condition
System.NotSupportedException	The property is being set and the current instance is

	read-only.
--	------------

1

2

1 NameValueCollection.Item Property

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

7 Summary

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

9 Parameters

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

10

11 Property Value

12 A `System.String` that contains the comma-separated list of values at the specified
13 index of the current instance.

14 Description

15 This property is read-only.

16

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

19

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

22

23]

24 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	<i>index</i> is outside the valid range of indices for the current instance.
System.NotSupportedException	The property is being set and the current instance is read-only.

1

2

1 NameValueCollection.SyncRoot Property

```
2 [ILAsm]  
3 .property object ICollection.SyncRoot { public hidebysig virtual abstract  
4 specialname object get_ICollection.SyncRoot() }  
5 [C#]  
6 object ICollection.SyncRoot { get; }
```

7 Summary

8 Implemented to support System.Collections.ICollection.

9

10 [*Note:* For more information, see System.Collections.ICollection.SyncRoot.]

11

12

13